Habilitation à Diriger les Recherches

Traffic Measurements and Data Analysis for DNS Security

Université Grenoble Alpes

Maciej Korczyński

December 17, 2021

Composition of the Jury:

Aiko Pras
Professor, University of Twente, Netherlands, Reviewer

Isabelle Chrisment
Professor, Université de Lorraine, France, Reviewer

Oliver Hohlfeld
Professor, Brandenburg University of Technology, Germany, Reviewer

Andrzej Duda
Professor, Grenoble Alpes University, France, Examiner

Philippe Elbaz-Vincent
Professor, Grenoble Alpes University, France, Chair of the Committee

Acknowledgements

Undoubtedly, this is the most important section of this dissertation. In advance, I would like to extend my apologies for not mentioning everyone, but there are many individuals without whom this thesis would not have been possible.

First, I would like to express my gratitude to five individuals who have profoundly influenced my approach to research, supported my academic journey, and shared similar values.

First and foremost, I want to express my deep appreciation to Andrzej Duda. He convinced me twice to come to Grenoble: first to begin my PhD, and later to return after several years as a permanent member of the Drakkar team. Both decisions proved to be the right ones. His constant support, trust, encouragement, and mentorship have been invaluable to me. Andrzej has played a pivotal role in shaping my academic journey. I am thankful for creating a workplace where we can all grow, for our collaborative efforts on research papers, and for dealing with French bureaucracy side by side. Andrzej has been like a father to me in the scientific world, for which I am deeply grateful.

I express my sincere appreciation to Michel van Eeten, whose emphasis on the empirical impact of research has shaped my perspective, highlighting the interdisciplinary nature of cybersecurity encompassing both technological and economic dimensions. He also provided me with the opportunity to supervise my first PhD students.

Furthermore, I am deeply grateful to Cristian Hesselman for his continued trust and support of my research throughout the years. Our collaboration demonstrates that industry and academia should and can effectively work together, creating significant impacts from research, technology and policy perspectives.

Special thanks are extended to Paul Vixie, for his consistent support and encouragement of our research. He emphasized the importance of taking a proactive position against cybercrime, highlighting the need to address it firmly to reduce the risks of Internet abuse.

I am also thankful to Nina Fefferman, who provided invaluable guidance during my early days in research, teaching me how to lead a team, be patient, and offer gentle support to young researchers.

I extend my appreciation to Aiko Pras, Isabelle Chrisment, Oliver Hohlfeld, Andrzej Duda, and Philippe Elbaz-Vincent for accepting to serve on my examination committee.

The strength lies within the team. I would like to express my gratitude to all the co-founders of our research, collaborators, co-authors (in particular to Victor Le Pochat, Tom Van Goethem, Michał Król, Carlos H. Gañán, Giovane C. M. Moura, and Oliver Gasser), and members of the Drakkar team. Special appreciation goes to the PhD students whom I have had the privilege to supervise and collaborate with over the years: Jan Bayer, Benjamin Ben, Simon Fernandez, Olivier Hureau, Qasim Lone, Sourena Maroofi, Arman Noroozian, Yevheniya Nosyk, and Samaneh Tajalizadehkhoob. Thank you for your trust, the shared moments, your contributions to the team, your initiatives to make the lab a pleasant place to come every day, and above all, for the invaluable human lessons I have learned from each of you. Not to mention all the scientific contributions we collectively managed to achieve.

Je tiens à exprimer mes sincères remerciements aux administrateurs du réseau et des systèmes universitaires, notamment à Gaëtan Enderlé et son équipe, ainsi qu'à Pierre Veyan, pour leur contribution à nos recherches. Leur ouverture à traiter un éventail de problèmes et de demandes atypiques liés à nos recherches en cybersécurité et en

mesures Internet a été inestimable. De plus, leur maintenance de nos serveurs et leur gestion patiente des notifications d'abus sont grandement appréciées. Sans leur aide, notre recherche n'aurait pas été possible. Un grand merci !

Je voudrais exprimer ma gratitude à tous les membres de notre service des ressources humaines pour avoir navigué à travers les complexités de la bureaucratie française, leur patience à mon égard, et avoir facilité nos vies, en particulier à Alexandra Guidi, Maud Chorier et Pascale Poulet.

Kochani rodzice i dziadkowie, dziękuję Wam, że zawsze byliście i jesteście ze mną, że wspieracie mnie w niełatwych decyzjach, dziękuję za Waszą miłość i wsparcie. To Wasza zasługa, że jestem tu gdzie jestem.

À côté de mes parents, je tiens à exprimer ma gratitude envers mes beaux-parents, Brigitte et Bernard, pour m'avoir accepté. Votre présence et votre soutien ont également contribué à cette thèse, même si vous n'en êtes peut-être pas pleinement conscients.

Last but certainly not least, Audrey, I want to thank you for being with me all these years and for your patience with my passion for research. You, Ewa, and Lia give me incredible strength and happiness in my life. I love you so much!

Abstract

The Domain Name System (DNS) protocol maps easy-to-remember domain names to their computer-friendly numeric labels, assigned to each Internet-connected device that uses the Internet Protocol. DNS is the most critical and largely unheralded protocol, in the absence of which Internet users would need to memorize IP addresses of all the Internet applications, including banking sites, emails, or social media.

In the early days of the Internet, as highlighted by Dr. Paul Vixie, scientists invested all their efforts in facilitating communications because they believed that "something like the Internet could become humanity's collective digital nervous system." When the DNS principles and specifications were designed nearly four decades ago, security consideration was not an issue because the Internet was a network of trusted users. Danny Hillis, an American inventor and scientist, when registering the third domain name on the Internet thought that he should register a few more just in case, but he felt that "it wouldn't be nice." This example illustrates the trust within the community; the trust that was also built into the protocols of the Internet, including DNS.

Today's Internet is not only "humanity's collective digital nervous system" but also a place where cybercriminals exploit technical vulnerabilities and human weaknesses for financial gain. Spammers, phishers, malware creators, speculators, or organized e-crime groups widely abuse the DNS protocol and domain names. DNS has become as critical for them to operate as it is for regular users.

Preventing registration of malicious domains is challenging because it requires assessing the (bad) intentions of domain owners. Prompt removal of domain names directly involved in e-crime requires collecting evidence or verifying evidence provided by trusted notifiers of malicious activity. DNS and hosting providers do not have the financial incentives to effectively confront domain name abuse.

The DNS infrastructure itself remains vulnerable to attacks due to not restrictive enough assumptions about cybercriminals and the threat model when designing protocols in the early days of the Internet. Newly discovered vulnerabilities inherent to the DNS design drive the development and deployment of new extensions to the DNS protocol. However, their uptake has been very slow. It has become less of a technology issue than an economic incentive problem, i.e., whether implementing such security technologies can be profitable for the operators deploying them.

The distributed nature and architecture of the DNS protocol also allow for increased Internet security and stability. One example in which DNS plays an important role is in email security protocols: the Sender Policy Framework (SPF) and the Domain-based Message Authentication, Reporting, and Conformance (DMARC). While the Simple Mail Transfer Protocol (SMTP), designed for email distribution, is inherently insecure, SPF and DMARC providing a set of rules stored in the 'TXT' records of DNS resources can eliminate the problem of domain spoofing. Cybercriminals also abuse the DNS protocol architecture and its features to enhance the resilience of malicious infrastructures, amplify attacks, and avoid detection. Just mention Automatically Generated Domains (AGD) combined with fast-flux networks or Distributed Reflective Denial-of-Service (DRDoS) attacks that leverage open DNS resolvers.

Motivated by the problems of DNS security and domain name abuse, this dissertation has been devoted to DNS security: to make communications more selective and more difficult for malicious actors so that the "collective digital nervous system" – the Internet – stays less affected, more secure, and trusted by their benign users. The first three contributions present DNS measurement studies related to weaknesses inherent

to Internet protocols and domain names that can lead to the exploitation of DNS infrastructure and domain names. The following three contributions present statistical and machine learning approaches related to domain name abuse based on traffic measurements and inferential analysis from DNS-related data.

The first contribution illuminates the problem of non-secure DNS dynamic updates, which allow a miscreant to manipulate DNS entries in the zone files of authoritative name servers. We refer to this type of attack as *zone poisoning*. In its simplest version, a malicious actor could replace an existing 'A' or 'MX' resource record (RR) in a zone file of an authoritative server and point the domain name to an IP address under control of an attacker, thus effectively hijacking the domain name. We present the first measurement study of the vulnerability. Among the vulnerable domains are governments, health care providers, and banks, demonstrating that the threat impacts important services. With this study and subsequent notifications to affected parties, we aim to improve the security of the DNS ecosystem.

Source Address Validation (SAV) is a standard aimed at discarding packets with spoofed source IP addresses. The absence of SAV for outgoing traffic is a root cause of DRDoS attacks and received widespread attention. While less obvious, the absence of *inbound* filtering enables an attacker to appear as an internal host of a network and reveals valuable information about the network infrastructure. It may enable other attack vectors such as DNS cache poisoning. As a second contribution, we present the results of the Closed Resolver Project that aims at mitigating the problem of inbound IP spoofing. We perform the first Internet-wide active measurement study to enumerate networks that do not enforce filtering of incoming packets based on their source addresses. To achieve this goal, we identify closed and open DNS resolvers that accept spoofed requests coming from the outside of their network. Our work implies that the absence of inbound SAV makes DNS resolvers vulnerable to several types of attacks, including DNS cache poisoning, DNS zone poisoning, NXNSAttack, or zero-day vulnerabilities in the DNS server software.

Sending forged emails by taking advantage of domain spoofing is a common technique used by attackers. The lack of appropriate email anti-spoofing schemes or their misconfiguration lead to successful phishing attacks or spam dissemination. In the third contribution, we evaluate the coverage of SPF and DMARC deployment in two large-scale campaigns measuring their global adoption rate and deployment by high-profile domains. We propose a new algorithm for identifying defensively registered domains and enumerating the domains with misconfigured SPF rules. We define for the first time, new threat models involving subdomain spoofing and present a methodology for preventing domain spoofing, a combination of good practices for managing SPF and DMARC records and analyzing DNS logs. Our measurement results show that a large part of the domains do not correctly configure the SPF and DMARC rules, which enables attackers to deliver forged emails to user inboxes. Finally, we report on remediation and its effects by presenting the results of notifications sent to Computer Security Incident Response Teams responsible for affected domains.

To enhance competition and choice in the domain name system, the Internet Corporation for Assigned Names and Numbers introduced the new generic Top-Level Domain (gTLD) program, which added hundreds of new gTLDs (e.g. .nyc, .top) to the root DNS zone. While the program arguably increased the range of domain names available to consumers, it has also created new opportunities for cybercriminals. To investigate this issue, in the fourth contribution, we present the first comparative study of abuse in the domains registered under the new gTLD program and legacy gTLDs (e.g. .com, .org).

We combine historical datasets from various sources, including DNS zone files, WHOIS records, passive and active DNS and HTTP measurements, and reputable domain name blacklists to study abuse across gTLDs. We find that the new gTLDs appear to have diverted abuse from the legacy gTLDs: while the *total* number of domains abused for spam remains stable across gTLDs, we observe a growing number of spam domains in new gTLDs, which suggests a shift from legacy gTLDs to new gTLDs. We also analyze the relationship between DNS abuse, operator security indicators, and the structural properties of new gTLDs. The results indicate that there is an inverse correlation between abuse and stricter registration policies. Our findings suggest that cybercriminals increasingly prefer to register, rather than hack, domain names and some new gTLDs have become a magnet for malicious actors. As the presented state of the art in gTLD abuse is in clear need of improvement, we have developed cases for modifying the existing safeguards and proposed new ones. ICANN is currently using these results to review the existing anti-abuse safeguards, evaluate their joint effects, and introduce more effective safeguards before an upcoming new gTLD rollout.

Malicious actors abuse thousands of domain names every day by launching large-scale attacks such as phishing or malware campaigns. While some domains are solely registered for malicious purposes, others are benign but get compromised and misused to serve malicious content. Existing methods for their detection can either predict malicious domains at the time of registration or identify indicators of an ongoing malicious activity conflating maliciously registered and compromised domains into common blacklists. Since the mitigation actions for these two types domains are different, in the fifth contribution, we propose COMAR (Classification of Compromised versus Maliciously Registered Domains), an approach to differentiate between compromised and maliciously registered domains, complementary to previously proposed domain reputation systems. We start with a thorough analysis of the domain life cycle to determine the relationship between each step and define its associated features. Based on the analysis, we define a set of 38 features costly to evade. We evaluate COMAR using phishing and malware blacklists and show that it can achieve high accuracy (97% accuracy with a 2.5% false-positive rate) *without* using any privileged or non-publicly available data, which makes it suitable for the use by any organization. We plan to deploy COMAR at two domain registry operators of the European country-code TLDs and set up an early notification system to facilitate the remediation of blacklisted domains.

In 2016, law enforcement dismantled the infrastructure of the Avalanche bulletproof hosting service, the largest takedown of a cybercrime operation so far. The malware families supported by Avalanche use Domain Generation Algorithms (DGAs) to generate random domain names for controlling their botnets. The takedown proactively targeted these presumably malicious domains, however, as coincidental collisions with legitimate domains are possible, investigators had first to classify domains to prevent undesirable harm to website owners and botnet victims. The constraints of this real-world takedown (proactive decisions without access to malware activity, no bulk patterns, and no active connections) mean that approaches based on the state of the art cannot be applied. The problem of classifying thousands of registered DGA domain names therefore required an extensive, painstaking manual effort by law enforcement investigators. To significantly reduce this effort without compromising correctness, we develop a model that automates the classification. Through a synergetic approach, we achieve an accuracy of 97.6% with ground truth from the 2017 and 2018 Avalanche takedowns. For the 2019 takedown, this translates into a reduction of 76.9% in manual investigation effort. Furthermore, we interpret the model to provide investigators with insights into

how benign and malicious domains differ in behavior, which features and data sources are the most important, and how the model can be applied according to the practical requirements of a real-world takedown. Finally, we assisted law enforcement agencies by applying our approach to the 2019 Avalanche takedown iteration.

It is beyond doubt that selective and secure DNS communication is the basis for a more secure and stable Internet. Armed with the experience of the early days of the Internet and technological advances providing several missing security blocks in DNS, our work contributes to the implementation of security protocols, the identification of new (old) security problems overlooked by the community, as well as the development of statistical and machine learning methods to help intermediaries more effectively mitigate domain name abuse.

Résumé

Le protocole DNS (Domain Name System) associe des noms de domaine faciles à mémoriser à leurs étiquettes numériques compréhensibles par les machines (adresses IP), attribuées à chaque appareil connecté à Internet. Le DNS est le protocole le plus critique et le plus méconnu, en l'absence duquel les utilisateurs d'Internet devraient mémoriser les adresses IP de toutes les applications, y compris les sites bancaires, les courriers électroniques ou les médias sociaux.

Aux premiers jours de l'Internet, comme l'a souligné le Dr. Paul Vixie, les scientifiques ont investi tous leurs efforts pour faciliter les communications, car ils pensaient que "quelque chose comme l'Internet pourrait devenir le système nerveux numérique collectif de l'humanité." Lorsque les principes et les spécifications du DNS ont été conçus il y a près de quarante ans, les considérations de sécurité ne posaient pas de problème, car l'Internet était un réseau d'utilisateurs de confiance. Danny Hillis, un inventeur et scientifique américain, lors de l'enregistrement du troisième nom de domaine sur Internet, a pensé qu'il devrait en enregistrer quelques autres au cas où, mais il a jugé que "ce ne serait pas bien." Cet exemple illustre la confiance au sein de la communauté, confiance qui a également été intégrée dans les protocoles de l'Internet, y compris le DNS.

L'Internet d'aujourd'hui n'est pas seulement "le système nerveux numérique collectif de l'humanité," mais aussi un lieu où les cybercriminels exploitent les vulnérabilités techniques et les faiblesses humaines à des fins lucratives. Les spammeurs, les phishers, les créateurs de malwares, les spéculateurs ou les groupes organisés de cybercriminalité abusent largement du protocole DNS et des noms de domaine. Le DNS est devenu aussi essentiel pour leur fonctionnement que pour celui des utilisateurs ordinaires.

La prévention de l'enregistrement de domaines malveillants est un défi car elle nécessite d'évaluer les intentions, possiblement mauvaises des propriétaires de domaines. La suppression rapide des noms de domaine directement impliqués dans la cybercriminalité nécessite de recueillir des preuves ou de vérifier les preuves fournies par des notificateurs de confiance de l'activité malveillante. Les fournisseurs de DNS et d'hébergement n'ont pas les incitations financières nécessaires pour lutter efficacement contre les abus de noms de domaine.

L'infrastructure DNS elle-même reste vulnérable aux attaques en raison de présomptions pas assez restrictives concernant les cybercriminels et du modèle de menaces lors de la conception des protocoles au début de l'Internet. Les vulnérabilités nouvellement découvertes qui sont inhérentes à la composition du DNS conduisent au développement et au déploiement de nouvelles extensions du protocole DNS. Cependant, leur adoption a été très lente. Il s'agit moins d'un problème technologique que d'un problème d'incitation économique, à savoir si la mise en œuvre de ces technologies de sécurité peut être rentable pour les opérateurs qui les déploient.

La nature et l'architecture distribuées du protocole DNS permettent également de renforcer la sécurité et la stabilité de l'Internet. Un exemple où le DNS joue un rôle important est celui des protocoles de sécurité du courrier électronique : Sender Policy Framework (SPF) et Domain-based Message Authentication, Reporting, and Conformance (DMARC). Alors que le protocole SMTP (Simple Mail Transfer Protocol), conçu pour la distribution du courrier électronique, est intrinsèquement non sécurisé, SPF et DMARC, en fournissant un ensemble de règles stockées dans les enregistrements 'TXT' des ressources DNS, peuvent éliminer le problème de l'usurpation de domaine. Cependant, les cybercriminels abusent également de l'architecture du protocole DNS

et de ses caractéristiques pour renforcer la résilience des infrastructures malveillantes, amplifier les attaques et éviter la détection. Il suffit de mentionner les domaines générés automatiquement (AGD) combinés aux réseaux à flux rapide ou les attaques par déni de service réflectif distribué (DRDoS) qui exploitent les résolveurs DNS ouverts.

Motivée par les problèmes de sécurité DNS et d'abus de noms de domaine, ce mémoire a été consacré à la sécurité DNS : rendre les communications plus difficilement exploitables par les acteurs malveillants afin que le "système nerveux numérique collectif" – l'Internet – reste moins affecté, plus sûr, et que ses utilisateurs légitimes lui fassent confiance. Les trois premières contributions présentent des études de mesure du DNS liées aux faiblesses inhérentes aux protocoles Internet et aux noms de domaine qui peuvent conduire à l'exploitation de l'infrastructure DNS et des noms de domaine. Les trois contributions suivantes présentent des approches statistiques et d'apprentissage automatique liées à l'abus de noms de domaine, basées sur des mesures de trafic et des analyses déductives à partir de données liées au DNS.

La première contribution met en lumière le problème des mises à jour dynamiques DNS non sécurisées qui permettent à un mécréant de manipuler les entrées DNS dans les fichiers de zone des serveurs de noms faisant autorité. Nous appelons ce type d'attaque "*zone poisoning*". Dans sa version la plus simple, un acteur malveillant peut remplacer un enregistrement de type 'A' ou 'MX' existant dans un fichier de zone d'un serveur faisant autorité et associer le nom de domaine à une adresse IP sous le contrôle d'un attaquant – détournant ainsi effectivement le nom de domaine. Nous présentons la première étude de mesure de cette vulnérabilité. Parmi les domaines vulnérables figurent des gouvernements, des hôpitaux et des banques, ce qui montre que la menace touche des services importants. Avec cette étude et les notifications consécutives aux parties concernées, nous visons à améliorer la sécurité de l'écosystème DNS.

La validation de l'adresse source (SAV) est un standard visant à rejeter les paquets dont l'adresse IP source est usurpée. L'absence de SAV pour le trafic sortant est une cause fondamentale des attaques de type DRDoS qui a été étudiée par un grand nombre de chercheurs. Bien que moins évidente, l'absence de filtrage *entrant* permet à un attaquant d'apparaître comme un hôte interne d'un réseau et révèle des informations importantes sur l'infrastructure du réseau. Elle peut permettre d'autres vecteurs d'attaque tels que l'empoisonnement du cache DNS. Comme deuxième contribution, nous présentons les résultats du projet Closed Resolver qui vise à atténuer le problème de l'usurpation d'adresse IP entrante. Nous réalisons la première étude de mesure active à l'échelle de l'Internet pour énumérer les réseaux qui n'appliquent pas le filtrage des paquets entrants en fonction de leurs adresses source. Pour atteindre cet objectif, nous identifions les résolveurs DNS fermés et ouverts qui acceptent les requêtes usurpées provenant de l'extérieur de leur réseau. Notre travail implique que l'absence de SAV entrant rend les résolveurs DNS vulnérables à plusieurs types d'attaques, y compris l'empoisonnement du cache DNS, l'empoisonnement de la zone DNS, l'attaque de type NXNSAttack, ou des vulnérabilités zero-day dans le logiciel de serveur DNS.

L'envoi de faux e-mails en profitant de l'usurpation de domaine est une technique courante utilisée par les attaquants. L'absence de mécanismes appropriés de lutte contre l'usurpation d'adresse électronique ou leur mauvaise configuration permettent de lancer avec succès des attaques de phishing ou de diffusion de spam. Dans la troisième contribution, nous évaluons le déploiement de SPF et DMARC dans deux campagnes à grande échelle, en mesurant leur taux d'adoption global et leur déploiement par des domaines importants. Nous proposons un nouvel algorithme pour identifier les domaines enregistrés de manière défensive et recenser les domaines dont les règles SPF sont mal

configurées. Nous définissons pour la première fois de nouveaux modèles de menace impliquant l'usurpation de sous-domaines et présentons une méthodologie pour prévenir l'usurpation de domaines, une combinaison de bonnes pratiques pour la gestion des enregistrements SPF et DMARC et l'analyse des journaux DNS. Nos résultats de mesures montrent qu'une grande partie des domaines ne configure pas correctement les règles SPF et DMARC, ce qui permet aux attaquants de délivrer de faux e-mails dans les boîtes de réception des utilisateurs. Enfin, nous rendons compte de la médiation et de ses effets en présentant les résultats des notifications envoyées aux équipes de réponse aux incidents de sécurité informatique responsables des domaines affectés.

Afin de renforcer la concurrence et le choix dans le système des noms de domaine, ICANN (Internet Corporation for Assigned Names and Numbers) a introduit le nouveau programme de domaine générique de premier niveau (gTLD) qui a ajouté des centaines de nouveaux gTLD (par exemple, .nyc, .top) à la zone DNS racine. Si le programme a sans doute augmenté la gamme de noms de domaine disponibles pour les consommateurs, il a également créé de nouvelles opportunités pour les cybercriminels. Pour étudier cette question, nous présentons dans la quatrième contribution la première étude comparative des abus dans les domaines enregistrés dans le cadre du nouveau programme gTLD et dans les gTLD traditionnels (par exemple, .com, .org). Nous combinons des ensembles de données historiques provenant de diverses sources, notamment des fichiers de zone DNS, des enregistrements WHOIS, des mesures DNS et HTTP passives et actives, et des listes noires de noms de domaine réputés pour étudier les abus dans les gTLD. Nous constatons que les nouveaux gTLDs semblent avoir détourné les abus des gTLDs traditionnels : alors que le nombre de domaines abusés pour le spam reste stable entre les gTLDs, nous observons un nombre croissant de domaines de spam dans les nouveaux gTLDs, ce qui suggère un déplacement des gTLDs traditionnels vers les nouveaux gTLDs. Nous analysons également la relation entre les abus de DNS, les indicateurs de sécurité des opérateurs et les propriétés structurelles des nouveaux gTLD. Les résultats indiquent qu'il existe une corrélation inverse entre les abus et les politiques d'enregistrement plus strictes. Nous constatons que les cybercriminels préfèrent de plus en plus enregistrer les noms de domaine plutôt que de les pirater et que certains nouveaux gTLD sont devenus un aimant pour les acteurs malveillants. Comme l'état actuel de la situation en matière d'abus des gTLD a clairement besoin d'être amélioré, nous avons élaboré des cas pour modifier les mesures de protection existantes et en avons proposé de nouvelles. L'ICANN utilise actuellement ces résultats pour réviser les mesures de protection anti-abus existantes, évaluer leurs effets conjoints et introduire des mesures de protection plus efficaces avant le prochain lancement d'un nouveau gTLD.

Les acteurs malveillants abusent chaque jour des milliers de noms de domaine en lançant des attaques à grande échelle telles que des campagnes de phishing ou de logiciels malveillants. Si certains domaines sont enregistrés uniquement à des fins malveillantes, d'autres sont bénins (légitimes) mais sont compromis et utilisés à mauvais escient pour servir du contenu malveillant. Les méthodes de détection existantes permettent soit de détecter les domaines malveillants au moment de leur enregistrement, soit d'identifier les indicateurs d'une activité malveillante en cours, en regroupant les domaines malveillants enregistrés et compromis dans des listes noires populaires. Étant donné que les mesures d'atténuation pour ces deux types de domaines sont différentes, dans la cinquième contribution, nous proposons COMAR (Classification of Compromised versus Maliciously Registered Domains), une approche permettant de différencier les domaines compromis et les domaines enregistrés de manière malveil-

lante, en complément des systèmes de réputation de domaines proposés précédemment. Nous commençons par une analyse approfondie du cycle de vie d'un domaine afin de déterminer la relation entre chaque étape et de définir les caractéristiques associées. Nous avons défini un ensemble de 38 propriétés qu'il est difficile de contourner. Nous évaluons COMAR à l'aide de listes noires d'hameçonnage et de logiciels malveillants et montrons qu'il peut atteindre une grande précision (97 % de précision avec un taux de faux positifs de 2,5 %) sans utiliser de données privilégiées ou non publiques, ce qui le rend utilisable par n'importe quelle organisation. Nous prévoyons de déployer COMAR chez deux opérateurs de registre de domaines des TLD européens (ccTLD) et de mettre en place un système de notification pour faciliter la remédiation des domaines figurant sur la liste noire.

En 2016, les forces de l'ordre ont démantelé Avalanche, l'infrastructure du service d'hébergement blindé, le plus grand démantèlement d'une opération de cybercriminalité à ce jour. Les familles de logiciels malveillants soutenues par Avalanche utilisent des algorithmes de génération de domaines (DGA) pour générer des noms de domaines aléatoires afin de contrôler leurs botnets. Le démantèlement cible de manière proactive ces domaines présumés malveillants ; toutefois, comme des collisions fortuites avec des domaines légitimes sont possibles, les investigateurs doivent d'abord classer les domaines pour éviter tout préjudice indésirable aux propriétaires de sites Web et aux victimes de botnets. Les contraintes de cette opération dans le monde réel (décisions proactives sans accès à l'activité des logiciels malveillants, absence de profils d'enregistrement en masse et de connexions actives) signifient que les approches basées sur l'état de l'art ne peuvent être appliquées. Le problème de la classification des milliers de noms de domaine enregistrés de la DGA a donc nécessité un effort manuel important et minutieux de la part des investigateurs des forces de l'ordre. Pour réduire considérablement cet effort sans compromettre l'exactitude, nous développons un modèle qui automatise la classification. Grâce à une approche synergique, nous obtenons une précision de 97,6 % avec la vérité terrain des démantèlements d'Avalanche de 2017 et 2018 ; pour le démantèlement de 2019, cela se traduit par une réduction de 76,9 % de l'effort d'enquête manuel. En outre, nous interprétons le modèle pour fournir aux investigateurs un aperçu de la façon dont les domaines bénins et malveillants diffèrent dans leur comportement, quelles caractéristiques et sources de données sont les plus importantes, et comment le modèle peut être appliqué en fonction des exigences pratiques d'un démantèlement dans le monde réel. Enfin, nous avons aidé les forces de l'ordre en appliquant notre approche à l'itération 2019 du démantèlement Avalanche.

Il ne fait aucun doute que la communication DNS sélective et sécurisée est le tremplin vers un Internet plus sûr et plus stable. Sur la base de l'expérience des premiers jours de l'Internet et des avancées technologiques fournissant plusieurs blocs de sécurité manquants dans le DNS, nos travaux contribuent à la mise en œuvre de protocoles de sécurité, à l'identification de nouveaux (et parfois anciens) problèmes de sécurité négligés par la communauté, ainsi qu'au développement de méthodes statistiques et d'apprentissage automatique pour aider les intermédiaires à atténuer plus efficacement les abus de noms de domaine.

Contents

Chapter 1

Introduction

During the 2014 Internet Hall of Fame Acceptance Speech, Dr. Paul Vixie, an American computer scientist who designed and deployed several Domain Name System (DNS) protocol extensions and applications used throughout the Internet today, said: "I spent the first half, let's say fifteen years, of my career trying to make communications easier because I could tell that something like the Internet could become humanity's collective digital nervous system. And I thought that it was a cool thing, I thought that it would be great. I spent roughly the second half, another fifteen years, trying to make communication harder, or at least more selective because of all criminals and spammers that we brought with us when we have created humanity's collective digital nervous system. I stand here, as it was mentioned before, on the shoulders of giants (...). When I realized that success is inevitable, I went around and I said: «How can I thank you guys?» and they said: «Pay it forward, that's what we did», so that's what I'm doing" [1]. This work was inspired and encouraged by Dr. Paul Vixie and has been devoted to DNS security, to make communications more selective, more difficult for malicious actors so that the "collective digital nervous system" – the Internet – stays less affected, more secure, and trusted by their benign users.

1.1 Domain Name System: Yesterday and Today

The Domain Name System (DNS) protocol maps human-readable, easy to remember domain names (e.g., `societegenerale.fr`) to their computer-friendly IP addresses (e.g., `193.178.154.48`) – numerical labels assigned to each device connected to the Internet that uses the Internet Protocol (IP). DNS can be then considered as the phone book of the Internet. In practice, DNS is the most critical (and largely unheralded) pro-

tocol, in the absence of which Internet users would need to memorize IP addresses of all the Internet applications, including banking sites, emails, or social media. It is misleading to believe that DNS is simple, well understood, or sufficiently well researched. It is enough to mention that 3,200 pages of various RFC documents related to DNS have been published by different authors since its inception [2].

1.1.1 Trust Built into the DNS Protocol and Internet Users

In the early days of ARPANET – the precursor of the Internet, the mapping of hostnames to IP addresses was maintained in a single file named `HOSTS.TXT` and distributed to all users via the File Transfer Protocol (FTP). The first version of this file was published in 1972. At that time, there were no Top-Level domains (TLDs), registries, or registrars yet. Numbers were assigned by the Internet Assigned Numbers Authority (IANA), managed at the time by Jon Postel [3]. The Network Information Center (NIC) maintained and published the `HOSTS.TXT` file for the rest of the network. It soon became evident that this solution had one systemic problem – it was not scalable. In 1983, Paul V. Mockapetris [4] proposed principles for dynamic and distributed domain name systems in RFC 882 [5] and RFC 883 [6] – essentially DNS as it is known today. Four years later, Mockapetris introduced the details of the DNS protocol implementation and specification [7, 8], but without *security considerations*, as the Internet was a network of trusted users.

Danny Hillis, an American inventor and scientist, to give an idea of the level of trust in the Internet community in the early days, gave an example of a domain name he registered at the time. He chose `think.com`, which was the third registered domain name on the Internet. He thought there were a lot of interesting domain names, and he should register a few more just in case, but he felt that "it wouldn't be nice" [9]. This example illustrates the trust within the community; the trust that was also built into the protocols of the Internet, including DNS.

1.1.2 Identifying and Filling the Gaps (Slowly) to Confront E-crime

Today's Internet is not only "humanity's collective digital nervous system" but also a place where cybercriminals exploit technical vulnerabilities and human weaknesses for financial gain. Spammers, phishers, speculators, bulletproof service providers, or organized e-crime groups widely abuse the DNS protocol and domain names. DNS has

become as critical for them to operate as it is for regular users. They register thousands (possibly more) new domain names every day or compromise legitimate websites to distribute malicious content and launch massive attacks ranging from phishing, botnet, malware drive-by-download to spam campaigns. Some believe that most new domain names are maliciously registered.

Preventing registration of malicious domains is challenging because it requires assessing the (bad) intentions of domain owners (registrants). Registrant identity verification, even if required, is rarely imposed. Some segments of the DNS industry are aggressively competing and facilitating the creation of new domain names in mere seconds for less than a dollar. Domain owners can use cryptocurrencies to pay registration fees or automatically register hundreds of domain names in bulk. Some intermediaries willingly or unwillingly facilitate cybercrime. In 2016, AlpNames Limited, an ICANN-accredited [10] registrar, supported the option to randomly generate and register up to 2,000 domains from a selection of 27 new generic TLDs (gTLDs), with registration prices sometimes under one dollar and using a variety of patterns such as time, cities, zip codes, or letters [11]. In the second quarter of 2016, Spamhaus blacklisted nearly 40% (1,2 Million) of all domains registered with AlpNames [12].

Prompt removal of domain names directly involved in e-crime is also a challenge. DNS service and hosting providers need to collect evidence (or verify evidence provided by trusted notifiers) of malicious activity. They need to assess whether mitigation at the DNS level is appropriate and whether removing the abused domain (perhaps benign but compromised) will not cause collateral damage to regular Internet users. Some entities involved in domain registration and hosting shift the responsibility of combating domain abuse from one to the other. They claim they do not have enough financial resources to fight DNS abuse or there are other intermediaries who are better positioned to mitigate abuse. In fact, they do not have enough incentives and pressure from end users, their competitors, and regulators to curb domain abuse. Currently, there is no widely accepted consensus on what DNS abuse is and what types of intermediaries should respond to address it.

The DNS infrastructure itself also remains vulnerable to attacks mainly (but not only) because the DNS protocol was designed with no (or little) security considerations. Newly discovered vulnerabilities in DNS are driving the development and deployment of new extensions to the DNS protocol. We do not mean flaws in the DNS software

code, but rather certain weaknesses inherent in its design. They are the result of not restrictive enough assumptions about cybercriminals and the threat model when designing protocols in the early days of the Internet. One of the classic examples is the DNS cache poisoning attack discovered by Dan Kaminsky in 2008 [13]. The DNS Security Extensions (DNSSEC) [14] protocol was proposed to protect users against this attack and ensure the authenticity and integrity of the results provided by DNS servers. However, the uptake of these extensions has been very slow [15]. It has become less of a technology issue than an economic incentive problem, i.e., whether implementing such security technologies can be profitable for the operators implementing them [16].

Since it is challenging for consumers or regulators to assess the security level of services provided by intermediaries, i.e., whether they effectively prevent malicious domain registrations or the extent to which they deploy security technologies (e.g., DNSSEC), researchers proposed security reputation metrics [17]. Existing metrics typically assess how frequently abuse incidents occur (or vulnerabilities are identified) or how timely incidents are remediated once they have occurred. They enable benchmarking operators and thus reduce so-called *information asymmetry* about the security of intermediary services [17]. Security reputation metrics may be used to govern responsible parties towards investing in security and help reduce cybercrime, which is as much a technical issue as a problem of economic incentives.

1.1.3 DNS and Internet Stability and Security

The distributed nature and architecture of the DNS protocol also allows for increased Internet security and stability. In 2002, there was a massive (for the time) DDoS attack of unknown origin on nine of the thirteen DNS root servers responsible for the operation of the DNS and essentially the entire Internet [18]. Internet Software Consortium (ISC) chairman at the time, Paul Vixie, said that the attack "was only visible to people who monitor root servers or whose backbones feed root servers" [18] and appeared to have minimal impact on end users. The root name server infrastructure is highly resilient and distributed, thus in practice its complete disruption seems unrealistic. The DNS infrastructure leverages the inherent features of DNS, such as caching or multiple authoritative name servers for the same zone. Unlike in the early 2000s, most of the thirteen individual root servers implement load balancing and anycast techniques [19] and are, in fact, globally distributed server clusters in multiple data centers.

Another example of where DNS plays a vital role in the security and stability of the Internet are DDoS Protection Services (DPS) [20], which victims can outsource remediation of DDoS attacks. Leading cloud-based DPS systems such as Cloudflare, Akamai, Incapsula, or Verisign redirect network traffic to the DPS infrastructure for cleaning. Once the traffic is filtered of malicious flows, the benign traffic is sent back to the customer network. The DNS protocol is often used in a variety of ways to redirect network traffic to a DPS, for example by replacing the 'A' record with an IP address assigned to the DPS. Alternatively, the DNS zone of a domain can be delegated to a name server belonging to the DPS [20].

Another example is the Simple Mail Transfer Protocol (SMTP) designed to distribute emails. SMTP is inherently insecure and provides no support for preventing email spoofing [21]. Therefore, sending bogus emails by using domain spoofing is a common technique used by attackers. The solution to the problem, and the first line of defense, is the Sender Policy Framework (SPF) [22] and the Domain-based Message Authentication, Reporting and Conformance (DMARC) [23] protocols, which restrict who can send an email on behalf of a domain and how an email should be processed. SPF and DMARC provide a set of rules in text form stored in 'TXT' records of DNS resources. Careful configuration of the extensions can completely eliminate the spoofing problem for a given domain [24, 25].

1.1.4 DNS as an Asset for Cybercriminals

Cybercriminals also abuse DNS protocol architecture and its features to enhance the resilience of malicious infrastructures, amplify attacks and avoid detection. Just mention malicious Algorithmically Generated Domains (AGD) used for botnet C&C communication combined with fast-flux networks [26] or Distributed Reflective Denial-of-Service (DRDoS) attacks that leverage open DNS resolvers [27]. The basic premise of the latter is to send relatively small requests to open hosts with a spoofed (modified) source IP address that reflect much larger responses to the attack victim. The primary cause of DRDoS attacks is the ability to spoof IP source addresses (network operators' failure to implement a standard called Source Address Validation, also known as BCP 38 [28]). Dr. Paul Vixie observed that: "Nowhere in the basic architecture of the Internet is there a more hideous flaw than in the lack of enforcement of simple source-address validation (SAV) by most gateways [29]." The second cause of DRDoS attacks are open

UDP-based hosts that respond to requests from all clients. Among the most abused protocols are misconfigured open DNS resolvers that allow unrestricted recursive resolution to any client on the Internet. In 2013, Jared Mauch presented at the North American Network Operators' Group (NANOG) meeting the Open Resolver Project [30]. He uncovered 34 Million DNS servers that responded to UDP/53 requests – twenty-six years after introducing the DNS protocol specification in RFCs 1034 and 1035. Despite many initiatives to mitigate the problem of open resolvers such as Computer Emergency Response Team (CERT) alerts [31], research indicating the scale and severity of the problem [27, 32], and continued notifications to network operators by ShadowServer or national CERTs [33], the issue has still not been resolved. According to the recent report of ShadowServer [34], over 6 million distinct IP addresses respond to DNS queries in some fashion, and almost 2 million unique IP addresses appear to be openly recursive DNS servers.

It is beyond doubt that selective and secure DNS communication is the gateway to a more secure and stable Internet. Armed with the experience of the early days of the Internet and technological advances providing several missing security blocks in DNS, the number one priority for the community should be implementing security protocols, incentivizing intermediaries to deploy them, and identifying new (old) security problems, possibly overlooked by the community. In the next section, we discuss the importance and impediments to traffic measurements and data analysis in improving DNS security and thereby increasing barriers to abuse by malicious actors.

1.2 From Traffic Measurements to Data Analysis

Many consider that the birth of Internet traffic measurements as a scientific discipline came with the Center for Applied Internet Data Analysis (CAIDA) project [35]. Since its beginning in 1997, DNS has been one of the CAIDA important research activities [36, 37]. Over the years, measurement and analysis of DNS data have proven essential for assessing the uptake of security protocols such as DNSSEC [15], identifying abused or maliciously registered domain names [38, 39], or even creating economic incentives by reducing information asymmetry about the security practices of hosting providers [40], for example.

1.2.1 Passive DNS Replication

One important source of DNS intelligence is the passive DNS data. Florian Weimer first introduced this concept in 2004 [41]. At that time, he described the idea of "passive DNS replication", which involves reconstructing a (partial) view of the data available in the global DNS in a central aggregated database that can be queried. This method is referred to as "passive" because the monitored DNS queries are triggered by clients and therefore it does not require active probing of name servers. The data is collected by sensors located "above" the recursive name server (DNS resolver), meaning that they replicate DNS communications between the DNS resolver and authoritative name servers, rather than monitors DNS queries sent by clients to the local resolver ("below" the recursive resolver). This principle generally assures the privacy of end-users. The original motivation for developing passive DNS replication was to create a reverse DNS lookup database that maps IP addresses to corresponding domain names. The DNS guidelines specified in RFC 1912 [42] require each 'A' record to have a corresponding 'PTR' (pointer) record that maps IP addresses to domain names. Since in practice this data is inadequate or incomplete (e.g., in case of shared hosting where hundreds or even thousands of domain names share the same IP address), passive DNS is a good alternative data source to provide reverse DNS lookups [40, 43, 44].

Passive DNS datasets very quickly proved to be very important sources of data for detecting and mitigating attacks such as botnet Command and Control (C&C) communication, phishing, trademark infringement, or spam delivery. Let us take the example of modern botnets. Early malware hard-coded the IP addresses of its C&C servers, so it was quite easy to blacklist or even take over the corresponding malicious infrastructure. Therefore, malware has evolved from hard-coding the IP addresses of C&C servers to dynamically creating domain names and updating the associated IP addresses of proxy or backend C&C servers. One technique for this dynamic approach is domain fluxing, in which domain generation algorithms (DGAs) regularly create hundreds or even thousands of algorithmically generated domains. Infected machines will then attempt to contact these domains, ignoring unreachable ones (unregistered or, sometimes, registered but unpublished in the zone). Such behavior can be detected using passive DNS: regularly queried unregistered domains trigger an abnormally high number of NXDOMAIN (non-existent domain) responses visible in passive DNS. Those algorithmically

generated domain names can be proactively blocked or even sinkholed by registry operators or registrars, for example. Moreover, the number of queries observed in the passive DNS can indicate the size of the botnet and the distribution of infected hosts across networks, even if we may only analyze traffic generated by recursive resolvers and not by end clients.

Since DNS is one of the key tools in the malicious activities of criminals, sometimes even subtle traces in DNS logs left behind at some point in time can be used against them to detect, for example, malicious domains or entire criminal infrastructures. Because DNS resource records are stored in passive DNS databases with timestamp information, it is possible to reconstruct the zone view and retrospectively analyze traffic patterns on a given day. Almost all the benefits of privacy-preserving passive DNS replication (i.e., no need to actively query DNS servers, discovery of resource records otherwise unavailable due to closed zone files, ability to analyze query patterns triggered by clients) have been extensively used by the research and operational security community. However, one aspect of passive DNS data that is often underestimated is the near real-time nature of the data. The implication is that methods can be developed to detect intrinsic relationships between DNS resource records or traffic patterns in near real-time to detect potential security incidents *before* they occur.

One problem with passive DNS data that we foresee over the next decade is its availability. On the one hand, there are initiatives like the European Data Sharing Collective – Security Information Exchange (SIE) Europe.[1] The mission of SIE Europe is to make the European digital economy safer by offering a platform to collect, aggregate, and share data, without personally identifiable information (PII). SIE Europe participants who share their data gain access to all other participants' aggregated passive DNS data for use in their cybersecurity initiatives. However, one of the obstacles to increasing the use of passive sensors monitoring DNS traffic worldwide and making them more accessible to the security research and operations community is the implementation of open public resolvers as well as DNS privacy solutions: the DNS-over-TLS (DoT) [45], DNS-over-HTTP (DoH) [46], DNS-over-QUIC (DoQ) [47] protocols, or Oblivious DNS (ODNS) [48]. Over the past three decades, we have seen a continuous push to move the access-side DNS (the recursive part of it) away from customer networks toward large providers that maintain open public resolvers, such as Google, Cisco, or Cloud-

[1] https://www.sie-europe.net

flare, which leads to a concentration of DNS queries with a small number of operators (Google, Clourflare, etc.) managing DNS traffic instead of local resolver operators. Some ISPs argue that it may lead to a risky monopoly over user DNS data [49]. While admins of local networks can prevent their customers from using public resolvers by blocking all DNS traffic leaving the local network on port 53, this becomes difficult when DNS traffic is sent over the HTTPS or TLS protocols. On the one hand, DoT or DoH protocols, a priori, increase user privacy and prevent user profiling. On the other hand, they reduce the visibility of DNS traffic, which is essential for threat detection.

1.2.2 Active DNS Measurements

Passive DNS is not without limitations. If users do not request domain names, they are requested sporadically, or clients use DNS-over-HTTPS-enabled browsers, domain names and returned values do not appear in passive DNS. Active DNS measurement is, therefore, an attractive alternative, especially since the research and operations community has high-speed DNS scanners such as ZDNS[2] available. More importantly, the technique does not raise privacy concerns. Since 2015, the OpenINTEL [50], an Active DNS Measurement Project aims to capture daily snapshots of the state of large portions of the global DNS. One of the motivations for developing the platform was the still limited access to passive DNS datasets. At the time of writing, the measurement platform includes registered domain names of almost all new gTLDs available through the Centralized Zone Data Service (CZDS) maintained by the Internet Corporation for Assigned Names and Numbers (ICANN), legacy gTLDs including .com, .net, .org or .biz, and a few country-code TLDs (ccTLDs) such as .nl, .dk or .at. OpenINTEL performs forward DNS measurements, for each domain name, with a fixed set of DNS requests (such as 'A', 'SOA', 'NS' or 'DS'), as well as reverse DNS measurements. Very much like historical passive DNS databases, OpenINTEL allows for retrospective analysis of the DNS state over time. For specific research and security problems, active DNS can compensate for information derived from passive DNS datasets without degrading the accuracy of these methods [51]. For example, in general, both active and passive DNS can give us insight into how a domain's behavior changes over time (e.g., IP address changes). However, due to the nature of active DNS, a researcher cannot assess how popular a domain name was in the past or is today, which can be accomplished using

[2] https://github.com/zmap/zdns

passive DNS data.

While OpenINTEL or other similar active DNS measurement projects such as Project Sonar by Rapid7[3] or a DNS-based Active Internet Observatory [52] provide invaluable resources for data analysis, some scientific problems require more tailored, specific measurements and data collection. Therefore, researchers develop specific active DNS measurement methods. They range from ad hoc collection of uncommon DNS records to sending particular DNS queries to check the behavior of resolvers or authoritative name servers to, finally, sophisticated measurement configurations involving, for example, IP or DNS spoofing. They require not only building a dedicated DNS infrastructure, but sometimes also developing custom DNS scanners for measurement, data collection and analysis.

Active (and passive) DNS methods can lead to discovering DNS misconfigurations or vulnerabilities, measuring the deployment of security technologies, analyzing attack vectors, or fingerprinting DNS servers. The following section will discuss the registration (WHOIS) data, which gives a complementary insight into the entities involved in domain registration and content hosting.

1.2.3 Registration Data

The DNS represents a large ecosystem in which several entities play a role for a domain name to be registered, secured, and maintained on the Web. In particular, domain registrars manage the registration of Internet domain names on behalf of their owners (registrants). Web hosting providers maintain server infrastructure used to host content related to the domain.

Mapping abused or vulnerable domain names to registrars and hosting providers that can prevent or mitigate security incidents (or misconfigurations) in the first place requires access to domain name (and IP) registration data often referred to as WHOIS. For example, a disproportionate concentration of domain names used for phishing attacks or spam distribution identified at a particular registrar may lead to questions about its preventive security measures or, in the most extreme cases, may indicate its criminal nature. Moreover, security researchers and cybercrime investigators consider the registration information stored in WHOIS to be vital to their efforts to keep Internet users and their organizations safe. On the one hand, analysis of WHOIS data

[3] https://opendata.rapid7.com/sonar.fdns_v2/

of maliciously registered domains may reveal specific registration patterns, which may lead to the discovery of other malicious domain names or even entire criminal infrastructures. On the other hand, researchers use WHOIS to identify at scale the contact details of operators of misconfigured DNS servers or abused domain names to inform them of security problems [53].

For years, however, there have been obstacles preventing security investigators and researchers from fully harnessing its potential to fight cybercriminals and enhance global security. The main problems are: collecting registration data, parsing data and its availability. *Collecting* registration data (regardless of the communication protocol used) requires large-scale measurements, feasible for companies with large infrastructures that can avoid blacklisting their IP addresses exceeding query limits used for harvesting WHOIS data. For the scientific community, there is no easy way to collect such data at scale.

For over 35 years, WHOIS has been the primary communication protocol for retrieving domain name (and IP) registration data (i.e., registrar, administrator, domain registrant and their contact information, registration and expiration dates, domain status, and authoritative name servers) [54, 55]. Its main limitation is its text-based, non-standardized format, resulting in cumbersome *parsing* to extract data for analysis (once it is collected). In 2015, the Internet Engineering Task Force (IETF) proposed the Registration Data Access Protocol (RDAP) [56] to standardize registration data in a common JSON format that generally does not require an extra parsing step to extract the information. ICANN has required generic TLD registries and registrars to implement the RDAP service by August 2019 [57], which undoubtedly has helped in the uptake of the protocol. However, as ICANN does not have contract authority to take compliance action against country-code TLD operators, RDAP implementation among them has been slow [58]. Finally, as recently established, response times to RDAP queries remain significantly slower than to WHOIS queries [59].

Retrieving registration information became even more problematic with the introduction of the General Data Protection Regulation (GDPR) on May 25, 2018. ICANN adopted the temporary specification for generic top-level domains (gTLDs) on how to publish the registration data of individuals [60] that prohibits domain registrars and registries from storing personal data in the public WHOIS database, in particular, the contact details of registrants (domain owners) and administrators. In the *absence* of di-

rect contact with the registrant, it is recommended to contact the relevant registrar who has to provide access to registrant contact information in "a reasonable time" [61]. However, this practice may cause significant delays in fixing vulnerabilities and mitigating abuse (e.g., a hacked website), and it does not scale. More importantly, security experts have relied heavily on WHOIS data (e.g., email addresses, zip codes, fake phone numbers provided by criminals at the time of registration) to investigate crimes, copyright infringement claims, track malware distribution, or large-scale phishing attacks. Following GDPR and a temporary specification introduced by ICANN, registrant data is now redacted from the public WHOIS, making it more challenging to identify the domain owner or patterns of mass malicious domain name registrations automatically. However, the passing of GDPR and the removal of personal information from the WHOIS database also directly impacted cybercriminals' strategies. For example, phishers used WHOIS information to personalize phishing messages by using registrants' personal information found in WHOIS [62]. More importantly, phishers used WHOIS information to build a list of recipients (registrants), i.e., potential victims of their attacks [62].

While there are obstacles to the collection, parsing, and – as a consequence of GDPR – lack of registrant data in the public WHOIS, it remains one of the valuable resources to combat e-crime. However, in more complex security problems requiring automated approaches, such as detection of maliciously registered domains, security researchers must collect additional types of data to build statistical or machine learning models that yield high accuracy to be practical. The next section will cover other types of data needed to address various DNS security problems.

1.2.4 Other Datasets Related to Domain Names

Several authors proposed techniques for detecting maliciously registered domains, malicious activity on compromised domains, or algorithmically generated domains used for botnet C&C communication. Building such a system first requires a better understanding of cybercriminals' and ordinary users' *intentions* when registering and maintaining a domain name. Registrants' intentions can be captured by several different characteristics obtained using passive or active data collection methods. In addition to the already discussed passively and actively gathered DNS and WHOIS information, there are other non-privileged and generally easily accessible datasets directly or indirectly related to domain names.

The linguistic (or lexical) features of registered domain names may indicate the intentions of their registrants. Ordinary users may choose meaningful, easier-to-remember domain names related to the services provided by the domain. Malicious actors preparing, for example, phishing attacks may choose deceptive names to lure ordinary users and steal their personal information (e.g., `bankofamerica-account.support`). For botnet C&C panels, domain names are likely to be longer and meaningless to increase the chance they have not yet been registered.

Once a domain name is registered, its owner prepares the necessary infrastructure for the offered (legitimate or malicious) service. These steps may include setting up a web server, deploying a web content management application, or ordering a Transport Layer Security (TLS) certificate for the domain name to build trust with site visitors. Regular domain owners typically put effort into creating meaningful content to increase visitor interest and thus site popularity. This popularity can be captured using popularity lists such as Alexa[4], Cisco Umbrella[5], Majestic[6], or Tranco[7] (freely available). Although the composition of popularity lists has proven susceptible to manipulation techniques [63, 64], they can provide more accurate classification results when combined with other domain names (and site) characteristics. Another useful data source indicating historical changes in site content and popularity is the Wayback Machine[8], a publicly available digital archive of the World Wide Web. It allows the user to "go back in time" and see how web pages looked in the past. The high number of registered captures over time may indicate that a given domain is harmless.

Certificate Transparency (CT)[9] is an Internet security standard developed by Google to monitor and audit digital SSL/TLS certificates. This standard has created a public log system that will ultimately record all certificates issued by trusted CAs, allowing for effective identification of non-compliant or maliciously issued certificates. Information from CT logs can provide vital information about the owners and their intentions. Using a TLS certificate, malicious actors can make their attacks appear more legitimate (for example, by displaying a green padlock in the browser address bar). Free TLS certificates do not require their owners to provide any personal information. Therefore,

[4] https://www.alexa.com/topsites
[5] https://umbrella-static.s3-us-west-1.amazonaws.com/index.html
[6] https://majestic.com/reports/majestic-million
[7] https://tranco-list.eu/
[8] https://web.archive.org/
[9] https://certificate.transparency.dev/

criminals may prefer to choose free TLS certificates over paid ones. On the other hand, domain owners who value their domain names may choose to deploy paid certificates. In particular, they may decide to go through the complicated process of issuing Extended Validation SSL Certificates (EV SSL), which requires strong identity proof of the owner and thus increases trust in the domain name.

In addition, malicious actors may or may not take the effort to create fully functional websites, depending on the type of abuse. Legitimate websites typically use more libraries and technologies to build a site, which is not required for malicious domains to function properly. Such datasets can be collected by web crawlers and are valuable for building machine learning-based web content features.

To date, however, there is no centralized platform for collecting and compiling the vast amount of Open Source Intelligence (OSINT) data available to the research community. Such a platform would help build machine learning methods to combat domain name-based cybercrime or compare the performance of proposed solutions (e.g., phishing detection systems) on standard datasets.

1.3 Organization of the Dissertation and Key Contributions

Motivated by the problems of DNS security and domain name abuse explained in Section 1.1, we present six contributions in this dissertation. The first three contributions (Chapters 2-4) present DNS measurement studies related to weaknesses inherent to Internet protocols and domain names that can lead to the exploitation of DNS infrastructure and domain names. The following three contributions (Chapters 5-7) present statistical and machine learning approaches related to domain name abuse based on traffic measurements and inferential analysis from DNS-related data, as explained in Section 1.2.

Below, we summarize each contribution, list relevant research articles, industry presentations, and blog posts to raise awareness about the identified problems and proposed solutions.

1.3.1 Chapter 2: "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates"

The first contribution illuminates the problem of non-secure DNS dynamic updates, which allow a miscreant to manipulate DNS entries in the zone files of authoritative name servers. We refer to this type of attack as *zone poisoning*. This chapter presents the first measurement study of the vulnerability. We analyze a random sample of 2.9 million domains and the Alexa top 1 million domains, and find that at least 1,877 (0.065%) and 587 (0.062%) of domains are vulnerable, respectively. Among the vulnerable domains are governments, health care providers and banks, demonstrating that the threat impacts important services. With this study and subsequent notifications to affected parties, we aim to improve the security of the DNS ecosystem.

List of Relevant Peer-Reviewed Publications:

1. "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates", Maciej Korczyński, Michal Król, and Michel van Eeten, ACM SIGCOMM Internet Measurement Conference (IMC), pages 271-278, November 2016

2. "Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning", Orcun Cetin, Carlos Ganan, Maciej Korczyński, and Michel van Eeten, WEIS, 2017

List of Industry Talks:

1. "Internet-wide Measurements for Cybersecurity: The Case of DNS Zone Poisoning" (speaker), French Cyber Defence and Strategy Conference organized by the Cercle National des Armees, France, July 2019

2. "Zone Poisoning and General Data Protection Regulation" (speaker), ICANN 63 meeting, Spain, October 2018

3. "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates (speaker), DNS-OARC Spring Workshop, Spain, May 2017

1.3.2 Chapter 3: "The Closed Resolver Project: Measuring the Deployment of Source Address Validation of Inbound Traffic"

Source Address Validation (SAV) is a standard aimed at discarding packets with spoofed source IP addresses. The absence of SAV for outgoing traffic is a root cause of Distributed Denial-of-Service (DDoS) attacks and received widespread attention. While less obvious, the absence of *inbound* filtering enables an attacker to appear as an internal host of a network and reveals valuable information about the network infrastructure. It may enable other attack vectors such as DNS cache poisoning or resource exhaustion attacks like NXNSAttack. In this chapter, we present the results of the Closed Resolver Project that aims at mitigating the problem of inbound IP spoofing. We perform the first Internet-wide active measurement study to enumerate networks that enforce (or not) filtering of incoming packets based on their source addresses, for both the IPv4 and IPv6 address spaces. To achieve this goal, we identify closed and open DNS resolvers that accept spoofed requests coming from the outside of their network. The proposed method provides the most complete picture of *inbound* SAV deployment by network providers. Our scans reveal that 48.9% IPv4 and 26% IPv6 of globally routable Autonomous Systems (AS) suffer from consistent or partial absence of inbound SAV. By identifying dual-stacked DNS resolvers, we additionally show that inbound filtering is less often deployed for IPv6 than for IPv4. Furthermore, we uncover approximately 2.5 M IPv4 and 100 K IPv6 closed resolvers that are not detectable by existing methodologies. Despite being closed, our work implies that the absence of inbound SAV makes these resolvers vulnerable to several types of attack, including NXNSAttack and zero-day vulnerabilities in the DNS server software.

List of Relevant Peer-Reviewed Publications:

1. "Don't Forget to Lock the Front Door! Inferring the Deployment of Source Address Validation of Inbound Traffic", Maciej Korczyński, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda, Passive and Active Measurement Conference (PAM), March 2020

2. "Inferring the Deployment of Inbound Source Address Validation Using DNS Resolvers", Maciej Korczyński, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, Andrzej Duda, ACM/IRTF Applied Networking Research Work-

shop (ANRW 2020), Spain, 2020

3. "The Closed Resolver Project: Measuring the Deployment of Inbound Source Address Validation" Yevheniya Nosyk, Maciej Korczyński, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, Andrzej Duda, *submitted for publication*

List of Industry Talks:

1. "Measuring the Deployment of Source Address Validation of Inbound Traffic and Notifications" (speaker), FIRST Symposium Latin America and Caribbean, Virtual, October 2021

2. "Closed Resolver Project: Measuring the Deployment of Source Address Validation of Inbound Traffic" (speaker), International Forum of Cybersecurity (FIC), France, September 2021

3. "Inferring the Deployment of Inbound Source Address Validation Using DNS Resolvers" (speaker), Rezopole, Internet eXchange Points (IXP) workshop, France, November 2020

List of Blog Posts:

1. "Are You Filtering for Inbound Spoofed Packets? Chances Are You're Not", Maciej Korczyński, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez and Andrzej Duda, Asia Pacific Network Information Centre (APNIC) Blog, *available at:* https://blog.apnic.net/2020/10/05/are-you-filtering-for-inbound-spoofed-packets-chances-are-youre-not

2. "Inferring the Deployment of Inbound Source Address Validation Using DNS Resolvers", Yevheniya Nosyk, Maciej Korczyński, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, Andrzej Duda, RIPE Network Coordination Centre Blog, *available at:* https://labs.ripe.net/author/yevheniya_nosyk/inferring-the-deployment-of-inbound-source-address-validation-using-dns-resolvers/

1.3.3 Chapter 4: "Adoption of Email Anti-Spoofing Schemes: Large Scale Analysis"

Sending forged emails by taking advantage of domain spoofing is a common technique used by attackers. The lack of appropriate email anti-spoofing schemes or their miscon-

figuration lead to successful phishing attacks or spam dissemination. In this chapter, we evaluate the extent of the SPF and DMARC[10] deployment in two large-scale campaigns measuring their global adoption rate with a scan of 236 million domains and high-profile domains of 139 countries. We propose a new algorithm for identifying defensively registered domains and enumerating the domains with misconfigured SPF rules by emulating the SPF check_function. We define for the first time new threat models involving subdomain spoofing and present a methodology for preventing domain spoofing, a combination of good practices for managing SPF and DMARC records and analyzing DNS logs. Our measurement results show that a large part of the domains do not correctly configure the SPF and DMARC rules, which enables attackers to successfully deliver forged emails to user inboxes. Finally, we report on remediation and its effects by presenting the results of notifications sent to CSIRTs responsible for affected domains in two separate campaigns.

List of Relevant Peer-Reviewed Publications:

1. "Adoption of Email Anti-Spoofing Schemes: A Large Scale Analysis", Sourena Maroofi, Maciej Korczyński, Arnold Holzel, and Andrzej Duda, IEEE Transactions on Network and Service Management, 2021

2. "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains", Sourena Maroofi, Maciej Korczyński and Andrzej Duda, Network Traffic Measurement and Analysis Conference (TMA 2020), Germany, 2020 (Best Paper Award)

List of Industry Talks:

1. "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains" (speaker), The Internet Days, DNS Meetup, Swedish Internet Foundation, November 2020

[10]SPF and DMARC rules are configured and stored in DNS 'TXT' records.

1.3.4 Chapter 5: "Cybercrime After the Sunrise: A Statistical Analysis of DNS abuse in New gTLDs"

To enhance competition and choice in the domain name system, ICANN introduced the new gTLD program, which added hundreds of new gTLDs (e.g. .nyc, .top) to the root DNS zone. While the program arguably increased the range of domain names available to consumers, it might also have created new opportunities for cybercriminals. To investigate this issue, we present the first comparative study of abuse in the domains registered under the new gTLD program and legacy gTLDs (18 in total, such as .com, .org). We combine historical datasets from various sources, including DNS zone files, WHOIS records, passive and active DNS and HTTP measurements, and 11 reputable abuse feeds to study abuse across gTLDs. We find that the new gTLDs appear to have diverted abuse from the legacy gTLDs: while the *total* number of domains abused for spam remains stable across gTLDs, we observe a growing number of spam domains in new gTLDs which suggests a shift from legacy gTLDs to new gTLDs. Although legacy gTLDs had a *rate* of 56.9 spam domains per 10,000 registrations (Q4 2016), new gTLDs experienced a rate of 526.6 in the same period–which is almost one order of magnitude higher. In this chapter, we also analyze the relationship between DNS abuse, operator security indicators and the structural properties of new gTLDs. The results indicate that there is an inverse correlation between abuse and stricter registration policies. Our findings suggest that cybercriminals increasingly prefer to register, rather than hack, domain names and some new gTLDs have become a magnet for malicious actors. ICANN is currently using these results to review the existing anti-abuse safeguards, evaluate their joint effects and to introduce more effective safeguards before an upcoming new gTLD rollout.

List of Relevant Peer-Reviewed Publications:

1. "Cybercrime After the Sunrise: A Statistical Analysis of DNS Abuse in New gTLDs", Maciej Korczyński, Maarten Wullink, Samaneh Tajalizadehkhoob, Giovane C.M. Moura, Arman Noroozian, Drew Bagley, Cristian Hesselman, ACM Asia Conference on Computer and Communications Security (AsiaCCS 2018), Korea, June 2018

List of Industry Talks:

1. "Trends in Abuse: New and Legacy gTLDs" (speaker), 41st M3AAWG General Meeting, Canada, September 2017
2. "Statistical Analysis of DNS Abuse in gTLDs (SADAG)" (invited speaker), ICANN 59 meeting, South Africa, June 2017
3. "Statistical Analysis of DNS Abuse in generic Top-Level Domains", ICANN meeting (invited speaker), Denmark, March 2017

1.3.5 Chapter 6: "COMAR: Classification of Compromised versus Maliciously Registered Domains"

Miscreants abuse thousands of domain names every day by launching large-scale attacks such as phishing or malware campaigns. While some domains are solely registered for malicious purposes, others are benign but get compromised and misused to serve malicious content. Existing methods for their detection can either predict malicious domains at the time of registration or identify indicators of an ongoing malicious activity conflating maliciously registered and compromised domains into common blacklists. Since the mitigation actions for these two types domains are different, in this chapter, we propose COMAR, an approach to differentiate between compromised and maliciously registered domains, complementary to previously proposed domain reputation systems. We start the chapter with a thorough analysis of the domain life cycle to determine the relationship between each step and define its associated features. COMAR uses a set of 38 features costly to evade. We evaluate COMAR using phishing and malware blacklists and show that it can achieve high accuracy (97% accuracy with a 2.5% false-positive rate) *without* using any privileged or non-publicly available data, which makes it suitable for the use by any organization. We plan to deploy COMAR at two domain registry operators of the European country-code TLDs and set up an early notification system to facilitate the remediation of blacklisted domains.

List of Relevant Peer-Reviewed Publications:

1. "COMAR: Classification of Compromised versus Maliciously Registered Domains", Sourena Maroofi, Maciej Korczyński, Cristian Hesselman, Benoit Ampeau and Andrzej Duda, IEEE European Symposium on Security and Privacy (EuroS&P 2020), Italy, September 2020

List of Industry Talks:

1. "Exploring the Edges to Reach Consensus" (invited speaker, panelist), DNS Abuse Forum, May 2021
2. "Classification of Compromised versus Maliciously Registered Domains" (speaker), ICANN 70 TechDay, March 2021
3. "Classification of Compromised versus Maliciously Registered Domains" (speaker), ICANN DNS Symposium, May 2021

List of Blog Posts:

1. "Franco-Dutch research project on automatic classification of domain name abuse", AFNIC (registry of .fr domain names) Blog, Maciej Korczyński, Cristian Hesselman, Benoît Ampeau, *available at:* https://www.afnic.fr/en/observatory-and-resources/expert-papers/franco-dutch-research-project-on-automatic-classification-of-domain-name-abuse
2. "Distinguishing exploited from malicious domain names using COMAR. Key findings and future directions" SIDN (registry of .nl domain names) Blog, Sourena Maroofi, Maciej Korczyński, Benoît Ampeau, Thymen Wabeke, Cristian Hesselman, Andrzej Duda, *available at:* https://www.sidnlabs.nl/en/news-and-blogs/distinguishing-exploited-from-malicious-domain-names-using-comar

1.3.6 Chapter 7: "A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints"

In 2016, law enforcement dismantled the infrastructure of the Avalanche bulletproof hosting service, the largest takedown of a cybercrime operation so far. The malware families supported by Avalanche use Domain Generation Algorithms (DGAs) to generate random domain names for controlling their botnets. The takedown proactively targets these presumably malicious domains; however, as coincidental collisions with legitimate domains are possible, investigators must first classify domains to prevent undesirable harm to website owners and botnet victims. The constraints of this real-world takedown (proactive decisions without access to malware activity, no bulk patterns and

no active connections) mean that approaches from the state of the art cannot be applied. The problem of classifying thousands of registered DGA domain names therefore required an extensive, painstaking manual effort by law enforcement investigators. To significantly reduce this effort without compromising correctness, we develop a model that automates the classification. Through a synergetic approach, we achieve an accuracy of 97.6% with ground truth from the 2017 and 2018 Avalanche takedowns; for the 2019 takedown, this translates into a reduction of 76.9% in manual investigation effort. Furthermore, we interpret the model to provide investigators with insights into how benign and malicious domains differ in behavior, which features and data sources are most important, and how the model can be applied according to the practical requirements of a real-world takedown.

List of Relevant Peer-Reviewed Publications:

1. "A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints", Victor Le Pochat, Tim Van hamme, Sourena Maroofi, Tom Van Goethem, Davy Preuveneers, Andrzej Duda, Wouter Joosen and Maciej Korczyński, Network and Distributed System Security Symposium (NDSS 2020), California, February 2020

List of Industry Talks:

1. "A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints" (speaker), Passive DNS Hunters Working Group, Austria, October 2019

Chapter 2

Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates

Coauthors: Maciej Korczyński, Michal Król, and Michel van Eeten

2.1 Introduction

The Domain Name System (DNS) provides a critical service for all Internet applications that depend on domain names. Over the years, a variety of threats have emerged that undermine the trustworthy resolution of domain names into IP addresses. Two well-known attacks are cache poisoning [65] and malicious name resolution services [66, 67]. What these attacks share in common is that they compromise the resolution path somewhere between the user and the authoritative name server for a domain.

In this study, we explore an attack against the authoritative end of the path: the zone file of the authoritative name server itself. We detail how the vulnerable-by-design, non-secure DNS dynamic update protocol extension potentially allows anyone who can reach an authoritative name server to update the content of its zone file. The attacker only needs to know the name of the zone and the name server for that zone. The vulnerability was indicated already in 1997 by Vixie *et al.* in RFC 2136 [68], but its relevance in the current DNS landscape has not been recognized nor studied.

We refer to this type of attack as to *zone poisoning*. In the simplest version of an attack, a miscreant could replace an existing `A` or `MX` resource record (RR) in a zone file

of an authoritative server and point the domain name to an IP address under control of an attacker.

We already know that criminals are interested in hacking DNS records of legitimate domains from the practice of *domain shadowing*, where registrant credentials are compromised in order to create a large volume of subdomains of a legitimate domain. They are used for, among other things, distributing malware exploit kits [69]. A more ambitious vector is hacking the registrars directly, as illustrated by the attack of Syrian Electronic Army on Melbourne IT, the registrar for the New York Times and Twitter [70]. In contrast to these attacks, zone poisoning does not require compromising registrants or registrars, but is as simple as sending a single RFC-compliant DNS dynamic update packet to a misconfigured server.

We present the first study to detail this vulnerability and measure its prevalence in the wild. Our main contributions are summarized as follows:

- We analyze the root cause of non-secure dynamic updates and how they can be exploited.
- We measure which domains allow non-secure dynamic updates in a random sample of 1% from 286 million domains and find that 0.065% is vulnerable. Surprisingly, we find a similar rate (0.062%, meaning 587 domains) for the Alexa top 1 million domains.
- Alarmingly, we find a significant number of domains of national governments, universities, and businesses, including nine domains belonging to banks in Europe, Middle East, and Asia, from the domain of a private banking firm to a domain belonging to one of the largest banks in the world.
- We find significant concentrations of the vulnerability: securing the zone files of just 10 providers would reduce the prevalence of the issue with 88.6% in the random sample.
- We observe suspicious domains among the vulnerable population, but find no direct evidence of ongoing attacks.
- We find that most vulnerable servers are running Windows DNS, NLnetLabs NSD, and ISC BIND.

The objective of this study is to strengthen the security of DNS. We notified all operators of non-secure servers discovered during our measurements.

2.2 Background

The DNS protocol was initially designed to support queries of a statically configured database. Most of the data in the system was updated manually and expected to change only slowly [7]. However, with the introduction of dynamic allocation of network addresses to hosts [71], a more dynamic update mechanism for DNS became essential.

2.2.1 Dynamic Updates in DNS

DNS dynamic update specifications have been introduced by Vixie *et al.* in RFC 2136 [68] in 1997. Following this specification, one can add or delete any type of RR, such as `A`, `AAAA`, `CNAME`, or `NS`. The proposed `UPDATE` message complies with the standard DNS message format (cf. RFC 1035 [8]).

When a primary master server that supports dynamic updates receives an update request, it verifies: *i)* if all prerequisites defined by the requestor are met (e.g. check whether a specific record does or does not exist) and *ii)* whether restrictions are set regarding which hosts are allowed to make updates and, if so, whether those restrictions are met. If no restrictions are defined, anyone who knows the name of the zone and the name server for that zone is capable of updating its content. This constitutes a serious technological vulnerability indicated by Vixie *et al.* in RFC 2136 [68]. If the request is sent to an authoritative slave server, it is expected that it will be forwarded towards the primary server that is able to modify the zone file.

2.2.2 Secure DNS Dynamic Updates

Vixie *et al.* strongly recommended the use of security measures such as those described in RFC 2137 [72] (superseded by RFC 3007 [73]). If secure communication is not implemented, it is expected that an authoritative server accepts the dynamic updates only from a statically configured IP address of, for example, a DHCP server [68]. In RFC 2137, Donald Eastlake describes how to use the DNS Security Extensions (DNSSEC) [14] to restrict dynamic updates to authorized entities based on cryptographic keys [72]. However, using the public key mechanism is less efficient and harder to manage. Three years after the introduction of DNS dynamic updates, Vixie *et al.* proposed an efficient, lightweight alternative to authenticate dynamic updates: Secret Key Transaction Authentication for DNS (TSIG), which is based on shared secret keys

and message authentication code (MAC) [74].

2.2.3 Implementations

We now analyze common implementations of DNS dynamic updates, paying special attention to the default protocol configurations.

BIND: Berkeley Internet Name Domain (BIND) is open source and the most widely used DNS software on the Internet [75]. Version 8, released in 1997, first included a dynamic DNS component [76, 77]. In BIND 8 and 9, dynamic updates are disabled by default. An administrator can add `allow-update` in the zone configuration and specify the hosts that are allowed to update records. An address match list can include entire subnetworks or the built-in argument `any`, that allows all hosts to make dynamic updates. Since BIND 8.2, released in 1999, the address match list supports TSIG. The basic configuration is still supported, however. Since BIND 9.1, slave servers are allowed to forward dynamic updates to a master server (RFC 2136 [78]). These can use address match lists similar to those of the master, meaning that non-secure configurations provide an additional path for a miscreant, as updates forwarded by the slave will be accepted by the master, regardless of the original requestor.

Microsoft DNS: Windows 2000 is the first operating system developed by Microsoft that supported DNS dynamic updates [79]. The server can be configured either as standard primary or as Microsoft's Active Directory–integrated zone [80]. Windows 2000 and its successors, i.e. Windows Server 2003 [81], 2008 [82], and 2012 [83], all support secure dynamic updates. They implement an extended TSIG algorithm (RFC 3645 [84]). When an administrator creates an Active Directory–integrated zone, by default the server allows only secure updates via extended TSIG. However, the server can also be configured for no or non-secure dynamic updates. More importantly, the secure update functionality is not available for standard primary zones. In any primary zone configured for DNS dynamic updates, anyone can modify zones.

Other Implementations: As indicated in RFC 2137 [72], any zone file allowing dynamic updates is less secure than the one configured statically. Some of the popular open-source authoritative servers such as Name Server Daemon (NSD) developed by NLnet Labs [85], DJBDNS created by Daniel J. Bernstein [86], or Unlogic Eagle DNS [87] do not support dynamic updates. However, the functionality is sometimes

added via external tools[1,2]. PowerDNS has recently added the dynamic update component. According to the documentation, by default all IP ranges are allowed to perform updates [88]. Our lab experiments (cf. Section 2.4.1) reveal, however, that by default only loopback IP space can make dynamic updates.

In short: common implementations not only support vulnerable configurations, such as accepting requests from all hosts, but some are vulnerable by default. Of the two common security mechanisms, TSIG-variants and address match lists, only the former provides a reliable defense to malicious updates. Since the attack only needs a single UDP packet, an attacker can guess and spoof source IP addresses on the match list. This risk could be mitigated by restricting dynamic updates to the TCP protocol only.

2.3 Threat Model

We refer to an attack that exploits non-secure dynamic updates as *zone poisoning*. This attack itself is nothing more than sending a single RFC-compliant pack- et. The requirements are: *i)* non-secure updates are allowed by an authoritative server for a given zone *ii)* the miscreant knows the name of a zone and its name server.

An attacker can replace existing `A` or `MX` RRs in a zone file and point the domain to an IP address controlled by the attacker and potentially running a fake web or mail server. This would hijack the domain and allow the attacker to determine where clients or their emails go.

A miscreant could also abuse the reputation of a legitimate domain (e.g. `shopping.pl`) and add an extra `A` RR to an existing zone file that associates an IP address of a fake web server with a malicious subdomain (e.g. `paypal.account.shopping.pl`). An interesting variant is to *delegate* a malicious subdomain of a legitimate domain to the criminal's own DNS server. This would allow him to generate as many new subdomains as needed, without making additional update requests.

Non-secure updates could also be abused to acquire a Domain Validated (DV) SSL certificate for the vulnerable domain name, to be used in impersonation attacks. DV SSL certs are validated and provisioned automatically using a system of "challenge-response" emails. The attacker could re-route the confirmation message to the contact email listed in `WHOIS` via a dynamic update for the mail server domain.

[1] https://www.sixxs.net/wiki/NSD
[2] http://www.thismetalsky.org/projects/dhcp_dns

2.4 Methodology

2.4.1 Lab Experiments

We performed lab experiments to establish if and how the protocol allows unauthorized dynamic updates, in particular adding, deleting and modifying existing records in the zone. We selected BIND 9.8.4 and PowerDNS 4.0.0-alpha2 as case studies, as both implementations are non-commercial and widely used. We configured master servers for our domain name (e.g., `example.com`) and we tested various configuration setups as explained in Section 2.2.3. To perform updates, we used both the standard Linux `nsupdate`[3] command and our own scanner (see Section 2.4.2). Updates were sent from both legitimate and spoofed source IP addresses on the address match list.

The update requests successfully added and deleted `A`, `AAAA`, `NS`, `MX`, `PTR`, `SOA` and `TXT` RRs corresponding to the domain name (`example.com`), as well as extra records for subdomain names (`researchdelft.example.com`). This way, we were also able to replace a pre-existing `A` RR (`example.com`) that had been manually added to the zone file at the beginning of the study. More specifically, using dynamic updates, we first added an extra `A` record that associated the domain name with a new IP address, and then removed the original one. Finally, for BIND we also configured the slave server to forward updates towards the master. As expected, the changes were accepted by the master even though the original requestor is allowed to make changes only in the slave server.

To conclude, our lab experiments demonstrate that systems which allow non-secure dynamic updates are vulnerable to attacks that can "modify" existing records and add new records. Non-secure update mechanisms cover both overly promiscuous address match lists ("`any`") as well as more focused match lists, which can be bypassed via IP spoofing.

2.4.2 Scanning Setup

To assess the potential impact of non-secure dynamic updates, we have developed an efficient scanner capable of sending DNS packets compliant with RFC 2136 [68]. The scanner attempts to add an extra `A` record to the zone file, associating a new upper-level domain, `researchdelft`, with the IP address of our project's web server. We do not

[3] http://linux.die.net/man/8/nsupdate

spoof the source IP address of the update request. Our web server describes the project and provides a method to opt-out from our scans. Note that we have not received a single abuse complaint or opt-out request – which might mean that the insertion of the record was not seen as problematic or, perhaps more likely, that the insertion went unnoticed. The scan does not interact with the existing data in the zone file. Since our request is technically equivalent to a regular update request, we do not expect it to interfere with normal activity and have seen no evidence to the contrary.

We analyzed responses of authoritative name servers and performed DNS lookups to verify if our domain resolved to our web server's IP address. We also performed a ten-day long study to estimate the time the added RR stays in a zone. Finally, we removed the test DNS record by sending a *delete* `UDPATE` request and then tried to resolve it again. All added records were successfully deleted.

2.4.3 Ethical Considerations

While vulnerability scanning has become an established part of security research, our approach does raise ethical questions because of the fact that the only valid method available to us for assessing the vulnerability of a DNS server was to add a record to the zone file.

We have submitted the study to the TU Delft Human Research Ethics Committee. The committee evaluated our request and stated that we did not need their authorization since we were not conducting human subjects research. While this makes sense, it also signals that current institutional review procedures are not set up to evaluate ethical issues in computer security.

We have assessed our work using the principles outlined in the Menlo report [89]. We do not collect data on persons. Getting informed consent before adding a record to the zone file is both unpractical and would introduce selection bias, since administrators of well-secured servers are more likely to consent. We do provide a clear opt-out mechanism via the website referenced in the added DNS record. The site also provides full transparency regarding the study and its objectives.

Our approach in testing the vulnerability has been designed to have as minimal impact as possible: we send a single RFC-compliant packet. We do not read, change or otherwise engage with any existing records. We feel the drawback of lacking consent from server operators is outweighed by the benefits of our measurement for those oper-

Table 2.1: Datasets

#	1% Sample	Alexa 1M
Domains	2,865,393	947,823
NS	510,850	487,515
IPs of NS	438,478	418,251
Domain–NS–IP	27,499,061	7,368,659

ators: to be made aware of a critical vulnerability in their DNS server. All notifications have been completed before the publication of this paper. The new record is highly unlikely to be discovered by accident and it is removed at the end of the study.

2.4.4 Dataset

To measure the prevalence of non-secure configurations, we collected data for two samples: a random sample of 1% of the domain space and the Alexa top 1 million domains (or Alexa 1M) [90].

First, we extracted all domains observed in two complementary datasets between Jan 2015 and Jan 2016: *i)* DNSDB that is a large passive DNS database fed by hundreds of sensors across the world, operated by Farsight Security [91], which generously provided access to us and *ii)* Project Sonar Data Repository obtained though `ANY` RR requests, made available by Rapid7 Labs [92].

From the total 286,788,250 unique domains in the set, we randomly sampled 1%. For that sample and for the Alexa 1M, we enumerated all observed combinations of name servers and their IP addresses in both datasets: over 27 and 7 million, respectively (cf. Table 2.1). The long period of observation and the fact that DNSDB contains many entries that are poisoned either maliciously [66, 67] or unintentionally [93], means we expected a lot of IP addresses on the list to be obsolete, but we wanted to find as many as possible.

We performed the vulnerability assessment against the random sample on Mar 30, 2016 and against the Alexa 1M on Apr 10, 2016. For each domain, we sent an `UPDATE` request directly to all IP addresses on the list. As expected, many did not respond. Next to obsolete NS information, this can also indicate network filtering and other policies at work. We received responses from 6.0 million (random sample) and 2.3 million (Alexa 1M) name servers (see Table 2.2).

Table 2.2: DNS responses to `UPDATE` requests

DNS	1% Sample		Alexa 1M	
Response	in #	in %	in #	in %
All	6,007,462	100	2,294,099	100
`REFUSED`	2,325,377	38.7	1,265,544	55.2
`FORMERR`	1,374,015	22.8	260,094	11.3
`NOTAUTH`	1,198,337	19.9	357,442	15.6
`NOTIMP`	727,734	12.1	357,592	15.6
`SOA`	237,175	3.9	18,241	0.8
`SQR`*	114,677	1.9	25,851	1.1
`NOERROR`	13,580	0.2	5,093	0.2
`SERVFAIL`	6,621	0.2	3,830	0.2
Other	9,946	0.2	412	0

* Standard Query Response

2.5 Results

2.5.1 Prevalence of Vulnerable Resources

Table 2.2 summarizes the DNS status codes received in response packets related to the `UPDATE` requests. As expected, the great majority of requests fail to add RRs to the zone. The most common code is `REFUSED`, meaning that the server refuses to perform the operation for security or policy reasons. Around 12.1% and 15.6% of name servers signal `NOTIMP` meaning that they do not implement the protocol extension, whereas 22.8% and 11.3% of servers are not even able to parse and interpret the dynamic update request and signal `FORMERR`. Next, 19.9% and 15.6% of name servers signal that they are not authoritative for the zone. The main reason for DNS responses with the `NOTAUTH` error flag is the presence of obsolete NS information in our dataset as described in Section 2.4.4. Approximately 0.2% of servers signal `SERVFAIL` meaning that a hardware error or an out-of-memory condition might have taken place and a zone is restored to its state before this transaction [68]. We find 13,580 and 5,093 systems to respond with `NOERROR` status code for 1% sample and Alexa 1M respectively, which in both cases corresponds to 0.2% of responses. Note that `NOERROR` includes all responses with this status flag set regardless of whether the actual content of the zone has been updated.

We sent an `A` RR request to each of the potentially updated servers to verify if the zone file was indeed updated. For the random sample, we observed 2,626 successfully added `A` RRs, corresponding with 188 unique name servers and 1,877 unique domain

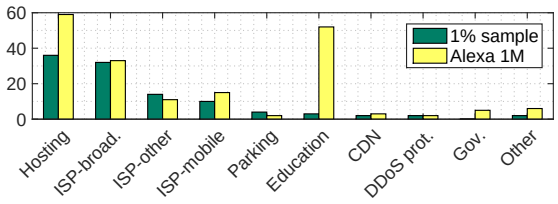


Figure 2.1: Types of providers hosting vulnerable domains.

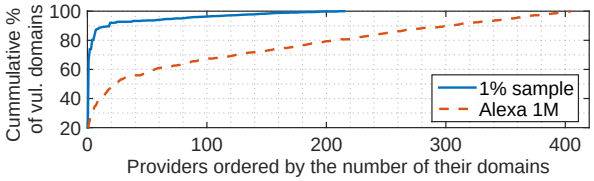
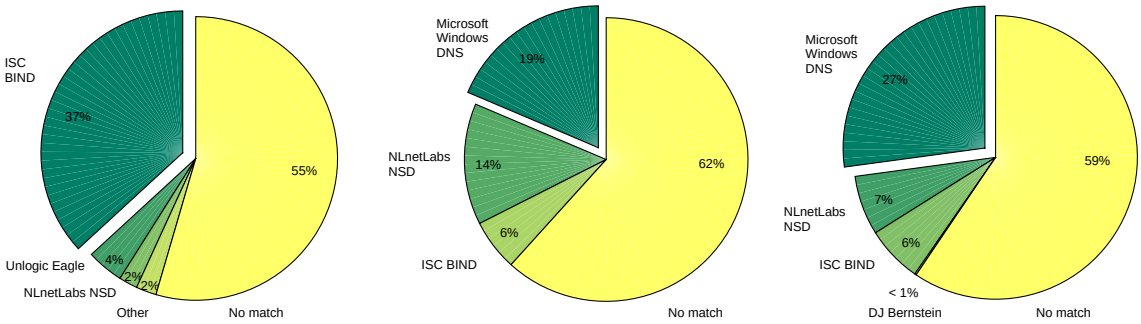


Figure 2.2: Cumulative distribution of vulnerable domains over providers.

names (0.065% of all randomly selected second-level domains). Surprisingly, we also observed 881 added `A` RRs that corresponded to 560 unique name servers and 587 domains from Alexa 1M (0.062%)



(a) All 1% sample (b) Vulnerable 1% sample (c) Alexa 1M

Figure 2.3: Results for `FPDNS` fingerprinting of authoritative servers for (a) all 1% sample of the domain space, (b) vulnerable 1% sample, (c) vulnerable Alexa 1M domains (data may not sum up to 100% due to the round-off error).

2.5.2 Affected Domains

To get a sense of the population of vulnerable domains, we first analyzed the type of network that hosts them. In earlier work, we developed a categorization of providers based on ground-truth data, manual labeling, `WHOIS` records and passive DNS data – for more details, see [94, 95]. We were able to classify 105 (out of 206) providers for the random sample and 210 (out of 398) for the Alexa 1M.

Figure 2.1 outlines the number of providers that have at least one vulnerable server in their network. As expected, hosting and ISP broadband constitute a great portion

of the affected providers. Interestingly, we observe misconfigured zones in as many as 52 educational networks in the Alexa 1M.

Figure 2.2 shows the cumulative distribution of vulnerable domains over providers. In the random sample, we find that 66.2% (1,149) of vulnerable domains are hosted on the infrastructure of a single Japanese broadband ISP. Reconfiguring the zone files of just 10 providers would reduce the prevalence of the issue with 88.6%. If this kind of concentration is representative of the overall domain space, then reaching out to a limited number of operators could greatly increasing the costs of finding vulnerable domains for cybercriminals. For the Alexa 1M, the pattern is much less concentrated. This might not be a major obstacle for remediation, though, as the high traffic sites in this set are typically professionally operated, so a comprehensive notification campaign might be effective.

We further analyze the cumulative distributions of vulnerable domains on DNS servers in descending order of the number of their common domains. For reasons of brevity, we highlight only the most interesting findings. In vulnerable 1% sample, we find that only one server is authoritative for as many as 1,635 (87%) domains, whereas in Alexa 1M, one DNS server is associated with 154 (26%) domains. As expected the cumulative concentrations per DNS servers are similar to the ones observed for providers (see Figure 2.2) as they operate the name servers themselves. In the 1% sample, for example, just six servers that share the same second- and top-level domain (`*.dnsserver.net`) are authoritative for 89.8% of the vulnerable domains, all hosted by the same broadband ISP in Japan.

We manually inspected the vulnerable domains from Alexa 1M. Table 2.3 lists the types of organizations affected. 'Business' is a large category that covers a heterogeneous set of companies, from small to large. In the latter category, we find a variety of sites related to global car manufacturers. We also find 56 vulnerable governmental sites in the North America, Europe, Asia – some national, some regional. Affected educational domains have a similar geographical distribution and include a few reputable universities. In health care, we found several hospitals and the domain of a national medical association. Remarkably, nine of the vulnerable domains belong to banks in Europe, Middle East and Asia, ranging from a small private banking firm to a domain of one of the largest banks in the world. In sum: the vulnerability is found to undermine the security of high-profile businesses, governments and organizations.

Table 2.3: Categories of vulnerable domains for Alexa 1M

Type	in #	in %
`Business`	181	31
`Entertainment`	92	15.7
`Educational`	90	15.3
`Governmental`	56	9.5
`News services`	41	7
`Adult`	13	2.2
`Financial services`	9	1.5
`Health care`	8	1.4
`Other`	95	16.2
Total	587	100

2.5.3 Exploitation

We looked for evidence of whether non-secure updates were exploited in the wild. We checked the overlap between the vulnerable domains and domains blacklisted by StopBadware [96] and the Anti-Phishing Working Group (APWG) [97] in 2015. The former consists of 1,016,961 unique fully qualified domain names (FQDNs) whereas the latter of 1,967,995. In APWG and StopBadware, respectively, we find 15 and 45 blacklisted FQDNs related to vulnerable second-level domains for Alexa 1M and only 1 and 5 for the random sample. After manual inspection of the website content, we did not find any compelling evidence that the observed domains are actually affected by malicious dynamic updates. The sites seemed legitimate and might either represent false positives or compromised resources.

We also searched in DNSDB for FQDN of vulnerable domains in association with common words in phishing attacks [98, 99], such as Paypal, Apple, Taobao, Amazon, etc. We find some suspicious FQDNs, for example, `shopping.*.com.*.*.edu` or `*.alibaba.com.*.ru`.

However, the sites are either offline or require some additional authentication to access. Some of them seem legitimate proxy services, e.g., university resources that require authorized access and redirect users to an external website.

2.5.4 Affected DNS Server Software

We surveyed the software running on non-secure authoritative name servers to see which packages were affected. On Apr 24, 2016 we scanned three groups of servers by using `FPDNS` software [100]: *i)* all 510,850 name servers from the random sample,

for comparative purposes; *ii)* the 188 vulnerable servers from the random sample; and *iii)* the 560 vulnerable servers from the Alexa 1M sample. Fingerprinting failed in many cases due to timeouts or inconclusive signatures. We were able to obtain software information for 45% (232,317), 38% (72), and 41% (227) of each respective group. We do not distinguish between different software versions as there are no major changes in the implementation of secure DNS dynamic updates (cf. Section 2.2.3). Figure 2.3 illustrates the results for DNS software fingerprinting. The majority of servers authoritative for the total random sample run BIND (37%). Microsoft Windows DNS constitutes just 0.5% of this group, while for the vulnerable groups it is the dominant package: 19% and 27%. The second and third largest groups of vulnerable server types are NLnetLabs NSD and ISC BIND. As the standard package of NLnetLabs NSD does not include the functionality for dynamic updates, we suspect that it might be added through some external, RFC-compliant plugin (see Section 2.2.3).

2.5.5 Survival Analysis

The final part of the study aimed to measure the survival times of the added records. We wanted to analyze whether these records would be removed and, if so, how soon. In other words, are there self-correcting mechanisms in place?

We initiated measurement on Apr 16, 2016. We first sent an update request to add an extra `A` RR (see Section 2.4.2) to the previously confirmed instances of vulnerable domains. We observe 3,920 successfully added `A` records that correspond to 1,870 domain names for 1% sample and 1,691 `A` RR associated with 584 domains for Alexa 1M domains.

Then, over a 10-day period, we performed DNS lookups every 4 hours—sending an `A` RR request to each of the IP addresses of the servers associated with vulnerable domains. We performed survival analysis on the results using the standard Kaplan-Meier estimator to approximate the survival function [101].

The results indicate a very small removal rate of the added record (cf. Figure 2.4). We do not know why some records were removed, but one plausible explanation is that the zone transfer from the primary master may have overwritten the added entries. At the end of our experiment, records were still present in around 94.3% (3,696) of the random sample and 95.9% (1,622) of the Alexa 1M domains. Interestingly enough, the Alexa 1M does not have a higher removal rate than the random sample; in fact, it does

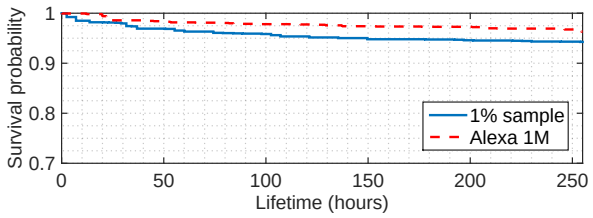


Figure 2.4: Survival analysis of `A` records added to vulnerable servers for 1% sample and Alexa 1M domains.

slightly worse. In light of the fact that we were not contacted by any of the operators of the non-secure servers, suggesting no one saw the added record, it seems that there are no other security mechanisms in place to discover and mitigate the threat.

2.6 Conclusions

We presented the first measurement study into the vulnerability of non-secure DNS dynamic updates, which enables an attack we referred to as *zone poisoning*. We have measured prevalence rates for a random sample of 2.9 million domains (0.065%) and for the Alexa top 1 million domains (0.062%) and found that the vulnerability poses a serious security flaw that deserves more attention from domain owners and DNS service operators.

Certain limitations have to be taken into account to contextualize the obtained results. First, and perhaps foremost, we should note that our measurements establish a conservative lower bound for the magnitude of the problem. The servers that rely on address match lists to secure dynamic updates are counted as 'secure' in our measurement, but they are still vulnerable to IP spoofing. The attack requires only a single packet, making it possible for attackers to guess addresses that are on the match list.

The datasets in our study also present certain inherent limitations. For example, DNSDB has extensive, but not complete coverage of the domain name space. It also contains entries that are poisoned or obsolete, so many servers did not respond to our dynamic updates. Finally, we should note that responsibility is distributed and complicated. The fact that we found certain providers and software packages to be associated with vulnerable domains, should not be interpreted as assigning blame.

The next step for this work is to expand measurement and notify all affected parties, in order to improve the security of the DNS ecosystem, a critical service for many applications.

Acknowledgments

Authors thank Paul Vixie and Eric Ziegast from Farsight Security for sharing DNSDB, Jeroen van der Ham from the National Cyber Security Center (NCSC), Jelte Jansen, Moritz Müller and Marco Davids from SIDN, and the anonymous reviewers for their constructive and valuable comments. This work was supported by SIDN, the .NL Registry and by NWO (grant nr. 12.003/628 .001.003), NCSC. This work has been carried out in the framework of the project "IMATISSE" (Inundation Monitoring and Alarm Technology In a System of SystEms), funded by the Region Picardie, France, through the European Regional Development Fund (ERDF).

Chapter 3

The Closed Resolver Project: Measuring the Deployment of Inbound Source Address Validation

Coauthors: Yevheniya Nosyk, Maciej Korczyński, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda

3.1 Introduction

The Internet relies on IP packets to enable communication between hosts with the destination and source addresses specified in packet headers. However, there is no packet-level authentication mechanism to ensure that the source address has not been altered [102]. The modification of a source IP address is referred to as "IP spoofing". It results in the anonymity of the sender and prevents a packet from being traced to its origin. Reflection-based Distributed Denial-of-Service (DDoS) attacks leverage this mechanism and become even more effective using amplification [103–106]. As it is not possible in general to prevent packet header modification, concerted efforts have been undertaken to prevent spoofed packets from reaching potential victims. Filtering packets at the network edge formalized in RFC 2827 and called *Source Address Validation* (SAV) [28, 107] can achieve this goal.

Given the prevalent role of IP spoofing in cyberattacks, there is a need to estimate the level of SAV deployment by network providers. Projects such as Spoofer [108] already enumerate networks that do not implement packet filtering. However, a great majority of this existing work concentrates on *outbound* SAV since it can prevent reflection-based DDoS attacks near their origin [104]. While less obvious, the lack of *inbound* filtering enables an external attacker to masquerade as an internal host of a network, which may reveal valuable information about the network infrastructure usually not seen from the outside. Inbound IP spoofing can serve as a vector for zone poisoning attacks [109] that may lead to domain hijacking or cache poisoning attacks [13] even if the Domain Name System (DNS) resolver is correctly configured as a closed resolver. A closed resolver only accepts DNS queries from known clients and does so by matching the source IP address of a query against a list of allowed addresses.

The lack of SAV for inbound traffic can also have devastating consequences when combined with the DNS Unchained [105] or the NXDOMAIN attack (also known as the Water Torture Attack) [110], or the recently discovered NXNSAttack [111]. These attacks result in Denial-of-Service against both recursive resolvers and authoritative servers with a maximum packet amplification factor of 1,620 for the NXNSAttack [111]. IP spoofing is not required for this attack to succeed because any client can attack a resolver if it is allowed to query it. However, IP spoofing can greatly increase the number of affected resolvers by allowing an external attacker to target closed DNS resolvers: the attacker simply needs to masquerade as a legitimate client by spoofing its source IP address. Deploying inbound SAV at the edge of a network is an effective way of protecting closed DNS resolvers from this type of external attacks.

In this chapter, we present the results of the Closed Resolver Project [112]. The goal is to enumerate networks vulnerable to inbound spoofing Internet-wide as the first step in estimating the scale of the problem. We extend our previous work [113] and make the following main contributions:

(1) We exhaustively enumerate networks that do not deploy inbound SAV for IPv4. We propose a new method to identify networks that do not filter inbound traffic by using spoofed IP addresses. We perform Internet-wide scans of all BGP prefixes maintained by RouteViews [114] for the entire IPv4 address space to identify closed and open DNS resolvers in each routable network of the Internet. We send a DNS request of type `A` to each routable IP address (target address) in a packet

with a spoofed source IP address: when sending the request to X, we choose $X + 1$ as the source IP address. If there is no filtering in either transit networks or at the network edge, the target will receive our request. If it is a DNS resolver and our spoofed address matches the list of allowed clients, the resolver will resolve our request. As we spoof the source IP address, the response from the resolver is not routed back to our scanner, preventing us from analyzing it. However, we control the authoritative name server for the queried domains and we can observe queries sent by the resolver under test, either directly or through a chain of forwarding resolvers. Overall, this method identifies networks that do not correctly filter *incoming packets* without the need for a vantage point inside the network itself. The only requirement is that the network contains a DNS resolver (possibly closed).

(2) We enumerate IPv6 networks not deploying inbound SAV. IPv6 adoption gradually increases [115] so, the IPv6 Internet is becoming an attractive attack vector partly due to network operators not protecting the IPv6 portion of their networks as well as IPv4 [116]. Given the number of available addresses, a complete scan of the IPv6 address space (as explained previously for IPv4) is not computationally feasible. Instead, there are other ways to discover active IPv6 hosts, for example, through DNS zone transfers [117, 118]. One source of responsive addresses is the IPv6 Hitlist Service [119] that we use in this study. To enrich this list, we also deploy a two-level DNS zone infrastructure that forces resolvers to use both IPv4 and IPv6 to resolve our domain names, thus discovering IPv6 resolvers as a by-product of the IPv4 scan. Then, we perform a scan of the enumerated IPv6 addresses using the same method as for IPv4.

(3) We enumerate IPv4 and IPv6 networks deploying inbound SAV. The above technique, when applied alone, can reveal the *absence* of inbound SAV at the network edge. However, we would also like to confirm the *presence* of inbound SAV (possibly in transit). To achieve this goal, we send unspoofed DNS queries and identify 5.3 M IPv4 and 15.9 K IPv6 open resolvers. If open resolvers reply to the unspoofed requests but not to the spoofed ones, we can infer the presence of SAV for incoming traffic either at the network edge or in transit networks. By using the two methods, we can detect both the absence and the presence of inbound SAV either at the network edge or in transit.

(4) We combine different methods to check SAV compliance in both di-

rections. We collect the latest Spoofer data (over 1 month) and use a method proposed by Mauch [120] to infer the absence and the presence of *outbound* SAV. In this way, we can study the SAV deployment policies per provider in both directions. Previous work demonstrated the difficulty in incentivizing providers to deploy filtering for outbound traffic as it benefits other networks and not the network doing the deployment [121]. This work shows that even though SAV for inbound traffic directly benefits the networks deploying it, it is less widely deployed than SAV for outbound traffic.

(5) We compare SAV deployment status over IPv4 and IPv6. We first perform the analysis at the individual host level by identifying potentially dual-stacked DNS resolvers. For every (IPv4, IPv6) address pair, we gather DNS-level information such as the `version.bind` and pointer (`PTR`) records to confirm that both addresses belong to the same host. We also use other general-purpose fingerprinting tools to identify services running on ports 22, 80, 123, 443, and 587. Hardware and software information about each pair provides evidence on whether the two addresses belong to the same host or not. As single dual-stack machines are likely to exhibit the configuration of the whole BGP prefix and an autonomous system [116], we then compare the filtering policies at the level of autonomous systems. As a result, we show that SAV is less often deployed for IPv6 than it is for IPv4, both at the autonomous system and individual host levels.

(6) We analyze the geographical distribution of resolvers and networks vulnerable to inbound spoofing. Identifying the countries that do not comply with the SAV standard is the first step in mitigating the issue with the possibility of contacting local Computer Security Incident Response Teams (CSIRTs).

The rest of the chapter is organized as follows. Section 3.2 provides background on Source Address Validation and Section 3.3 discusses related work. Section 3.4 introduces our methodology. Section 3.5 provides the main results and their analysis. Section 3.6 discusses the geographic location of vulnerable networks. Finally, Section 3.7 concludes the paper and gives some directions for future work.

3.2 Background

In 2000, RFC 2827 proposed Source Address Validation (SAV) as a means for mitigating a growing number of DDoS attacks. The proposed solution was to discard packets with

source addresses not following filtering rules. This operation is most effective when applied at the network edge [28]. RFC 3704 proposed different ways to implement SAV including static Access Control Lists (ACLs) and reverse path forwarding [122]. Packet filtering can be applied in two directions: *inbound* with respect to a customer (for packets coming from the outside to the customer network) and *outbound* from a customer (for packets coming from the customer network to the outside). The lack of SAV in any of these directions may result in different security threats.

Attackers may benefit from the absence of outbound SAV to launch DDoS attacks, in particular, amplification and reflection attacks: they use public services prone to amplification [103, 104] to which they send requests on behalf of their victims by spoofing their source IP addresses. The victim is then overloaded with the traffic coming from the services rather than from the attacker. In this scenario, the origin of the attack is not traceable. One of the most successful attacks against GitHub resulted in traffic of 1.35 Tb/s: attackers redirected Memcached responses by spoofing their source addresses [123]. In such scenarios, spoofed source addresses are usually random globally routable IPs. In some cases, to impersonate an internal host, a spoofed IP address may be chosen to match a legitimate IP address of the target network, which may reveal the absence of inbound SAV [122].

Pretending to be an internal host reveals information about the inner network structure such as the presence of closed DNS resolvers that only accept queries from clients within the same network. Attackers can further exploit closed resolvers, for instance, to leverage misconfigurations of the Sender Policy Framework (SPF) [124]. In case of an incorrectly deployed SPF configuration, attackers can trigger closed DNS resolvers to perform an unlimited number of requests on behalf of mail servers, thus introducing a potential DoS attack vector. Combined with inbound spoofing, mail servers, not otherwise reachable from the outside, can also be exploited.

The absence of SAV for inbound traffic may also have serious consequences when combined with the DNS Unchained attack [105], the NXDOMAIN attack (also known as the Water Torture Attack) [110] or the recently discovered NXNSAttack [111]. These attacks result in Denial-of-Service against both recursive resolvers and authoritative servers. The NXNSAttack exploits the way recursive resolvers deal with referral responses (domain delegations) that provide the mapping between a given domain name and its authoritative nameserver without a glue record, i.e., the IP addresses of the

nameserver. The maximum packet DDoS amplification factor of the NXNSAttack attains 1,620 [111]. It also saturates the cache of the resolver, even the closed one, if the attack uses IP spoofing and inbound SAV is not in place.

The possibility of impersonating a host on the victim network can also assist in the zone poisoning attack [109]. A master DNS server, authoritative for a given domain, may be configured to accept non-secure DNS dynamic updates from a DHCP server on the same network [68]. Thus, sending a spoofed update from the outside with an IP address of that DHCP server will modify the content of the zone file [109]. The attack may lead to domain hijacking. Another way to target closed resolvers is to perform DNS cache poisoning [13]. An attacker can send a spoofed DNS request for a specific domain to a closed resolver, followed by forged replies before the arrival of the response from the genuine authoritative server. In this case, the users who query the same domain will be redirected to where the attacker specified until the forged DNS entry reaches its Time To Live (TTL).

Inbound IP spoofing is not limited to DNS-based attacks and can be combined with other vulnerable protocols (e.g., NTP, SNMP, SSDP [103], FTP, HTTP, Telnet [125], etc.) to launch self-directed amplification DDoS or attacks against other hosts in the same network. For example, NTP is known for its high amplification rates up to 4,670. An attacker sending spoofed requests on behalf of the victim trusted by private NTP servers can generate a huge amount of traffic towards the victim in the same network [103, 104].

Despite the knowledge of these attack scenarios and the costs of the damage they may incur, SAV is not yet widely deployed. Lichtblau et al. surveyed 84 network operators to learn whether they deployed SAV and what challenges they faced [126]. The reasons for not performing packet filtering included incidentally filtering out legitimate traffic, equipment limitations, and lack of a direct economic benefit—in case of outbound SAV, a compliant network cannot become an attack source, but it can still be attacked itself, which creates few incentives to become compliant. By contrast, inbound SAV protects networks from direct threats as described above, and is thus beneficial from an economic perspective.

Table 3.1: Methods to infer deployment of Source Address Validation

Method	Direction	Presence/ Absence	Remote	Relies on misconfigurations
Spoofer [108, 121, 127]	both	both	no	no
Forwarder-based [30, 104]	outbound	absence	yes	yes
Traceroute loops [128]	outbound	absence	yes	yes
Passive detection [126]	outbound	both	no	no
Spoofer-IX [129]	outbound	both	no	no
Our method [112]	inbound	both	yes	no

3.3 Related Work

3.3.1 Source Address Validation

Table 3.1 summarizes several methods proposed to infer SAV deployment. They differ in terms of the filtering direction (inbound/outbound), whether they infer the presence or absence of SAV, whether measurements can be done remotely or on a vantage point inside the tested network, and if the method relies on existing network misconfigurations.

The Spoofer project [108, 121, 127] deploys a client-server infrastructure mainly based on volunteers (and "crowdworkers" hired for one study trough five crowdsourcing platforms [130]) that run the client software from inside a network. To test outbound SAV compliance, the active probing client sends both unspoofed and spoofed packets to the Spoofer server either periodically or when it detects a new network. The server inspects received packets (if any) and analyzes whether filtering disables spoofing and to what extent [102]. For each client running the software, Spoofer identifies its /24 IPv4 address block (or /40 for IPv6) and the autonomous system number (ASN). It makes the results publicly available.[1] Testing inbound SAV compliance operates in the opposite direction—the Spoofer server sends packets to the client with spoofed source addresses belonging to the client network. However, the authors do not make the results public to protect vulnerable networks. This approach identifies the absence and the presence of SAV in both directions. The results obtained by the Spoofer project provide the most confident picture of the deployment of outbound SAV and have covered tests from 8,779 ASes since 2015. However, the network administrators not aware of the spoofing issue or those who do not deploy SAV are less likely to run Spoofer in their networks.

[1] https://spoofer.caida.org/summary.php

A more practical approach is to perform such measurements remotely. Kührer et al. [104] scanned for open DNS resolvers, as proposed by Mauch [30] to detect the absence of outbound SAV. They leveraged misconfigured forwarding resolvers that forward a request to a recursive resolver with either i) the packet source address not changed to its own address or ii) the response to the client sent with the source IP of the recursive resolver [104,131]. They fingerprinted those forwarders and found out that they were mostly embedded devices and routers. Misconfigured forwarders originated from 2,692 autonomous systems. We refer to this technique as *forwarder-based*.

Lone et al. [128] proposed another method that does not require a vantage point inside a tested network. When packets are sent to a customer network with a routable but not allocated address, it is sent back to the provider router without changing its source IP address. The packet, having the source IP address of the machine that sent it, should be dropped by the router because the source IP does not belong to the customer network. The method detected 703 autonomous systems not deploying outbound SAV.

While the above-mentioned methods rely on actively generated (whether spoofed or not) packets, Lichtblau et al. [126] passively observed and analyzed inter-domain traffic exchanged between more than 700 networks at a large interconnection point (IXP). They classified observed traffic into bogon, unrouted, invalid, and valid based on the source IP addresses and AS paths. The most conservative estimation identified 393 networks that generated the invalid traffic. Müller et al. [129] developed Spoofer-IX, another methodology to detect spoofing at the IXP level. Their traffic classification took into account AS business relationships, asymmetric routing, and traffic engineering. Deployed at one mid-sized IXP during five weeks, it measured 40 Mb/s as the upper bound of spoofed traffic.

We are the first to propose a remote method (no vantage points are needed in the tested networks) to detect the absence of inbound SAV that does not rely on existing misconfigurations. Instead, we take advantage of the presence of local DNS resolvers in remote networks (both open and closed) to infer the absence of packet filtering or the presence of SAV either at transit networks or the edge.

3.3.2 Dual-Stack

To compare the SAV deployment status over IPv4 and IPv6, we identify seemingly dual-stacked DNS resolvers.

Several researchers used DNS to obtain candidate (IPv4, IPv6) address pairs that likely indicate to be the same physical machine (also called dual-stacked). Berger et al. [132] developed two passive and active techniques to find such pairs. They deployed the passive method over the existing production infrastructure consisting of a two-level authoritative nameserver hierarchy in which the first-level server, reachable over IPv4, returns records of the second-level server. In its DNS response, it also encodes the IPv4 address of the contacting client. Each request arriving at the second-level nameserver over IPv6 gives the initial IPv4 query. This method is not restricted to open resolvers and does not actively generate additional DNS requests. The method discovered 674 K candidate pairs during a period of six months. The second, active technique relies on sending requests to open resolvers for such multi-level domains, which imply switching between IPv4 and IPv6 protocols using `CNAME` records. In a one-day measurement session, they probed 200 times 7 K open resolvers and revealed 41 K address pairs.

Hendriks et al. [133] enumerated the population of open IPv6 resolvers to analyze whether they could be used as efficient DDoS amplifiers. They first performed an Internet-wide scan to find open resolvers over IPv4 and queried them for specifically-crafted domains that could only be reached by traversing from IPv4 to IPv6. This method discovered 1.49 M unique candidate pairs and 1,038 unique IPv6 resolvers.

The two approaches described above do not necessarily find candidate pairs that are single dual-stacked machines (also called siblings). There is a need to validate those results. Beverly et al. [134] proposed a technique not limited to DNS resolvers based on the collected TCP-level information such as option signatures and timestamps. The algorithm was 97% accurate in identifying sibling relationships. In 2017, Scheitle et al. [135] developed a machine-learning algorithm that also gathered various TCP-level features (options, timestamp clock frequency, timestamp value, clock offset, etc.) and calculated a variable clock skew. The precision of the algorithm exceeded 99%.

Czyz et al. [116] showed that the IPv6 Internet is more open than IPv4. They developed two candidate lists: router IP pairs and pairs derived from DNS zone files. They probed all addresses on various ports for services expected to run on routers and DNS servers. To ascertain that some pairs were indeed dual-stacked machines, they collected fingerprinting information of the following applications: HTTP, HTTPS, SNMP, NTP, SSH, and MySQL. Based on this information, they confirmed that 96%

of router and 97% of nameserver pairs, open on at least one of the ports, were the same physical machines.

To compare the SAV deployment status over IPv4 and IPv6, we have deployed a two-level hierarchical DNS zone infrastructure that forces a recursive resolver to switch from IPv4 to IPv6 (and vice versa) to resolve our domain names. Whenever we detect that an IPv4 or IPv6 resolver is also reachable over IPv6 and IPv4, respectively, we consider such address pairs to be dual-stack candidates. We send spoofed and non-spoofed packets to target both open and closed resolvers. We then fingerprint them on different ports to gather evidence on whether each pair belongs to the same physical machine.

3.4 Methodology

In this section, we present the methodology for identifying networks that do not correctly filter incoming packets.

3.4.1 IPv4 Spoofing Scan

The core idea of the spoofing scan is to send a hand-crafted DNS `A` record request packet with a spoofed source address to all IP addresses in a tested network. We have developed an efficient scanner[2] running on a machine in a network that does not deploy outbound SAV so that we can send packets with spoofed IP addresses. When a resolver inside a network vulnerable to inbound spoofing performs query resolution, we observe it on our authoritative DNS servers. To prevent caching and to identify the true originator in case of forwarding, we query every time the following unique domain name composed of: a random string, the hex-encoded resolver IP address (the destination of our query), a scan identifier, the IP version subdomain and the domain name itself. The encoded IP address lets us identify forwarders: if the IP address seen on our authoritative nameservers is not the same as originally queried (extracted from the domain name), we know that the query destination is a forwarder. An example domain name is `dklL56.01020305.s1.v4.drakkardnsv4.com`.

Figure 3.1 shows the scanning setup for the example `1.2.3.0/24` network. In Step ①, the scanner sends one spoofed packet to each potential host of the network (packets to

[2]We make the scanner available to the interested researchers upon request.

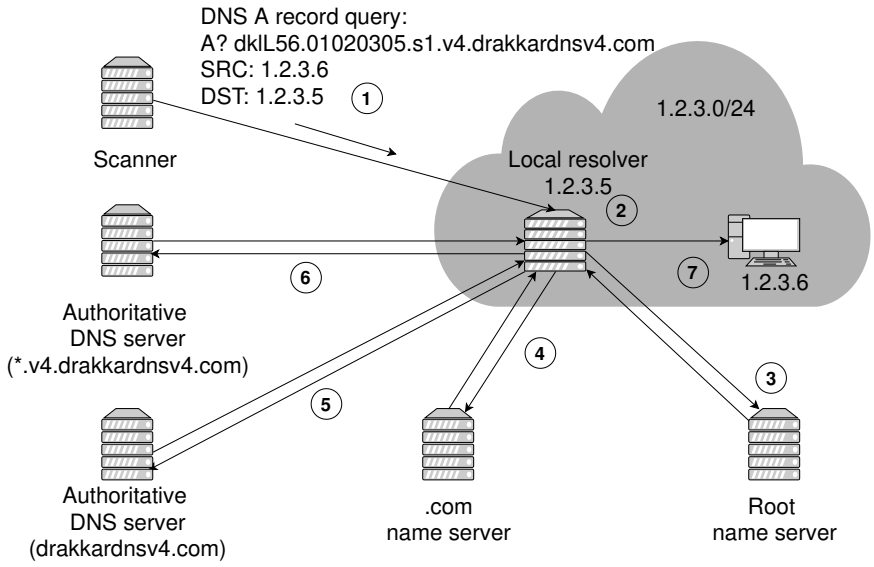


Figure 3.1: Setup of the spoofing scan over IPv4. We set up devices on the left-hand side (scanner, authoritative nameservers) and do not have control over the remaining infrastructure.

256 destinations in total). The spoofed source IP address is always the next one after the destination. When the scanner sends the spoofed packet containing the DNS query, there are four possible cases:

- **Packet filtering in transit network or random losses**. The spoofed packet can be filtered anywhere in transit or dropped due to reasons not related to IP spoofing such as network congestion [102].
- **Packet filtering (inbound SAV) in place**. When the spoofed DNS packet arrives at the destination network edge (therefore it has not been filtered anywhere in transit), the packet filter inspects the packet source address and detects that such a packet cannot arrive from the outside because the address block is allocated inside the network. Thus, the filter drops the packet.
- **No packet filtering (inbound SAV) in place and no DNS resolver**. The packet enters the networks, but there is no local DNS resolver on the tested network, so the DNS query is not resolved. In some cases, the DNS resolver is present but may be configured to refuse queries coming from its local area network (for example, if the whole separate network is dedicated to the infrastructure), so the packet is also dropped.
- **No packet filtering (inbound SAV) in place and the destination host**

is a DNS resolver. The scanner eventually reaches all the hosts in the network and the local DNS resolver if there is one (`1.2.3.5` in Figure 3.1). When the local resolver receives a DNS A record request (Step ②) from a host on the same network (`1.2.3.6`), it performs query resolution (Steps ③–⑥) so that our authoritative DNS server receives the query and replies. The local resolver sends the response to the source address (Step ⑦), dropped by the destination.

Note that only the last case allows **inferring the absence of inbound SAV** and we cannot distinguish between the first three cases.

There are two types of resolvers: forwarders that forward queries to other recursive resolvers and non-forwarders that resolve queries they receive. Therefore, the non-forwarding local resolver (e.g., `1.2.3.5`) inspects the query that looks as if it was sent from `1.2.3.6` and performs the resolution by iteratively querying the root (Step ③) and the top-level domain name (Step 4) servers until it reaches our authoritative DNS servers in Steps ⑤ and ⑥. Alternatively, it forwards the query to another recursive resolver that repeats the same procedure as described above for non-forwarders. In Step ⑦, the DNS `A` query response is sent to the spoofed source (`1.2.3.6`).

Our goal is to scan the whole IPv4 address space, yet taking into account only globally routable and allocated address ranges. We use the data maintained by the RouteViews Project [114] to get all the IPv4 blocks currently present in the BGP routing tables and send spoofed DNS requests to all the hosts of the prefixes.

3.4.2 IPv6 Spoofing Scan

The complete scan of the IPv6 space is not possible, even considering only the networks present in the BGP routing tables. We use source IPv6 addresses discovered by dual-stack identification described in Section 3.4.4 and the addresses from the IPv6 Hitlist Service [119]. On the day of measurements, the IPv6 Hitlist Service contained 270 M addresses for scanning.

We send spoofed DNS `A` requests to all hosts from our hitlist and spoof the source to be the next IP address after the target. The format of the domain name is similar to the IPv4 one: `qGPDBe.long_int(ipv6).s1.v6. drakkardnsv6.com`. We represent the IPv6 address as a long integer to identify the initial query destination uniquely and to distinguish forwarders from non-forwarders. We still send requests for the DNS `A` record, as changing the network protocol does not influence the retrieved resource records.

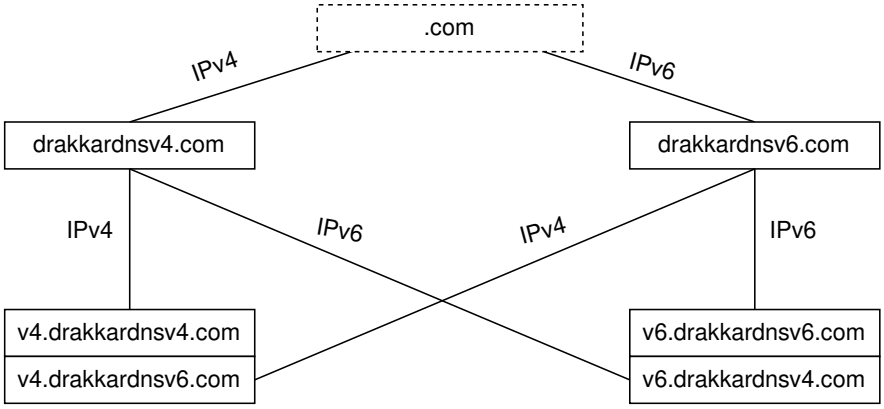


Figure 3.2: DNS zone setup. Rectangles with solid lines represent authoritative nameservers for the corresponding DNS zones (domain names) under our control. The `.com` zone (dashed) only contains glue records (IP addresses) of nameservers authoritative for our domains and is out of our control. Edges indicate the network protocol (IPv4 or IPv6) needed to reach a given zone.

3.4.3 Open Resolver Scan

In parallel to the spoofing scan, we perform an open resolver scan over IPv4 and IPv6 by sending DNS `A` requests with genuine source IP addresses of the scanner. To avoid temporal changes, we send a non-spoofed query just after the spoofed one to the same host. The format of a non-spoofed query is almost the same as the spoofed one. The only difference is the scan identifier (`n1` referring to a non-spoofed scan identifier instead of `s1`): `qGPDBe.02ae52c7.n1.v4.drakkardnsv4.com`, `qGPDBe.long_int(ipv6).n1.v6.drakkardnsv6.com`.

If we receive a non-spoofed request on our authoritative nameservers, it means that we have reached an open resolver. Moreover, if this open resolver did not resolve the spoofed query, we **infer the presence of inbound SAV** either in transit or at the tested network edge.

3.4.4 Identifying Dual-Stack Candidates

To compare the level of SAV deployment over IPv4 and IPv6 at the machine level, we need to collect (IPv4, IPv6) address pairs likely belonging to the same physical machine. We do so by deploying two-level DNS zones as shown on Figure 3.2. We set up zone files for two domains (`drakkardnsv4.com` and `drakkardnsv6.com`) on two distinct machines configured with both IPv4 and IPv6 addresses.

For each domain, we configure an authoritative name server with only one glue record (i.e., IP address of the name server) via the registrar control panel: the IPv4

address for `drakkardnsv4.com` and the IPv6 address for `drakkardnsv6.com`. For example, for `drakkardnsv4.com`, we configure the `ns1.drakkardnsv4.com` authoritative nameserver and the corresponding glue record (e.g., `ns1.drakkardnsv4.com A 5.6.7.8`). For `drakkardnsv6.com`, we set up the `ns1.drakkardnsv6.com` authoritative nameserver and the corresponding glue record (e.g., `ns1.drakkardnsv6.com AAAA 2001::6`). Thus, on the DNS level, each nameserver can only be reached over one network layer protocol (IPv4 or IPv6) but not over both.

Similarly, two more nameservers host child DNS zones: `v4.drakkardnsv4.com` and `v6.drakkardnsv6.com` also reachable over only IPv4 and IPv6, respectively. They are the domain names we use for IPv4 (Section 3.4.1) and IPv6 (Section 3.4.2) scans. We also add two more child zones: i) `v4.drakkardnsv6.com` with the `ns1.v4.drakkardnsv6.com` authoritative nameserver and the IPv4 glue record added to the parent `drakkardnsv6.com` zone and ii) `v6.drakkardnsv4.com` with the `ns1.v6.drakkardnsv4.com` authoritative nameserver and the IPv6 glue record added to the parent `drakkardnsv4.com` zone.

Figure 3.3 shows how a recursive resolver (on the left) resolves the following domain name: `qgPDBe.01020304.nf.s1.v6.drakkardnsv4.com`. We assume that it previously obtained the IPv4 address of the authoritative nameserver `ns1.drakkardnsv4.com` for the `drakkardnsv4.com` domain. It contacts the nameserver over IPv4 asking for the `A` record of the queried domain name. The `ns1.drakkardnsv4.com` nameserver cannot directly provide the answer. Instead, it points the resolver to the `ns1.v6.drakkardnsv4.com` nameserver that has only the configured `AAAA` glue record (the IPv6 address). The resolver now has to contact the nameserver only reachable via the IPv6 address. In our example, the resolver with the `2001::4` IPv6 address sends the DNS `A` query to `2001::8` and finally, receives the response on its IPv6 address.

As explained in Section 3.4.1, we deal with two types of DNS resolvers: forwarders and non-forwarders. Forwarders are likely to be a part of a complex DNS infrastructure, not visible from our authoritative nameservers, which includes, but is not limited to, load balancing and DNS cache sharing [116]. Thus, non-forwarders are good candidates for dual-stack testing. Even if IPv4 non-forwarders may forward IPv6 requests (or the other way around), we consider them better sibling candidates.

During the spoofing scan (IPv4 or IPv6), we continuously process traffic captures from our nameservers. It is crucial to do it on-the-fly to avoid temporal changes such as IP address churn [136]. When we find non-forwarders, we send them requests with such

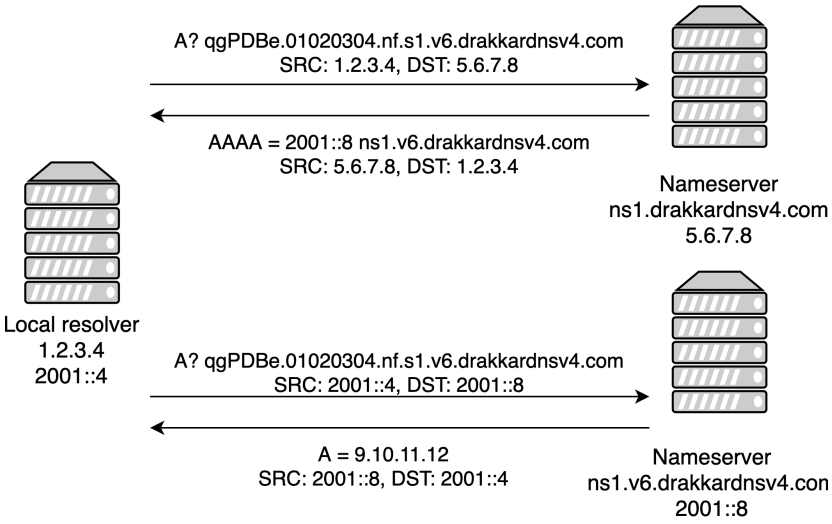


Figure 3.3: Domain name resolution that requires switching from IPv4 to IPv6. The local resolver on the left-hand side contacts the `ns1.drakkardnsv4.com` nameserver over IPv4. It does not receive the answer to the `A` request directly, but rather a referral to the `ns1.v6.drakkardnsv4.com` nameserver only reachable over IPv6.

domains that imply switching to the other version of IP. The domain name formats for IPv4 and IPv6 non-forwarders are: `qgPDBe.02ae52c7.nf.s1.v6.drakkardnsv4.com` `qgPDBe.long_int(ipv6).nf.s1.v4.drakkardnsv6.com`. We also send similar queries to the remaining IPv4 resolvers (forwarders and sources of queries), but exclude the `nf` part from the domain name. In this way, apart from identifying dual-stack candidates, we learn more IPv6 addresses in addition to the IPv6 Hitlist Service [119].

The second round of the capture analysis yields the requests containing the presented domain names. We retrieve the source IP and the domain-encoded address from non-forwarding requests to form an (IPv4, IPv6) sibling candidate pair. We only use the requests coming from forwarding IPv4 resolvers to reveal IPv6 addresses then scanned as described in Section 3.4.2.

3.4.5 Fingerprinting

We have performed a preliminary measurement campaign and gathered 1 K candidate pairs. We have scanned all the addresses with `nmap`[3] for 1 K most common ports [137]. Our candidates had open services on ports: 22 (SSH), 53 (DNS), 80 (HTTP), 443 (HTTPS), and 587 (SMTP). We also scanned for port 123 (NTP), as NTP is a protocol commonly used to amplify DDoS attacks [103] [104], and found that more than 10%

[3] https://nmap.org/

of addresses had port 123 open. Open ports may reveal the running software version, underlying operating system, and other information, such as public keys and certificates. However, we consider the fraction of the remaining open ports negligible and not suitable for fingerprinting. Thus, we have deployed the following technique to gather the evidence whether the (IPv4, IPv6) sibling candidate pair belongs to the same physical machine.

DNS: A pointer (`PTR`) resource record (or reverse DNS record) is the mapping between an IP address and a domain name. It is a recommended practice to have a hostname configured for every IP address [138]. Nevertheless, it was shown that only 1.2 billion responsive IPv4 addresses (28.17% of the whole IPv4 space) have an associated `PTR` record [139]. We perform reverse DNS lookups for a given (IPv4, IPv6) sibling candidate pair and check for an exact match between returned domain names as it is common for shared host names to represent a single machine [116]. Moreover, we query the sibling candidate pair for the domain name `version.bind` as a DNS `TXT` record in the `CHAOS` class [140]. Unless explicitly hidden, a DNS resolver replies with the exact installed software version. The example return values include "9.11.10-RedHat-9.11.10-1.fc29" or "unbound 1.10.0". We look for candidate pairs for which the same version is displayed for both. We ignore the cases when the arbitrary string is returned.

NTP: We fingerprint resolvers over UDP port 123 using the nmap scanner. The NTP standard [141] specifies a special packet header variable called `version` that reveals the running software. We retrieve it using the `ntp-info` Nmap Scripting Engine (NSE) [142], which not only returns the NTP server version but also the underlying system information [116].

SMTP: Port 587 is used for email submission by email clients and servers [143]. An extension to SMTP allows secure communication over the Transport Layer Security (TLS) protocol [144]. We use `openssl` tool[4] to initiate a connection and to obtain the server certificate.

HTTP: We use the ZGrab 2.0 application layer scanner[5] to get home pages, headers, and certificates for all the remaining protocols [145]. The software initiates a GET request to the potential web server over HTTP. In case of a successful connection, there may be an HTTP `Server` header field with the webserver software version that we retrieve, examine, and search for an exact match between IPv4 and IPv6 sibling

[4] https://www.openssl.org/
[5] https://github.com/zmap/zgrab2

candidate pair.

HTTPS: Web servers delivering content over the TLS protocol provide more information about the machine in addition to what we can learn with HTTP. The TLS specification [146] defines a handshake protocol between the client and the server. The server responds to the client request with the `ServerHello` message [147]. We retrieve the following parameters: `cipher_suite` and `server_version` (the TLS version chosen by the webserver based on what is proposed by the client). We also check the Certificate message for the returned certificate and `ServerKeyExchange` message for the actual used `tls_version` [116].

SSH: Machines open on port 22 provide us with the SSH software version, the server public key fingerprint, and the length of the key [116].

3.4.6 Network Granularity Levels for Evaluation

Each request received on our authoritative name server reveals the IP address of the original target of the query that we can associate with the longest matching BGP prefix and its ASN as it appears in the RouteViews data [114]. For a more fine-grained analysis, we consider /24 IPv4 and /40 IPv6 networks. This granularity leads to the evaluation of SAV deployment at different levels:

- Autonomous systems: the proposed method does not allow to determine if an entire AS is vulnerable to inbound IP spoofing. However, we can conclude that an AS contains at least one network that does not deploy inbound SAV. We compare SAV deployment for IPv4 and IPv6 as autonomous systems are known to contain both types of networks [148].

- Longest matching BGP prefixes: as the provider ASes may sub-allocate their address space to customers by prefix delegation [149], the longest matching prefix is another commonly used unit of analysis [121, 127].

- /24 (IPv4) and /40 (IPv6) networks: they are the smallest units for evaluating SAV deployment by the existing methods [108, 121].

- Individual hosts: dual-stacked resolvers may have different security policies in IPv4 and IPv6 parts and, consequently, different packet filtering rules [116].

Table 3.2: Types of discovered DNS resolvers

	# scanned hosts	Total DNS resolvers	Closed resolvers in networks without iSAV	Open resolvers in networks without iSAV	Open resolvers in networks with iSAV
IPv4	2,831,160,434	7,871,673	2,522,869	3,970,827	1,377,977
IPv6	270,703,379	115,610	99,718	8,977	6,915

3.4.7 Limitations

Our approach has some limitations that may impact the accuracy of the results. We rely on the main assumption—the presence of an (open or closed) DNS resolver or a forwarder in a tested network. If there is no DNS resolver, we cannot conclude on the filtering policies. If the probed resolver is closed, our method only concludes that the network does not perform SAV for inbound traffic, at least for some part of its IP address space. Only the presence of an open DNS resolver may reveal the inbound SAV presence assuming that the transit networks do not deploy SAV.

Transit networks with SAV may influence the measurement results by eliminating the possibility of detecting spoofing vulnerability at the network under measurement: if some transit networks filter spoofed probes, this means that we will not detect some target networks not deploying inbound SAV (if only closed resolvers are present) or we will incorrectly detect some networks as deploying inbound SAV (if open resolvers are present). However, if our spoofed probes do arrive in the target network, we can detect the absence of inbound SAV. In this sense, our results are optimistic—in an ideal measurement setup without SAV in transit networks, we could detect a larger number of networks vulnerable to inbound spoofing.

Some other reasons may also explain the absence of data for certain IP addresses: packet losses or temporary network failures.

3.4.8 Ethical Considerations

To make sure that our study follows the ethical rules of network scanning, yet providing complete results, we have adopted the recommended best practices [89, 150]. For the IPv4 scan, we aggregate the BGP routing table to eliminate overlapping prefixes. In this way, we send no more than two DNS `A` request packets (spoofed and non-spoofed ones) to every tested host. Due to packet losses, we potentially miss some results, but we accept this limitation not to disrupt the normal operation of tested networks. In

Table 3.3: Deployment of inbound SAV

Network Type	Consistent absence of inbound SAV		Partial absence of inbound SAV		Consistent presence of inbound SAV		No data		Total
	Count	Ratio (%)	Count	Ratio (%)	Count	Ratio (%)	Count	Ratio (%)	
IPv4 AS	21,314	31.8	11,441	17.1	2,092	3.1	32,131	48.0	66,978
IPv4 BGP prefixes	152,316	17.9	45,292	5.4	39,341	4.7	609,839	72.0	846,788
IPv4 /24 networks	765,233	6.9	173,239	1.5	266,498	2.4	9,948,051	89.2	11,153,021
IPv6 AS	4,639	24.8	127	0.7	138	0.7	13,806	73.8	18,710
IPv6 BGP prefixes	6,731	8.0	142	0.2	274	0.3	76,526	91.5	83,673
IPv6 /40 networks	7,562	0.02	136	0.0002	2,874	0.006	49,408,039	99.9	49,418,611

addition, we randomize our input list for the scanner so that we do not send consecutive requests to the same network (apart from two consecutive spoofed and non-spoofed packets). We spread our scanning activities over 15 days due to limited resources on the scanning machine (8 vCPUs and 3GB of RAM).

We have set up a website for this project on `closedresolver.com` and provided all the queried domains and the fingerprinting server with a description of our project as well as the contact information if someone wants to exclude her networks from testing. We have received 9 requests from operators of, among others, /8, /9 and /10 IPv4 networks, who noticed our DNS requests. In total, we excluded 29 M IPv4 addresses from our futures scans as well as two IPv6 prefixes (/128 and /48). We also exclude these addresses from our analysis. We do not publicly reveal SAV policies of individual networks and AS operators. Yet, website visitors can see the results for the network they connect from.

3.5 Inferring Presence and Absence of SAV

We have been performing spoofing and open resolver scans since July 2019. For the purpose of this study, we use data from the scan carried out in March 2020, using the methodology described in Section 3.4.

3.5.1 IPv4 Scan

For the IPv4 scan, we sent two DNS requests (one spoofed and one non-spoofed) to more than 2.8 billion hosts (we excluded 24 M addresses from the BGP table as a result of not-to-scan requests, see Section 3.4.8). We captured 10.9 M spoofed and 9.2 M non-spoofed `A` requests on our `ns1.v4.drakkardnsv4.com` authoritative DNS server. Previous work has shown that DNS resolvers tend to issue repetitive queries due to proactive caching or premature querying [151]. Thus, we leave unique tuples of the source IP

address and the domain name, which results in 8.7 M spoofed and 7.5 M non-spoofed unique requests.

Each `A` request contains a domain name with hexadecimally encoded IP address of the original query destination corresponding to a DNS resolver. Table 3.2 presents the types of DNS resolvers with the IP addresses extracted from the domain names observed on our authoritative nameserver. In total, we identified 7.9 M unique DNS resolvers. Spoofed queries revealed 6.5 M resolvers (2.5 M closed and 4 M open) located inside networks without inbound SAV. 5.3 M open resolvers responded to non-spoofed queries: 4 M in networks without and 1.4 M in networks with inbound SAV in place as they dropped the spoofed queries and only resolved the non-spoofed ones.

3.5.2 IPv6 Scan

The IPv6 scan immediately followed the IPv4 experiment. We analyze all the spoofed/non-spoofed `A` requests received on our `ns1.v6.drakkardnsv6.com` authoritative name server. Our target list is composed of 270 M IPv6 addresses leveraged from the IPv6 Hitlist Service and our dual-stack scan by traversing from IPv4 to IPv6-only zones as discussed in Section 3.4.4. On our authoritative nameserver, we received 290 K `A` requests related to our initial spoofed DNS queries and 40 K to non-spoofed queries. After filtering out the duplicate queries, we get 120 K and 23 K unique queries respectively. Importantly, 62 K resolvers were discovered by traversing from IPv4 to IPv6, 76 K from IPv6 hitlist, whereas 22 K appeared in both groups. The results highlight the added value of the method to identify IPv6 addresses by sending spoofed requests to dual-stack resolvers as explained in Section 3.4.4.

Table 3.2 presents the types of IPv6 resolvers. The great majority of 116 K unique IPv6 DNS resolvers are closed (100 K) and without the proposed spoofing discovery technique, they are not detectable. Similarly to IPv4, most of the open resolvers come from networks without inbound SAV in place.

3.5.3 Deployment of Inbound SAV

For each discovered DNS resolver, we associate its IP address with the corresponding /24 IPv4 (/40 IPv6) network, BGP routing prefix, and the autonomous system number using `pyasn`.[6] Note that multiple resolvers may belong to a single network/prefix/AS.

[6] https://pypi.org/project/pyasn/

We define three types of networks/prefixes/ASes with respect to the deployment of inbound SAV—they can be characterized by:

- **Consistent absence of inbound SAV**: all the discovered DNS resolvers inside a single network/prefix/AS indicate the absence of inbound SAV.
- **Partial absence of inbound SAV**: some resolvers indicate the absence while the others indicate the presence of inbound SAV.
- **Consistent presence of inbound SAV**: all the discovered DNS resolvers indicate the presence of inbound SAV at the edge of the network under measurement or filtering in transit.

As highlighted before, with the proposed method, we cannot unambiguously ascertain whether an entire network/prefix/AS is vulnerable to inbound IP spoofing. However, when reporting the deployment of inbound SAV, we refer to the results of our measurements, i.e., whether they consistently or partially indicate the absence or presence of inbound SAV.

Table 3.3 presents the inferred state of the inbound SAV deployment at different network levels in the IPv4 and IPv6 address spaces. We measured the filtering policies of 52% of IPv4 (26.2% of IPv6) autonomous systems, 28% of IPv4 (8.5% of IPv6) BGP routing prefixes, and 10.8% of IPv4 /24 (0.1% of IPv6 /40) networks. The coverage of the IPv4 address space is naturally bigger than that of IPv6 as we scanned the whole routable IPv4 address space.

Our measurements indicate that very few of covered networks consistently implement inbound SAV[7] (3.1% of IPv4 and 0.7% of IPv6 ASes, 2.4% of IPv4 /24 and 0.006% of IPv6 /40 networks) and are thus protected from spoofing attacks. Moreover, our measurements reveal that as many as 48.9% of IPv4 ASes (out of 52% of ASes for which we collected data) and 25.5% of IPv6 ASes (out of 26.2% of ASes for which we collected data) are consistently or partially vulnerable to inbound spoofing. Most of networks for which we obtained measurements show consistent or partial absence of inbound SAV, whether in the IPv4 (78% for /24 networks) or in the IPv6 (73% for /40 networks) address spaces.

The obtained results set a lower bound on the number of networks suffering from partial or complete lack of inbound SAV as we do not have measurement data for 48%

[7]Note that the number of networks deploying inbound SAV include the cases of filtering in transit.

of IPv4 and 73.8% of IPv6 ASes. Note that the fact that we do not have measurements for a given network does not necessarily mean that there are no resolvers in that network. It could also mean that the network contains only closed resolvers and spoofed packets that could reach them are filtered in transit networks. If we presume a uniform distribution of our measurements, by extrapolating these numbers for the entire IP address space, we obtain over 94% of IPv4 ASes and 97% of IPv6 ASes with consistent or partial absence of inbound SAV.

The results also set an upper bound for the networks consistently implementing inbound SAV. As the reported numbers include the cases of filtering in transit, the number of networks deploying inbound SAV is actually lower.

3.5.4 Impact of Network Characteristics on SAV Policies

Multiple factors may influence the decision of operators to deploy filtering in their networks.

We approached thirty providers among personal contacts for which we have measurements and asked their motivation (not) to perform packet filtering. We got replies from three operators of networks with partial deployment of inbound SAV. One /24 IPv4 network is logically divided into two parts. Some IP addresses belong to virtual machines and their OpenStack configuration provides inbound and outbound SAV, while others are physical servers or Internet access subscribers that do not deploy inbound SAV due to complexity, time, and financial issues. Another network administrator confirmed being responsible only for a subset of the /24 IPv4 network, thus having no control over the other part. Indeed, upstream providers perform route aggregation of smaller customer networks, maintained by different organizations [128] that possibly implement different anti-spoofing policies. Finally, one network operator reported that the whole /24 network had no inbound SAV, so we must have encountered packet losses.

One of the factors correlated with the deployment of inbound filtering policies of an AS or a network is the ***size of its IP address space***. Previous work has shown that the size plays an important role in the implementation of SAV for *outbound* spoofing: it is unlikely that smaller organizations have the resources and incentives to implement *outbound* SAV in their networks [128, 131]. Operators with a larger address space are more likely to adhere to best current practices and promote routing security (e.g., MANRS (Mutually Agreed Norms for Routing Security regulations) [152] members).

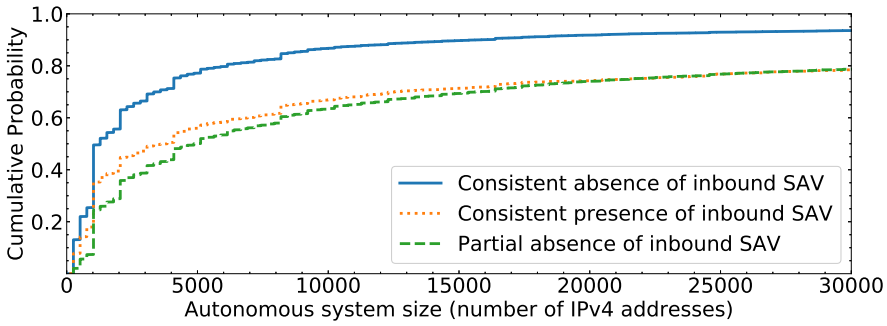


Figure 3.4: Sizes of IPv4 ASes computed based on the number of unique IPv4 addresses present in the BGP routing table. The cumulative probability shows that ASes with consistent absence of inbound SAV tend to be smaller than other ASes.

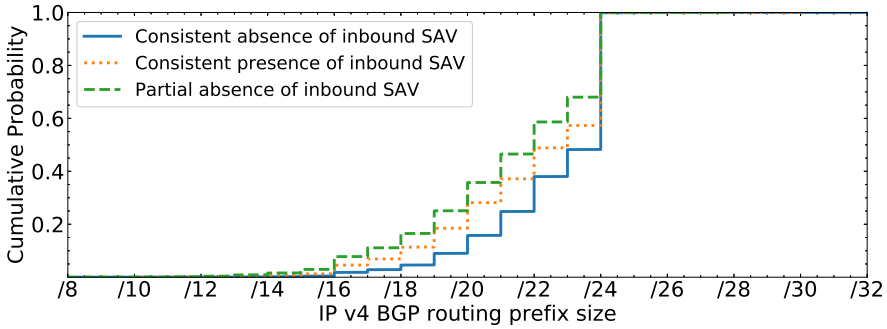


Figure 3.5: Sizes of the IPv4 longest matching prefixes from the BGP routing table. Larger prefixes are more likely to suffer from partial absence of inbound SAV.

To be compliant (consistently or at least partially), they would have to implement SAV on edge routers [131].

Figure 3.4 presents the cumulative distribution of the IPv4 AS sizes (the number of announced IPv4 addresses in the BGP routing table) with **i) consistent absence**, **ii) partial absence**, and **iii) consistent presence of inbound SAV**. For example, as many as 75.3% of consistently vulnerable ASes have 4,096 and fewer IP addresses. For comparison, 54.2% of consistently non-vulnerable ASes and 48.1% of partially vulnerable ASes have 4,096 addresses and less. Therefore, the size distribution of ASes with consistent absence of inbound SAV is driven by smaller ASes as compared to ASes with consistent presence and partial absence (presence) of inbound SAV. We observe similar trends for the longest matching BGP prefix sizes (see Figures 3.5 and 3.6). Our results show similar trends to those found in previous work exploring the relationship between the size and deployment of SAV for outbound spoofing. A possible explanation for the observed distributions is that the operators of smaller networks have generally fewer resources and lower competence to implement SAV in both directions.

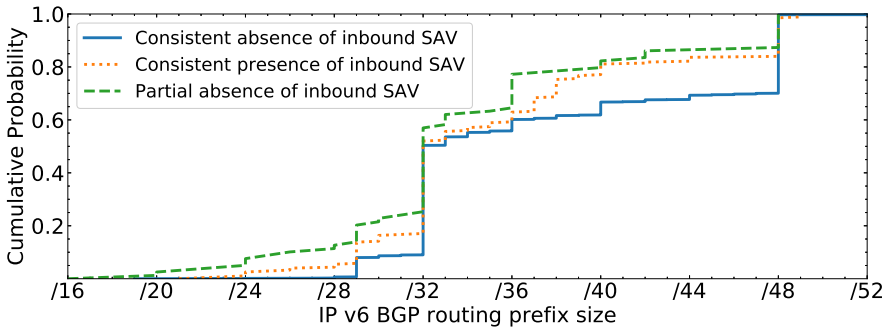


Figure 3.6: Sizes of the IPv6 longest matching prefixes from the BGP routing table. Prefixes with consistent absence of inbound SAV tend to be the smallest.

We also analyze ***AS stability*** in the IPv4 space as one of the factors that may influence the decision of operators to deploy SAV. If BGP advertisements frequently change, implementing ACL-based source address filtering becomes more challenging. We define AS stability as the percentage of prefixes that remain the same compared to all announced prefixes in September 2019–March 2020 based on weekly BGP announcements [114]. We find that 87% of ASes with consistent presence and 86% of ASes with consistent absence of inbound SAV advertise exactly the same prefixes, while less ASes (81%) that partially deploy inbound SAV are stable. The difference in stability between ASes with the partial absence, consistent presence, and consistent absence of inbound SAV is not significant. However, the marginal differences do suggest that ASes with partial absence of inbound SAV are less stable in advertising their IP space and it miwght be cumbersome to maintain ACL-based filtering.

Another factor in the deployment of SAV is asymmetric routing, particularly for multi-homed networks. It is important to note that strict filtering policies apply to so-called single-homed stub ASes that connect to their sole transit provider ASes [122]. The problem with non-stub or transit providers is that they might have customer ASes that do not announce all routes to them due to load balancing or fault tolerance [122,153]. It is less of an issue for inbound spoofing since an AS announcing the prefixes would know its own IP space. However, if the customer AS has more dynamic policies to announce prefixes, they may result in inconsistent filtering policies. Therefore, we define another factor correlated with SAV deployment—***the type of AS: stub or non-stub***. In the analysis, we use the Caida AS relationship data for IPv4 addresses [154]. We find that 95% and 90% of ASes with consistent absence and presence of inbound SAV, respectively, are stub ASes. We observe that less ASes (77%) with partial absence of

inbound SAV are stubs. We see that ASes with partial deployment have the largest ratio (23%) of non-stub ASes than those with consistent presence/absence of inbound SAV. It is likely that due to peering relationships, non-stub ASes might not know the IP space of downstream AS and hence cannot deploy SAV.

Finally, we consider the number of interconnections with other ASes, or ***the number of edge routers*** as another factor correlated with SAV deployment. We used the Caida bdrmapIT dataset to determine the ownership of the routers in ASes [155]. Their methodology uses traceroute from multiple vantage points, performs alias resolution for the routers, and infers AS relationship to determine the boundaries of ASes. We aggregate the number of border routers that link to other ASes for each AS in our dataset to estimate the number of its edge routers.

ASes use multiple links with upstream providers to avoid a single point of failure. To configure SAV, they would have to implement filtering policies on multiple routes near the exit routers. We observe that the average number of edge routers is around 15 (median 3) for ASes with the consistent presence of inbound SAV, while for the consistent absence, the average number is around 100 (median 5) and for ASes with partial absence of inbound SAV, it is around 200 (median 10). Please note the significant difference in the mean and median values for ASes with partial absence and consistent presence of inbound SAV, which shows that the distribution is skewed by a few ASes with a large number of edge routers.

We can conclude that ASes consistently vulnerable and non-vulnerable to inbound spoofing have similar network properties, different from ASes with partial absence of inbound SAV. The latter are more complex with comparatively larger sizes, with less AS stability, and with more non-stub ASes and edge routers. In the case of ASes with consistent absence of inbound SAV, there might be other socio-economic factors at play rather than the discussed network characteristics, since they are generally similar to ASes with deployed SAV.

3.5.5 Outbound versus Inbound SAV Policies

3.5.5.1 Network Level

To identify the most deployed type of SAV (inbound or outbound), we need to consider the networks for which we measure SAV compliance in both directions. We have already obtained the /24 IPv4 and /40 IPv6 networks with consistent absence and presence of

inbound SAV (we do not include here the networks with partial absence of inbound SAV). The goal is to find which of these networks comply to outbound SAV.

The first outbound SAV compliance dataset we use comes from the Spoofer Project. The Spoofer client sends packets with the IP address of the machine on which it is running as well as packets with a spoofed source address. The results are anonymized per /24 IPv4 and /40 IPv6 address blocks. Spoofer identifies four possible states: *blocked* (only an unspoofed packet was received, the spoofed packet was blocked), *rewritten* (the spoofed packet was received, but its source IP address was changed on the way), *unknown* (neither packet was received), *received* (the spoofed packet was received by the server).

In March 2020, we collected and aggregated the latest Spoofer data for one month. We obtained the tests for 3,731 /24 IPv4 and 579 /40 IPv6 networks (we only kept vulnerable to spoofing (*received*) and non-vulnerable to spoofing (*blocked*) networks). Note that these numbers are much smaller than 8,779 ASes tested by Spoofer since 2015. For the comparison in this section, we only choose the newest tests run around the same days as our inbound spoofing scan. As a result, the overlap between our inbound method and Spoofer is 473 /24 IPv4 and 17 /40 IPv6 networks. The minority of those have consistent filtering in both directions: 91 /24 IPv4 and 3 /40 IPv6 networks have no filtering in both directions while 77 /24 IPv4 and 2 /40 IPv6 networks implemented both inbound and outbound SAV. Interestingly, whenever filtering is deployed only in one direction, it is mostly outbound (59.4% for IPv4 and 70.6% for IPv6).

The next outbound SAV dataset comes from the forwarder-based measurement technique. We have deployed the method proposed by Mauch [30] to detect the absence of outbound SAV. We analyze the open resolver scan responses on the machine on which we run the scanner and we look for the cases in which the responses come from the IP address in different networks than the ones originally queried [104, 131].

We enumerated 446 K IPv4 and 5 IPv6 misbehaving forwarders, originating from 20 K /24 IPv4 and 4 /40 IPv6 vulnerable to outbound spoofing networks. The important limitation of the forwarder-based method is that it does not identify the presence of outbound SAV. The overlap with the inbound SAV dataset is 16 K IPv4 and 3 IPv6 networks. All the IPv6 networks had no SAV in both directions. For IPv4, there are 33.2% of networks with no SAV in both directions, whereas most of the IPv4 networks without outbound SAV (66.8%) deploy inbound SAV.

Table 3.4: Fingerprinting dual-stack candidate pairs

Protocol/ Application	Both closed	Only IPv4 open	Only IPv6 open	Both open	Same fingerprint
DNS (version.bind)	16,743	13,081	1,743	50,015	37,338 (45.8%)
DNS (PTR)	11,380	38,104	1,152	30,946	24,004 (29.4%)
NTP	67,009	2,034	2,498	10,041	128 (0.2%)
HTTP	27,406	15,986	3,292	34,898	34,218 (41.9%)
HTTPS	29,106	16,806	675	34,995	22,531 (22.6%)
SSH	33,825	2,055	2,442	43,260	5,622 (6.9%)
SMTP	47,597	10,140	653	23,192	23,060 (28.3%)
Total (unique)					61,313 (75.2%)

3.5.5.2 Autonomous System Level

We now analyze SAV policies for outbound and inbound traffic at the AS level. One of the most well-known initiatives to improve the security and resilience of the Internet global routing system is MANRS [152]. At the time of writing, 515 autonomous systems are its signatories. MANRS strongly encourages its members to implement SAV in their networks "to prevent packets with an incorrect source IP address from entering or leaving the network" [152]. However, recent work shows that MANRS members are not more likely to deploy SAV than the general population [121]. 81 MANRS ASes out of 515 are vulnerable to *outbound* spoofing shown by Spoofer and the forwarder-based datasets. However, as many as 311 ASes are at least partially vulnerable to *inbound* spoofing. Therefore, the results suggest that when network operators are familiar with the concept of SAV, they tend to secure traffic leaving their networks.

3.5.6 SAV Deployment for IPv4 and IPv6

As IPv6 deployment is growing, it becomes an attractive attack target. Individual dual-stacked machines and networks are generally more open on the IPv6 part [116]. In this section, we analyze whether dual-stacked networks are more vulnerable to inbound spoofing using IPv6. We do it at the individual host and AS levels.

3.5.6.1 Individual Host Level

We queried all non-forwarding IPv4 and IPv6 DNS resolvers (either open or closed) for a domain name requiring switching to IPv6 and IPv4, respectively. Out of 2.6 M IPv4 and 36 K IPv6 non-forwarders, 2.7% and 28.5% had also IPv6 and IPv4 connectivity, respectively, thus forming (IPv4, IPv6) candidate address pairs. Clearly, due to the

Table 3.5: Geolocation results

Rank	Resolvers (#)				Networks, vulnerable to inbound spoofing (#)				Proportion of networks, vulnerable to inbound spoofing (%)	
	Country	IPv4	Country	IPv6	Country	IPv4	Country	IPv6	Country	IPv4
1	China	1,970,410	USA	22,992	China	260,047	USA	1,319	Kosovo	63.6
2	Brazil	667,036	Germany	13,373	USA	162,259	Brazil	930	Comoros	52.6
3	USA	661,943	Netherlands	11,514	Russia	54,451	Germany	680	Western Sahara	50.0
4	Iran	404,134	Belarus	7,455	Italy	32,026	Netherlands	336	Armenia	49.5
5	India	348,491	Russia	6,410	Brazil	28,836	UK	309	Maldives	39.7
6	Algeria	249,931	China	5,840	Japan	27,890	China	304	Moldova	38.2
7	Russia	224,985	UK	5,151	India	27,426	Russia	289	Niue	37.5
8	Indonesia	222,602	Spain	3,996	Mexico	23,288	Czech Republic	254	Palestine	36.3
9	Italy	105,476	Czech Republic	3,357	UK	16,976	France	223	Afganistan	36.2
10	Argentina	104,850	France	2,837	Indonesia	16,798	Japan	183	Bulgaria	36.0

IPv6 adoption being far from universal [156–158], it is crucial for IPv6 resolvers to be reachable over IPv4 as well.

We collected 82 K candidate address pairs in total, most of them (72 K) during the IPv4 scan. DNS resolvers are known to have complex relationships and a single address can appear in multiple address pairs [132]. However, for our analysis, we consider each address pair separately.

We fingerprint each address in the pair as described in Section 3.4.5. Table 3.4 presents the results per address pair. Importantly, almost 98.1% of pairs had open ports for at least one fingerprinted protocol/application. The most largely open fingerprints are `version.bind` and SSH, which is consistent with the fact that we deal with DNS servers requiring remote access. While the NTP port is relatively largely open, we merely extracted the timestamp in most cases. Only 128 server pairs returned software and operating system versions. Among pairs with one or more open fingerprinted port, 75.2% have identical signatures in IPv4 and IPv6 for at least one fingerprinted application, which increases the confidence that they belong to the same physical machine. Two of the three network operators that responded to our survey operate dual-stacked resolvers and they confirmed the correctness of our mappings. In particular, 6 pairs had the same `PTR` record and 7 pairs had the same `version.bind` records (the remaining pairs had either no record at all or only a record for one address in the pair).

Most of the resolvers in the pairs show the absence or presence of SAV. However, there are cases in which we have discovered an IPv6 resolver through IPv4, sent a spoofed and a non-spoofed query, and did not get any results. We observe similar behavior in the opposite direction. From 61 K seemingly dual-stacked pairs, 43 K reveal the absence or presence of spoofing for IPv4 and IPv6. Most of them (99.2%) have consistent filtering policies. However, out of the remaining 324 hosts, 195 (60.2%) are vulnerable to inbound spoofing only over IPv6. Thus, at the individual host level, IPv6

tends to be slightly more vulnerable than IPv4.

3.5.6.2 Autonomous System Level

Whenever a certain security policy exists for an individual dual-stacked host, it is likely to hold for the whole autonomous system [116]. Consequently, we expect inbound SAV to be less deployed over IPv6 at the AS level as well. As of March 2020, there are 66,978 IPv4 and 18,710 IPv6 ASNs present in BGP routing tables. 18,016 of them advertise both IPv4 and IPv6 prefixes.

For this analysis, we choose vulnerable and non-vulnerable to inbound spoofing ASes and keep those having results for both IPv4 and IPv6. The resulting set includes 2,873 ASes. The great majority of them (94.2%) have consistent filtering policies for IPv4 and IPv6—2,650 are vulnerable and 55 are non-vulnerable to inbound spoofing. However, our results indicate that the remaining 168 ASes are not vulnerable to inbound spoofing over IPv4 (88.7% deployed inbound SAV) but are vulnerable over IPv6. Thus, at the AS level, SAV for inbound traffic is less deployed over IPv6.

3.6 Geographic Distribution

Identifying the countries that do not comply with the SAV standard is the first step in mitigating the issue by, for example, contacting local CSIRTs. We use the MaxMind database[8] to map every resolver IP address of the spoofed query retrieved from the domain name to its country. Table 3.5 summarizes the results.

In total, we identified 232 countries and territories vulnerable to inbound spoofing of incoming network traffic for either IPv4, IPv6, or both. We first compute the number of DNS resolvers per country. As explained in Section 3.5.2, the coverage of the IPv6 scan is smaller than that of IPv4, which is why we see less identified resolvers. Interestingly, only 3 countries are present in both IPv4 and IPv6 top 10 resolver ranking.

We now map the resolvers to the corresponding /24 IPv4 and /40 IPv6 address blocks to evaluate the number of vulnerable to inbound spoofing networks per country. We see that the top 10 countries by the number of DNS resolvers are not the same as the top 10 for vulnerable to inbound spoofing networks because a large number of individual DNS resolvers by itself does not indicate how they are distributed across

[8] https://dev.maxmind.com/geoip/geoip2/geolite2/

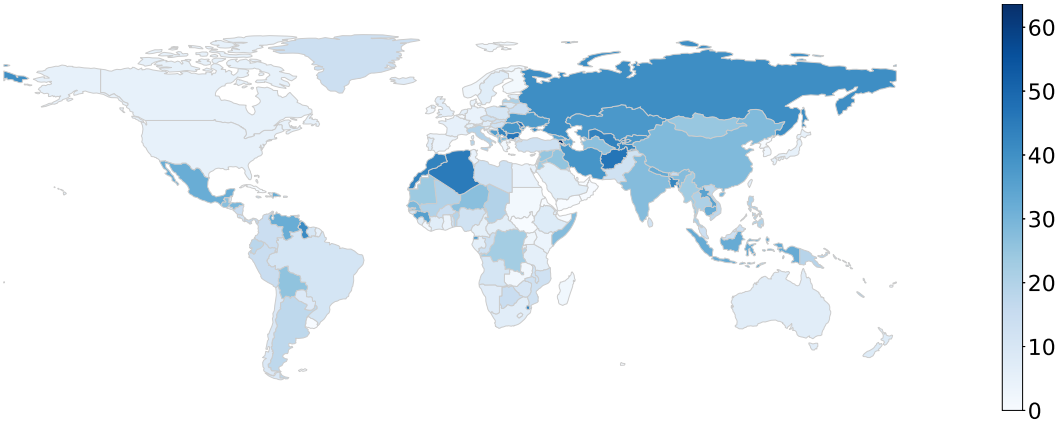


Figure 3.7: Fraction of vulnerable to inbound spoofing (inbound traffic) vs. all /24 IPv4 networks per country (in %)

different networks.

Such absolute numbers are still not representative as countries with a large Internet infrastructure may have many DNS resolvers and therefore reveal many vulnerable to inbound spoofing networks that represent a small proportion of the whole. For this reason, we compute the fraction of vulnerable to inbound spoofing vs. all /24 IPv4 networks per country. To determine the number of all the /24 networks per country, we map all the individual IPv4 addresses from the BGP routing table to their location, then to the /24 block, and keep the country/territory to which most addresses of a given network belong. Figure 3.7 presents the resulting world map. We can see in Table 3.5 that the top 10 ranking has changed. Small countries such as Western Sahara and Niue that have two and eight identified resolvers each, suffer from a high proportion of vulnerable to inbound spoofing networks. One of the two /24 networks of Western Sahara allows inbound spoofing. On the other hand, Bulgaria is a country with a large Internet infrastructure (16,439 /24 networks in total) and with a large percentage of vulnerable to inbound spoofing networks.

3.7 Conclusions

In this chapter, we have presented a novel method to infer the deployment of inbound SAV for the IPv4 and IPv6 address spaces. We have measured the filtering policies of 52% of routable IPv4 autonomous systems (26% for IPv6) and 28% of all the IPv4 BGP prefixes (almost 9% for IPv6). We show that the great majority of the networks for which we obtained measurements are consistently or partially vulnerable to inbound

spoofing.

Reflection and amplification DDoS attacks have extensively used open DNS resolvers in recent years. We have found 5.3 M IPv4 and 16 K IPv6 open resolvers. New ways to misuse open resolvers constantly emerge. NXNSAttack, one of the most-recently discovered attacks, can exploit open recursive resolvers to reach an amplification factor of up to 1,620. Even worse, inbound spoofing combined with the NXNSAttack results in additional 2.5 M closed resolvers for IPv4 (100 K for IPv6) either vulnerable themselves or possibly misused against other victims.

Open resolvers when they do not resolve spoofed queries identify the presence of inbound SAV at the edge of the tested network or filtering in transit. We found that while many providers deploy consistent filtering policies network-wide, there are cases when a single network is only partially protected from inbound spoofing. The results indicate that different network characteristics are factors that prevent operators from correctly configuring packet filtering. Overall, the proportion of non-vulnerable networks is much lower compared to networks with consistent or partial absence of inbound SAV.

We have identified and fingerprinted dual-stacked DNS resolvers and shown that at the individual host level, inbound filtering is slightly less deployed for IPv6 than for IPv4. This observation also holds for dual-stack autonomous systems, which is not surprising given that the IPv6 address space tends to be less secured than IPv4.

We have gathered different datasets to analyze whether outbound filtering is less deployed than inbound. Outbound SAV faces the problem of misaligned economic incentives—it protects other networks but not the one deploying it. Interestingly, SAV for outbound traffic turned out to be more deployed than inbound at the AS level among network operators committed to the MANRS initiative. The absence of outbound packet filtering gained widespread attention since it enables DDoS attacks. Under these circumstances, inbound SAV remains neglected (or overlooked) by network operators.

Vulnerability to inbound spoofing is not limited to any geographic territory and is spread worldwide. To draw attention to the problem of inbound spoofing, we launched the Closed Resolver Project at `https://closedresolver.com` in collaboration and funded by the Dutch CERT—National Cyber Security Centre (NCSC) and RIPE NCC.[9] Anyone can visit the website of the project and check whether his/her network is vulnerable

[9] `https://www.ripe.net/support/cpf/funding-recipients-2020`

to inbound spoofing and how many closed resolvers we found inside. The ultimate objective is to run notification campaigns for network operators and provide them with an accessible platform to investigate results for their networks. The service is particularly useful for operators planning to become a MANRS participant since MANRS strongly recommends deploying SAV. We expect these efforts will result in better packet filtering in the Internet.

Acknowledgments

This work has been carried out in the framework of the PrevDDoS project co-funded by RIPE Network Coordination Centre, the National Cyber Security Centre (NCSC) - the dutch CERT, and the IDEX Université Grenoble Alpes “Initiative de Recherche Scientifique (IRS)”.

Chapter 4

Adoption of Email Anti-Spoofing Schemes: Large Scale Analysis

Coauthors: Sourena Maroofi, Maciej Korczyński, Arnold Hölzel, and Andrzej Duda

4.1 Introduction

Email spoofing consists of sending a message with a forged sender address and other parts of the email header so that it appears as sent from a legitimate source. Attackers commonly use this method to mislead the receivers, gain their trust, and eventually, achieve some malicious goals. Phishing and spam campaigns are examples of attacks that rely on email spoofing. Despite tremendous efforts deployed to mitigate this technique, it is still one of the most successful attacks responsible for significant damage. According to the Internet crime report [159], email spoofing costed US victims more than 1.2 billion dollars in 2018.

Email spoofing comes in two types. The first one consists of *compromising legitimate servers* and using their mail transfer agent to send spoofed emails to victims either by specifying a different `Reply-to:` address or providing a phishing URL in the body of the message. The second type is *domain spoofing* in which attackers send emails on behalf of legitimate domains, e.g., a forged email from *account-security-noreply@accountprotection.microsoft.com* impersonating the Microsoft support team with a fake landing page looking alike a real Microsoft login page to steal user credentials [160]. In this paper, we investigate the second type of email spoofing.

The Simple Mail Transfer Protocol (SMTP) for email distribution does not provide support for preventing spoofing [21]. The system needs to rely on *security extensions* such as the Sender Policy Framework (SPF) [22], the DomainKeys Identified Mail (DKIM) [161], and Domain-based Message Authentication, Reporting, and Conformance (DMARC) [23] to authenticate the sender and decide what to do with suspicious emails. The extensions define a set of rules that specify who is allowed to send emails on behalf of a given domain name and how to deal with suspicious messages. A careful deployment of the extensions can completely mitigate the problem of domain spoofing. However, to be effective, both the domain owner and the mail transfer agent of the recipient should implement the extensions: the domain owner needs to correctly set SPF, DKIM, and DMARC rules, and the recipient has to authenticate incoming messages and correctly implement the verification of the SPF and DMARC rules.

In this chapter, we evaluate the extent of the SPF and DMARC deployment and analyze spoofing possibilities enabled by the absence or misconfiguration of their rules. We do not analyze DKIM as it requires access to the email header selector tag, not publicly available (see RFC 6376 for more details [161]).

While previous work already investigated the adoption of SPF and DMARC by the Alexa top-ranked one million domains [162,163], we consider different datasets as well as threat models. We scan approximately **236 million domain names** including generic top-level domains (gTLD), country-code TLDs (ccTLD) and new gTLDs collected from different sources such as the Centralized Zone Data Service (CZDS)[1] made available by the Internet Corporation for Assigned Names and Numbers (ICANN), OpenData project from Rapid7[2] as well as zone files that are public and available for download (e.g, .se). The second dataset includes **32,042 high-profile domains** of 139 countries and their **defensive domain registrations**. The high-profile domains correspond to most popular targets of email spoofing: well-known companies, governmental websites, or financial institutions. To the best of our knowledge, this is the first study reporting on the global-scale measurement of the adoption of email authentication extensions.

We investigate the global adoption of SPF and DMARC protocols by scanning each domain in our datasets. Then, we define a threat model in which attackers use subdomains (both existent and non-existent) for email spoofing. We also identify *defensively registered domains* and evaluate their adoption of email anti-spoofing schemes. We show

[1] https://czds.icann.org
[2] https://opendata.rapid7.com

that even if defensive registrations can mitigate some types of attacks like *cybersquatting* and *brand name abuse*, these domains need to be protected against domain spoofing as well.

We extend our previous work [164] and make the following main contributions:

1. we investigate the global adoption of SPF and DMARC for 236 million domain names of different TLDs,

2. in a separate measurement campaign, we evaluate the adoption of SPF and DMARC by top 500 most popular domains of 139 countries including local businesses, national websites, local governments, and financial sectors,

3. we propose a method to find defensively registered domains for top-ranked websites and assess the extent of their adoption of email security extensions,

4. we are the first to measure the extent of SPF and DMARC deployment by the subdomains of the top-ranked websites to gain better insight into how attackers can abuse subdomains to send spoofed emails,

5. we show that it is possible to send forged emails from non-existent subdomains when a DMARC rule is not strict enough regarding subdomains,

6. we demonstrate how syntactically wrong SPF rules may break the trust-based authentication system of selected email service providers by allowing forged emails to land in the user inbox,

7. we present a methodology for preventing domain spoofing based on good practices for managing SPF and DMARC records and analyzing DNS logs,

8. finally, as a proof of concept, we perform an end-to-end email spoofing for subdomains of high profile domains with misconfigured SPF and/or DMARC.

To remediate misconfigured SPF rules, we have contacted relevant Computer Security Incident Response Teams (CSIRTs) responsible for misconfigured domains and measured the effectiveness of our notifications. To encourage reproducibility, we make our measurement data available upon request.

The rest of the paper is organized as follows. Section 4.2 provides background on SPF and DMARC. Section 4.3 specifies possible threat models and introduces our

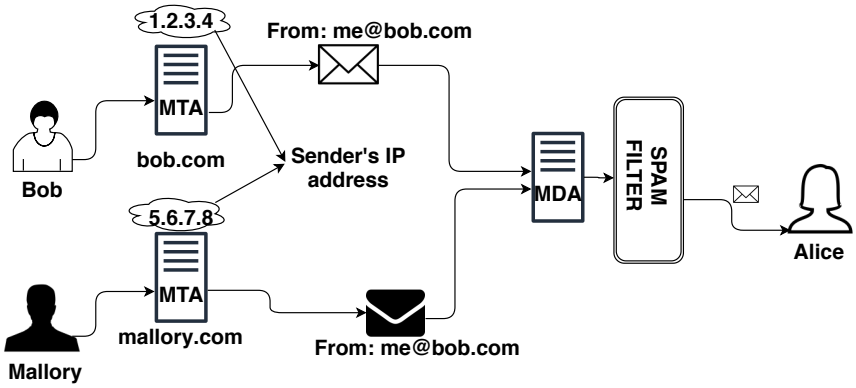


Figure 4.1: Email sending and receiving procedure.

approach to generate the datasets and find defensively registered domains. Sections 4.4 presents the analysis of the results for scanned domains and subdomains as well as for emulation of SPF rules. In Section 4.5, we study the trust-based authentication issue and Section 4.6 presents a methodology for preventing domain spoofing. Section 4.7 describes our remediation actions. Section 4.8 reviews related work and Section 4.9 concludes the chapter.

4.2 Background on Anti-Spoofing Schemes

To understand the issue of email authentication, we briefly explain the process of mail delivery. Figure 4.1 shows Bob (sender) who sends legitimate mails to Alice (receiver). Mallory (attacker) wants to send an email that impersonates Bob to Alice. Mallory and Bob use their respective servers (`mallory.com` and `bob.com`) to send mails. The Mail Delivery Agent (MDA) on the Alice server delivers two emails with the same sender address (`me@bob.com`) but coming from different IP addresses (assuming there is no spam filtering involved). One mail is from Bob (originated from the `1.2.3.4` IP address) and the other from Mallory (originated from `5.6.7.8`).

An effective anti-spoofing mechanism needs to differentiate the Mallory message from the legitimate Bob's mail. The current first lines of defence to protect end-users from spoofed emails include SPF [22], DKIM [161], and DMARC [23].

4.2.1 SPF – Sender Policy Framework

SPF is a set of text-form rules in `TXT` resource records of the Domain Name System (DNS). SPF specifies a list of servers allowed to send emails on behalf of a specific domain. During mail delivery over the SMTP protocol, the recipient server authenticates the sender Mail Transfer Agent (MTA) using a given `HELO` or `MAIL FROM` identity based on the published SPF record and the IP address of the sender—SPF needs to contain the domain portion of the `MAIL FROM` identity. In our example, the Alice server gets the `TXT` records of the `bob.com` domain from DNS. Then, it checks whether the sender IP address is on the list of IP addresses allowed to send emails from the `bob.com` domain and decides whether the message should be rejected or delivered to Alice.

The decision is made by the `check_host` function described in RFC 7208 [22] that takes three arguments on input (IP address of the sender, the domain, the `MAIL FROM` or `HELO` identity) and returns one of the seven possible results shown in Table 4.1. The third column of the table presents the actions recommended by RFC 7208.

Below, we review the most common SPF rules useful for understanding the threat models presented in the next section (see RFC 7208 for more details). A valid SPF version 1 record must begin with string `v=spf1` followed by other SPF *mechanisms*, *qualifiers*, and *modifiers*. Mechanisms describe the set of mail servers for a domain and can be prefixed with one of four qualifiers: `+` (*Pass*), `-` (*Fail*), ~ (*SoftFail*), `?` (*Neutral*). If a mechanism results in a match, its qualifier value is used. *Pass* (i.e., `+`) is the default qualifier.

The most common SPF mechanisms are the following:

- `ip4` and `ip6` – they specify an address or a set of IPv4 (or IPv6) addresses to match by the `check_host` function with respect to the sender IP address.
- `a` and `mx` – they tell the `check_host` function to perform first a DNS lookup for `A` (or `MX`) records of a given domain and then compare the returned IP addresses with the IP address of the sender.
- `exists` – it indicates a DNS domain name used for a DNS `A` query. If the query returns any `A` record, this mechanism matches.
- `include` – it tells the `check_host` function to include the SPF rule of another domain in the evaluation, which may result in calling the `check_host` function

Table 4.1: Possible results of the SPF `check_host` function and their definitions.

Result	Definition	Recommended action
None	1. No valid domain name was extracted from the SMTP session. 2. No SPF record was retrieved from the domain name.	1. The action must be the same as the *Neutral* output.
Neutral	1. There is no definite assertion (authorized or not) about the sender.	1. Depends on the receiver system.
Pass	1. Client is authorized to send emails with the given identity.	1. Whitelist the domain in terms of SPF.
Fail	1. Client is not authorized to send emails with the given identity.	1. Depends on the receiver system. 2. Make decision based on the DMARC policy.
Softfail	1. Client is not authorized to send emails with the given identity. 2. No strong policy specified by the domain owner.	1. Receiver should not reject the message. 2. May mark the message as suspicious.
Temperror	1. A temporary error occurred during retrieving the SPF policy.	1. May defer the message. 2. May deliver the message and mark it.
Permerror	1. Parsing problem in published SPF.	1. May deliver the message and mark it.

recursively to fetch and analyze the SPF records of the included domains.

- `all` – it always matches, so its corresponding qualifier results in the final decision. For example, `v=spf1 mx -all` means: allow `MX` servers of the domain to send mail and prohibit all others.

The final result of the mechanisms could be *Match*, *No match*, or *Exception*. Qualifiers combined with mechanisms, generate the final input for the `check_host` function that evaluates the SPF rule.

Modifiers provide additional information about the SPF records, for instance:

- `redirect=another-domain` – the SPF record for `another-domain` replaces the current record. The redirected domain becomes the target of all DNS queries and evaluations instead of the original domain.

Let us consider the following example:

```
v=spf1 a ip4:1.2.3.0/24 -all
```

when the `A` record `example.com A 6.7.8.9` is stored in DNS. The SPF rule states that only machines with the IP address of `6.7.8.9` (the `a` mechanism) or with the IP address in the range of `1.2.3.0...255` (the `ip4` mechanism) are permitted senders (all others are forbidden). However, by only changing `-all` to `+all`, any machine is permitted to send emails on behalf of the domain `example.com` with the successful SPF *Pass* result.

4.2.2 DMARC

DMARC [23] builds on top of SPF and DKIM by explicitly stating the policies to apply to the results of SPF and DKIM. In particular, DMARC binds names checked by SPF with what is listed in the `FROM:` field of the mail header by means of *alignment*, which expresses the fact that these domain names should match (or partially match when using a relaxed setup). For instance, DMARC checks whether the name in the `MAIL FROM` SMTP command and the `FROM:` field of the mail header match or not. In the case of the alignment test failure, a DMARC policy can specify what to do with the message (accept, reject, or quarantine) and where to send reports in case of a mismatch. For a given domain name `domain.tld`, the DMARC policy is stored in the `TXT` record of `_dmarc.domain.tld`. Below, we present selected tags of DMARC that, when misconfigured, can be exploited by an adversary.

- `aspf` (Alignment mode for SPF) – it specifies whether the strict (`s` value) or relaxed (`r` value) alignment mode is required by the domain owner. The default value is the relaxed mode. In the strict mode, the domain name used in SPF must be the same as the domain used in the `FROM:` field of the header. In the relaxed mode, any subdomain of the domain can be used in the `FROM:` field of the header and will result in *Pass*.

- `p` (Policy) – it specifies the action to be taken by the receiver if the alignment test results in *Fail*. Possible values for this tag are: 1) `none` – no specific action,

2) `quarantine` – the message is suspicious and depending on the mail system of the recipient, it could be delivered as spam, 3) `reject` – the domain owner wishes to reject emails during the SMTP transaction that fail the alignment test.

- `ruf` (Reporting URI for failure) – it specifies the email addresses to which message-specific failure information is to be reported. This tag is important since this is the only bridge between the receivers and the true domain owners to fight spam emails [165].

- `sp` (Subdomain policy) – it has the same syntax as `p` but applies to subdomains of the domain name. In the absence of this tag, the policy of the `p` tag must be applied to all subdomains [23]. If subdomains are not used to send emails, the owner can set this tag to the `reject` value to prevent subdomain email spoofing.

Let us assume that the DMARC rule of the domain `example.com` is `v=DMARC1; p=none; aspf=r;`. If we have the previously mentioned SPF rule for this domain, an illegal sender with the IP address of `9.10.11.12` can forge emails on behalf of `example.com` or any (existent or non-existent) subdomain of `example.com`, and the delivery decision is up to the receiver since no strict rule has been specified in DMARC. However, changing the DMARC rule to `v=DMARC1; p=quarantine; sp=reject; aspf=s;` tells the receiver to label all the emails that did not pass the SPF evaluation as spam and reject all the emails from the subdomains of `example.com` at the SMTP level.

4.2.3 Threat Models

We now consider threats regarding SPF and DMARC in detail. To mitigate mail spoofing, domain owners set up SPF and DMARC rules then used by inbound mail servers. Therefore, if the recipient MTA does not support the SPF or DMARC check, no matter how strict the rules are, they will not be effective. A misconfigured SPF or DMARC (either syntactically or semantically) rule is as dangerous as the absence of the rules since the output of the evaluation does not lead to a correct decision.

We consider three possible types of threats:

- **Related to domain names**. If a domain uses a misconfigured SPF rule, then it is possible to send forged emails from any IP address with the SPF *Pass* result. For example, we have discovered that `microsoft.com.tr` used the `+all` mechanism

in its SPF rule, which made it easy for attackers to send forged emails on behalf of Microsoft from any IP address. Note that after notifying Microsoft, the issue was fixed.

- **Related to subdomains**. Each subdomain should have its own SPF and DMARC rules. Another possibility is to use the `sp` tag in DMARC of the domain name (lower-level domain) to explicitly specify the action to take when receiving messages from subdomains. A possible abuse of subdomains is the following:

  - If a subdomain has no SPF rule (and there is no specified wildcard rule) and no explicit DMARC action, then it is possible to misuse the subdomain for sending forged emails. For example, while `icann.org` has a strict SPF rule, there is no rule specified in `account.icann.org` and no DMARC policy regarding subdomains (also the default action for domains is `none`, which in this case applies to subdomains). Hence, it is possible to send emails with forged sender addresses (e.g., `support@account.icann.org`) with the SPF *Neutral* result.

  - If a subdomain does not exist, the result of the DNS query for the `TXT` record returns a name error (NXDOMAIN). Thus, the `check_host` function returns the *None* result (see Table 4.1). If there is no wildcard `TXT` record that covers non-existing subdomains and there is no DMARC policy specified for subdomains and the domain itself, then again, it is possible to send spoofed emails.

- **Wrong SPF rules**. If the `check_host` function cannot evaluate the existing SPF record of a domain name because of a syntax error, then the result is either *Temperror* or *Permerror*, and a legitimate email will likely arrive in the spam box. However, when the user marks this email as safe, the mail service may also accept spoofed emails from other IP addresses. We show in Section 4.5 how syntactically wrong SPF rules may break the trust-based authentication system of email service providers by allowing forged emails to land in the user inbox.

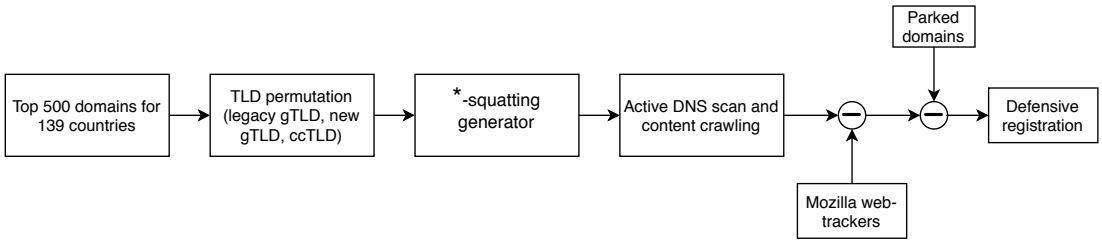


Figure 4.2: Generating the list of defensively registered domains.

4.3 Methodology for analyzing SPF and DMARC deployment

In this section, we describe the methodology for analyzing the deployment of SPF and DMARC. We start with three datasets to perform two different measurements: in one campaign, we use a dataset of approximately 236 million domains from various resources to measure the global adoption of SPF and DMARC. In the second campaign, we use top 500 domains of 139 countries from the Alexa list [166] and online banking systems for all countries provided by FONDY.[3] In the second campaign, our focus is on high-profile domains (well-known companies, governmental websites, and financial institutions) and their defensive domain registrations.

4.3.1 Global Measurements

Regarding the global scan of domains for SPF and DMARC, we collected approximately 333 million domains from open zone files, OpenData project of Rapid7, and all the available zone files in Centralized Zone Data Service (CZDS) offered by ICANN. Our data consist of all domains with `.com`, `.net`, `.org`, `.biz` legacy generic TLDs (gTLDs), approximately 1,100 new gTLDs, `.se` and `.nu` country-code TLD (ccTLDs), operated by the Internet Foundation in Sweden, and samples of other domains obtained from Rapid7. Then, we scanned all the domains for `A` resource record using the ZDNS[4] scanner from the ZMap project [167] to keep only alive ones. Finally, our dataset consists of 235,960,991 active domain names in total. We performed the measurement in September 2020.

[3] https://fondy.eu
[4] https://github.com/zmap/zdns

4.3.2 Top 500 Websites of All Countries

The Alexa website ranging system provides top 500 lists of most visited websites for 139 countries, which we collect for the purpose of this study as high-profile domains. Previous work [163, 168] used the Alexa top 1 million domains. However, we are interested in specific domains that may not be in the top 1M global popularity list but in the top list of each country, e.g., government websites or national businesses. In total, we collect 69,500 fully qualified domain names (FQDNs), which lead to 32,042 unique domains. Domain names are defined as 2^{nd}–level, or lower-level if a given TLD operator provides such registrations, e.g., `example.br` or `example.com.br` [169]. We use a modified version of the public suffix list maintained by Mozilla[5] to get domains from FQDNs. For the purpose of this study, we exclude all private TLDs such as `s3.amazonaws.com` or `blogspot.com`. The dataset consists of 14,084 domains with legacy gTLDs, 1,070 domains with new gTLDs, and 14,084 domains with country-code TLDs. We refer to this list as the TOP500 list.

4.3.3 Defensive Registrations

Defensive registration refers to the process of registering domain names (often across multiple TLDs) with different grammatical formats to protect brands from attacks like *typo-squatting* [170]. For example, the `brand.com` company may register `brand.net` and `brand.org`, then redirect them to the original website. Figure 4.2 shows the algorithm to generate their list. We use the following steps to generate defensively registered names using the names in the TOP500 list:

- For each domain name in the TOP500 list, we generate the domain names over all the possible TLDs including new gTLDs, legacy gTLDs, and ccTLDs. For example, for `paypal.com`, we generate `paypal.tld` where `tld` refers to all the ccTLDs (e.g, `paypal.in`), legacy gTLDs (e.g., `paypal.net`), and new gTLDs (e.g., `paypal.support`).

- For each domain in the TOP500 list that uses country code TLD or legacy gTLD, we generate *-squatting domains (for *-squatting, we use insertion, deletion, substitution, and internationalized domain names using DNSTwist package[6]). We

[5] https://publicsuffix.org
[6] https://pypi.org/project/dnstwist/

generate 145,250,849 unique domain names.

- We scan all generated domains for `TXT` records with ZDNS. By excluding all DNS error results (e.g., NXDOMAIN, TIMEOUT, and SERVFAIL), we end up with 1,185,167 unique domains. Then, we extract the defensively registered domains based on the following three conditions:

  1. IP address in the requested `A` record of the domain is the same as for the `A` record of at least one corresponding domain in the TOP500 list,
  2. authoritative name server in the `NS` record of the domain is the same as in the `NS` record of at least one corresponding domain in the TOP500 list,
  3. domain part of the automatically visited domain homepage URL is the same as one domain in the TOP500 list, and the list reduces to 235,508 domains.

- Some of the domains in the list are related to web trackers [171] and parked domains. For parked domains, we exclude them using the method proposed by Vissers et al. [172], whereas for web trackers and advertising domains, we exclude them by using the Mozilla blacklist for trackers [173].

Our final list contains 55,059 defensively registered domains. For example, we find 226 domain names either registered by Google Inc. for `google.com` or by MarkMonitor[7] on behalf of Google, and 201 domain names related to PayPal Inc.

4.3.4 Subdomain Enumeration

We have generated the list of known subdomains for each entry of the TOP500 list using the Spyse[8] API. We only consider 'first-level' subdomains and exclude `www` and name servers since it is more likely thet attackers use a first-level subdomain for sending spoofed email since it looks more legitimate. In total, we generate 212,361 subdomains for domains in the TOP500 list.

4.3.5 Banks and Financial Websites

For banking and financial websites, we leverage a list of 7,022 domains from the FONDY Github repository[9] and generate 39,310 subdomains using the same method as de-

[7] https://markmonitor.com
[8] https://spyse.com
[9] https://github.com/cloudipsp/all_banks_ips

scribed in the previous section.

4.4 Results on SPF and DMARC Adoption

After collecting all the datasets, we perform three types of scans for all domains and subdomains: 1) find `TXT` records to extract SPF rules, 2) find `TXT` records by prepending `_dmarc` to the domains and subdomains (i.e., `_dmarc.domain.tld`) to retrieve DMARC rules, and 3) analyze SPF and DMARC rules by emulating the `check_host` function [174] using our server IP address as the IP address of the sender (without actually sending emails).

In this section, we present the results of the first two scans for SPF and DMARC rules at each domain and its subdomains.

4.4.1 Global Scan of the SPF and DMARC Rules

As the result of scanning 236 million domain names, we find that only 73,833,342 domains have SPF records set, which is approximately **31%** of all domains. The comparison of the obtained results with the scanning results of the top 1M domains in the Alexa list performed by Hu et al. [163] with 44.9% SPF adoption rate, shows that the global adoption of SPF is approximately **13.9%** lower than in the Alexa top 1M domains. We expected this result because Alexa top 1M domain names are more valuable and well-established in terms of DNS resource records, and therefore, they do not give a representative overall picture of the global SPF deployment.

Regarding DMARC, only 310,185 out of 236 million domains have DMARC corresponding to approximately **0.13%** of the population. For the domains with a DMARC rule, 41% of them have `p=reject`, 9.3% have `p=quarantine`, and 39.6% have `p=none` rule. These figures are also far different from the 5.1% of the domain names in the Alexa top 1M domains with DMARC rules [163], which again confirms that more popular domain names deploy email anti-spoofing schemes on a wider scale.

4.4.2 High-Profile Domains and Defensive Registrations

Tables 4.2 and 4.3 present the results of the scans using ZDNS to retrieve SPF and DMARC rules. Columns contain the following information: 'norecord' – domains exist but there is no SPF rule in the `TXT` record of the domains, 'noerror' – the record exists

Table 4.2: Scan results for SPF rules.

dataset	total	norecord (%)	noerror (%)	servfail (%)	nxdomain (%)	timeout (%)
TOP500 domains	32,017	29.88	65.92	0.23	0.18	3.78
TOP500 subdomains	212,361	76.15	5.77	0.1	16.31	1.68
Bank domains	7,022	22.39	64.95	1.28	2.75	8.63
Bank subdomains	39,310	70.34	3.53	0.09	22.96	3.09
Defensive domains	55,095	1.2	95.37	0.43	1.03	1.97

Table 4.3: Scan results for DMARC rules.

dataset	total	noerror (%)	servfail (%)	nxdomain (%)	timeout (%)
TOP500 domains	32,017	34.32	0.24	63.44	2.0
TOP500 subdomains	212,361	12.61	0.36	82.95	4.09
Bank domains	7,022	35.86	1.21	52.32	10.61
Bank subdomains	39,310	7.95	0.55	87.92	3.58
Defensive domains	55,095	40.08	0.36	57.86	1.7

and can be retrieved successfully, 'servfail' – DNS lookup failure, 'nxdomain' – the domain name does not exist in the zone file, 'timeout' – the DNS timeout error. For DMARC, the 'nxdomain' column is the same as 'norecord' column for SPF (if we get 'NXDOMAIN' answer to the DNS query for `_dmarc.domain.tld`, it means that `_dmarc` subdomain does not exist so there is no DMARC rule).

We can notice in Table 4.2 that **29.9%** of the domains in the TOP500 list and **22.4%** of the online banking domains do not have SPF rules at all. As the `check_host` function for the domains without SPF rules returns *None* (see Table 4.1), it is up to the receiver of the email to decide on whether to deliver a message and/or mark it as suspicious or not. While this behavior can be acceptable for regular domains, it is insecure for transactional domains (e.g., banking domains) as well as for high-profile domains (e.g., domains in the TOP500 list).

For defensively registered domains, Table 4.2 shows that only **1.2%** of them have no SPF rules, which is significantly lower than the results for TOP500 and banking domains. However, evaluating SPF alone is not sufficient since it is up to DMARC policies to make the final decisions about the delivery of messages.

As shown in Table 4.3, as many as **63.4%** and **52.3%** of TOP500 and banking domains have no DMARC rule, which means that even with correctly configured SPF rules, it is still possible to spoof emails. Furthermore, for the domains with a DMARC rule in place (34.3% and 35.9% for TOP500 and banking domains, respectively), we

have observed that a large part of them have the tag `p` equal to `none` (**60%** and **53.8%**, respectively, not shown in the table), which make them prone to email spoofing as well.

For defensively registered domains (see Table 4.3), **57.9%** of them do not have a DMARC rule, which means that it is possible to send spoofed emails. Among 40.1% of the domains with a DMARC rule, **26.7%** have the `p` tag equal to `none` and 65% have the `p` tag set to `reject`, which makes them bulletproof from domain spoofing at the SMTP level.

Overall, we expect much wider deployment of SPF and stricter DMARC rules for defensively registered domains in comparison to high-profile domains—if organizations decide to register domains defensively to avoid domain name abuse, they are also more likely to configure the appropriate SPF and DMARC rules.

4.4.3 Analysis of Spoofing Possibilities for Subdomains

Regarding subdomains, the results are worse since **76.1%** of the subdomains related to the domains in the TOP500 list and **70%** of the subdomains related to banking websites do not have SPF records at all (see Table 4.2). While it is not dangerous in itself, the absence of strict DMARC rules for subdomains makes them prone to subdomain spoofing. To mitigate this vulnerability, domains need to provide appropriate DMARC rules. The `sp` tag (or `p` tag in the absence of `sp`) in a DMARC rule specifies the default action to be taken upon receiving messages from subdomains with no SPF rule [23].

Table 4.4 shows the DMARC results for subdomains without SPF rules in both TOP500 and banking website lists. To obtain this result, we first scan `_dmarc.sub.domain.tld` to extract a `p` tag from each subdomain and in case of no DMARC rule in the subdomain, we scan `_dmarc.domain.tld` for `sp` or (in the case of its absence) `p` tags and apply the rule to subdomains (cf. RFC 7489 for more details [23]). In Table 4.4, `none`, `reject`, and `quarantine` columns correspond to the extracted rules as explained in Section 4.2.2. The 'invalid rule' column refers to the rules that do not follow the syntax specified in RFC 7489 and 'no-DMARC' column corresponds to the domains without DMARC rules in subdomains nor in the domain name. Note that sending emails from a subdomain of any domain with 'no-DMARC' (**67.1%** for TOP500 and **68.9%** for banking websites), with `none` rule (**19.7%** for TOP500 and **17.5%** for banking websites), and 'invalid-rule' (less than 0.1% in both cases), regardless of the fact if the subdomain exists or not (non-existing subdomains), does not result in a strict reject decision. This

Table 4.4: Specified DMARC action for subdomains with no SPF rule in the `TXT` resource record.

data	total	no-DMARC	`none`	`reject`	`quarantine`	invalid rule
TOP500-sub-no-SPF	161,720	108,535 (67.1%)	32,008 (19.7%)	13,286 (8.21%)	7,803 (4.82%)	88 (0.05%)
Bank-sub-no-SPF	27,650	19,070 (68.9%)	4,849 (17.5%)	2,682 (9.6%)	1,023 (3.69%)	26 (0.09%)

Table 4.5: Result of the SPF `check_host` emulation.

Result	TOP500	bank	defensive	bank subdomains	TOP500 subdomains
None	10,106	1,956	1,441	37,149	198,615
Neutral	1,497	236	6,220	56	683
Pass	50	10	114	2	37
Fail	7,083	2,268	22,255	860	4,511
Softfail	10,617	1,591	21,804	354	6,019
Temperror	135	155	523	778	1,485
Permerror	2,529	806	2,738	111	1,011
Total	32,017	7,022	55,095	39,310	212,361

behavior is potentially dangerous for transactional domains as it is possible to send emails with forged sender address using subdomains with no SPF record for as many as approximately **87%** of TOP500 and banking domains.

4.4.4 SPF Emulation Results

To analyze the validity of SPF rules using the `check_host` function further, we take advantage of `pyspf` [174] with our server IP address as the IP address of the mail sender. `pyspf` evaluates the SPF rule for a given domain and returns the SPF result. For the global scan of SPF and DMARC, 213,112 out of 73,833,342 domains result in SPF *Pass*, approximately 0.28% of the domains with SPF records. We also found that 6,199,210 domains (8.3% of the domains with SPF records) result in SPF *Permerror*.

Regarding the second database, Table 4.5 shows the results of the SPF emulation (see also Table 4.1 for the definition of each result and the corresponding recommended action). The reason for the SPF *Pass* result is either the `+all` mechanism in the SPF rule or the possible `redirect` modifier. Among the defensively registered domain names with the *Pass* result (114 domains), we have observed some well-known names like `microsoft.com.tr`[10] registered by MarkMonitor Inc.[7] on behalf of the Microsoft Corporation, as well as some major IT companies, local government, and TV channels

[10]The issue was fixed after sending notifications.

Table 4.6: Selected syntactically wrong rules that lead to the *Permerror* result in SPF.

Error type	Example	Correct rule	Frequency
Too many DNS lookups	-	SPF rule must generate less than 10 DNS query	4,349,463 (70%)
Two or more SPF records	-	must set one SPF record for each domain	733,750 (12%)
No valid SPF record for included domain	-	must set one SPF record for included domains	556,811 (9%)
Unknown mechanism found: all.	v=spf1 a mx -all.	v=spf1 a mx -all	153,455 (2.5%)
Invalid IP4 address: ip4:	ip4:xxx.xxx.xxx.xx?all	ip4:xxx.xxx.xxx.xx ?all	72,011 (1.1%)
Empty domain:: a:	v=spf1 mx a: -all	v=spf1 mx a:example.com -all	18,190 (0.2%)

websites for which we cannot provide the names for security considerations. However, the emulation results are available upon request.

We have noticed 12 different banking websites (1 in Spain and 11 in the United States) with the SPF *Pass* result. Although the number is fairly low, it is still enough for attackers to conduct a successful attack if they obtain the list of customer emails. In the TOP500 list for domains and subdomains, we have found 87 records with the SPF *Pass* result (50 for domains and 37 for subdomains) including several local government websites (mostly in the US), national financial websites, and national mobile operators with thousands of customers.

Table 4.5 shows 7,195 *Permerror* as the result of the `check_host` function. The majority of these domains and subdomains have at least one of the following three problems: i) syntax problem in the published SPF rule (approximately 5,400 records), ii) excessive number of DNS lookups because of too many recursive `include` mechanisms [22] (1,131 records), and iii) published more than one valid SPF records (640 samples). Table 4.6 shows selected syntactically and semantically wrong published SPF records. We can observe that not only the syntax is important to parse an SPF record correctly, but also the number of DNS lookups must be limited to 10 queries based on RFC 7208 (cf. Section 4.6.4). Approximately 91% of the SPF *Permerror* results are related to only three types of misconfigurations with a 70% violation in the number of DNS queries, followed by 12% of domains with more than one SPF record.

The domains and subdomains with *Permerror* are important because they may cause serious problems. Since the domains have SPF records, *Permerror* indicates that they are used by their owners to send legitimate messages to users. However, emails may never get delivered or delivered but labeled as spam (based on the action recommended for *Permerror* as described in Table 4.1). Importantly, we find that any attempt by the end user to detach the spam label from the legitimate email may whitelist all the emails from that domain name with the SPF *Permerror* result including forged emails

Table 4.7: Measurements of message delivery to inbox (IN), spambox (SP), or no delivery (ND) for five major email service providers.

Threat model	Gmail			Yahoo			Outlook			Yandex			Laposte		
	IN	SP	ND	IN	SP	ND	IN	SP	ND	IN	SP	ND	IN	SP	ND
+all in SPF of domain	10	0	0	8	1	1	6	0	4	10	0	0	6	0	4
Defensive registration	9	1	0	9	1	0	3	7	0	9	1	0	9	0	1
Non-existent subdomain	8	2	0	3	0	7	2	8	0	10	0	0	10	0	0
Existent subdomain	7	3	0	7	2	1	4	6	0	10	0	0	10	0	0
Trust-based authentication issue		✗			✓			✓			N/A			✗	

(see Section 4.5).

Moreover, a wrong implementation of the `check_host` function on the receiver without strict limitation of the number of DNS queries, may allow the attacker to put extra burden on the local recursive DNS resolver, which may lead to a Denial of Service (DoS) attack against the DNS server, as explained by Scheffler et al. [175]. Among the domains with syntactically wrong SPF rules, we observe some major IT companies e.g., `eset.lu`, the defensively registered domain for `eset.com` related to the ESET Internet Security.[10]

The SPF emulation results show that for several major IT companies, government websites, and one of the topmost banking website in the world, it is possible to send spoofed emails from both existent and non-existent subdomains as well as from some of their defensively registered domains due to weak or misconfigured SPF or DMARC rules.

4.4.5 End-to-End Spoofing Measurement

To show the possibility of email spoofing based on the different threat models presented in Section 4.2.3, we have tested end-to-end email spoofing from well-known brands to our own registered email addresses at i) Gmail, ii) Yahoo, iii) Outlook, iv) Yandex, and v) Laposte email services. We follow the same steps as Hu et al. [163] to ensure research ethics. Table 4.7 shows the test results. We have considered four different possibilities, namely, a) the SPF record of the domain has `+all` in its rule set, b) the defensively registered domain has neither an SPF nor DMARC rule to reject our emails, c) non-existent subdomains (e.g., `accounts`.icann.org), and finally, d) an existent subdomain without proper SPF configuration or a restrictive DMARC rule (e.g., `account`.icann.org). For ethical reasons, we do not provide the brand names of high-profile domains on behalf of which we sent emails, because for some of them, the problem is still unsolved.

We can observe in Table 4.7 that in the first case (for which there is a `+all` in the SPF record), almost all the emails were delivered into the inbox by Gmail, Yahoo, and Yandex. Outlook and Laposte perform slightly better with 60% inbox delivery and 40% rejected emails. For the defensively registered domains, except for Outlook (with 70% delivered into the spam-box), all other email service providers successfully delivered almost all the sent mails into their inbox. Regarding non-existent subdomains, Outlook labeled 80% of the emails as spam while Yahoo rejected 70% of them. Other three services delivered almost all the emails. For the existent subdomain, Outlook performed the best by labeling 60% of the emails as spam. Surprisingly, Yandex delivered 97.5% of all sent emails into inbox, the worst performance in terms of the SPF and DMARC evaluation. The results show that attackers can successfully spoof all the tested email services by sending emails from non-existent subdomains, if domains do not have a strict reject DMARC policy.

4.5 Trust-based Authentication Issue

In this section, we show how a syntactically wrong SPF rule in a legitimate domain can push users to break the trust-based authentication system by labeling a legitimate email as safe and letting forged emails land in the user inbox. We examine five popular email providers: Outlook, Yahoo, Gmail, Laposte, and Yandex. We explain the issue using the Outlook service as an example, but the process is the same for other email service providers. Table 4.7 presents the summary of results.

First, we register a domain (`dnsabuse.xyz`), set up a mail server, and the DNS `A` record of the domain. We use `v=spf1 a aaaa -all` as the SPF rule in the `TXT` record for our registered domain (i.e., syntactically wrong SPF rule because of a nonexistent `aaaa` mechanism to generate the *Permerror* result). Then, we send a legitimate email with our server to our `outlook.com` email address. Since the SPF record is syntactically wrong and the reputation of our domain is low, the legitimate email lands in the spam box (as we expect) with the SPF *Permerror* result. If the user marks the email as *'safe sender'* (in case of Yahoo, the button label is *'add sender to contacts'*), then the Outlook service considers this email as safe (a correct assumption as it is a legitimate email). However, from now on, Outlook (as well as Yahoo) also accepts spoofed emails from other IP addresses that spoof the domain name.

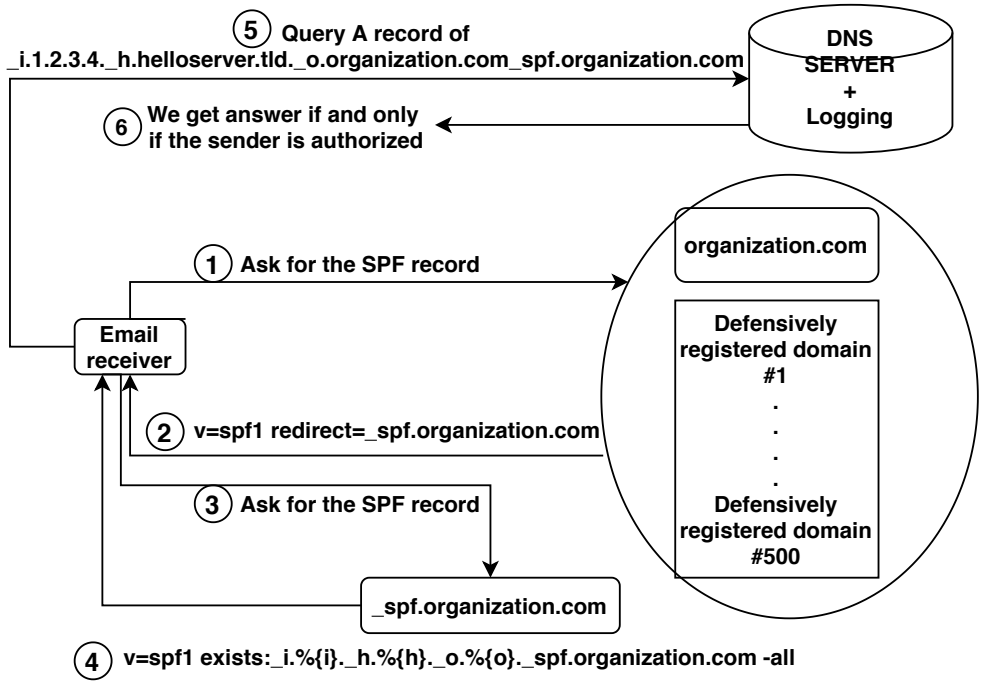


Figure 4.3: Methodology for preventing domain spoofing.

We suspect that Yahoo and Outlook services whitelist the sender domain name instead of their IP addresses. On the other hand, the Laposte service rejects the sender with SPF *Permerror* at the SMTP level and sends a bounce message informing the sender about the reason for rejecting the mail (i.e., syntax error of SPF). We were not able to evaluate the trust-based authentication for Yandex since both emails (from the legitimate and illegitimate servers) land in the user inbox. Finally, Gmail does not suffer from the issue. We assume that when users detach the spam label from a legitimate email, Gmail only whitelists the IP address instead of the domain name.

4.6 Methodology for Preventing Domain Spoofing

In this section, we present a methodology for preventing domain spoofing elaborated based on the experience gained in a study of a real-world scenario related to attacks performed on one of the government financial sectors in a European country. Due to security and ethical considerations, we do not give the name of the organization nor the name of the country.

The organization had one official registered domain (with ccTLD) and more than 500 defensively registered domains to protect the official one. In 2019, the domain administrators realized that a quite considerable number of attacks targeted their or-

ganization using different attack vectors: i) sending a forged email on behalf of the main domain, ii) sending emails with the `MAIL FROM` address of the defensively registered domains, and most importantly iii) sending emails from non-existent subdomains of either the main domain or defensively registered domains. The main problem was that the targeted organization had no control over any part of the attack scenarios. They did not know anything about the sender, which could be the attacker or a compromised machine sending spoofed emails on behalf of the attackers, nor anything about the receivers of the emails. Thus, to solve the problem, not only they had to identify the sender but also inform possible recipients so that they do not accept incoming messages and potentially send a report related to these emails. Figure 4.3 illustrates the resulting methodology for preventing domain spoofing, a combination of good practices for managing SPF and DMARC records and analyzing DNS logs.

Assume that the IP address of the attacker is `1.2.3.4`, the host name used in the SMTP `HELO/EHLO` command is `helloserver.tld`, and the `MAIL FROM` field used in the spoofed email is `organization.com`, the same as the targeted brand. In this scenario, the SPF rule of the main domain, as well as all the defensively registered domain names, point to a single subdomain (`_spf.organization.com`) under the control of the organization using `redirect` modifier. When the receiver receives a spoofed email on behalf of `organization.com` (or of any defensively registered domain), it asks for the `TXT` record, retrieves the SPF rule of the domain (step ①), and gets the following answer: `v=spf1 redirect=_spf.organization.com` (step ②). In step ③, the receiver again asks for the SPF record of the specified domain name in the `redirect` modifier and receives a macro specified by the `exists` mechanism (step ④). The `exists` mechanism tells the receiver to create the domain name based on the specified rules and query the generated domain for the `A` resource record. The receiver can make the final decision based on step ⑥. If the domain name in step ⑤ resolves (no matter to which IP address), it means that the email is legitimate in terms of SPF. However, if the query returns no result (e.g., NXDOMAIN) not only the SPF will fail but also the DNS server logs the IP address of the attacker (or the compromised machine used by attacker) as well as the targeted domain (the main domain of the organization or one of the defensively registered domains). In addition, the receiver sends an extra `TXT` query for the DMARC policy to make the final decision about the received email. By specifying the `ruf` field in the DMARC rule, the domain administrators will receive a copy of the rejected

email (e.g., phishing email) for further forensic analysis both to identify bugs in their mail software and gain better insight into the possible phishing/spam attacks on their domains.

After one year of using the methodology to protect the targeted organization, the results show that this technique can effectively reduce the number of phishing attempts on the organization, which we can consider as a good practice not only to protect the brands from phishing/spam attacks that use domain spoofing but also to identify the malicious email senders.

4.7 Remediation

Notifying the owners of the affected domains with misconfigured or missing SPF and DMARC rules is highly problematic since there is no straight way to retrieve the contact information of the domain owners [176, 177]. Public availability of the domain WHOIS data is affected by the introduction of the General Data Protection Regulation (GDPR) and "Temporary Specification for gTLD Registration Data" adopted by ICANN [178]. It obliges generic TLD registries and registrars to redact the Registrant and Administrative Contact in the public WHOIS.

Therefore, we decided to perform notifications through the Computer Security Incident Response Teams (CSIRTs). We use the following bottom-up approach to send notifications—we send email notifications if there is a CSIRT responsible for: 1) the domain name, 2) the TLD of the domain (mostly in case of private TLDs), 3) the IP range to which the IP address of the domain belongs to, 4) the autonomous system of the IP address for that domain, or 5) the national CERT responsible for the TLD (in case of country-code TLD) or the entire IP address space. We used this approach to perform two notification campaigns: the first one for high-profile domains, which are more critical to be fixed as soon as possible, in December 2019, and the second campaign related to the global scan in September 2020.

4.7.1 Results of the First Notification Campaign

Regarding high-profile domains, we have sent 128 emails to notify CSIRTs responsible for 7,653 domains with SPF *Pass* or *Permerror* results. We were not able to find any abuse contact address of responsible CSIRTs for 573 domains. For some high-profile

domains prone to phishing attacks, e.g., `microsoft.com.tr`, we manually visited their websites and contacted them directly. In the first 5 days after sending notifications, we repeated our scans and found that 160 domain owners re-configured their SPF rules. The quickest clean-up action was initiated by the US government CERT (50 domains), national CERT of Austria (7 domains), Spain (7 domains) followed by CERT Polska, French CERT (ANSSI) and Danish CERT (CFCS-DK): 5 domains each.

Re-scanning the same set of domain names in October 2020 shows that 1,734 domain names changed their status from *Permerror* to *Softfail* (663 domains), *Fail* (569), *Neutral* (83), *None* (361), *Pass* (2), and *Temperror* (56). Moreover, 43 out of 152 high-profile domains changed their status from *Pass* to another status. Note that it is challenging to assess the effectiveness of our notification campaign because administrators may replace, for example, one misconfiguration by another (e.g., *Permerror* by *Pass*), however overall, after notifying CSIRTs responsible for misconfigured domains, as many as **23.2% (1,777 out of 7,653) were re-configured**.

4.7.2 Results of the Second Notification Campaign

Regarding the global scan, we found the total number of 6,412,322 misconfigured domains, 213,112 with SPF *Pass* results, and 6,199,210 with SPF *Permerror* results. For 23,116 domain names, we were not able to find any contacts to responsible CSIRT. Using the same above-mentioned notification approach, we sent emails to 110 CSIRTs. For some CSIRTs, due to the large size of the attachment files, we had to send two separate emails, one related to domains with SPF *Pass* and the other one related to SPF *Permerror*. Then, we re-scanned the domains every week to see how CSIRTs react to our notifications. After one week, we observed changes in the SPF results of 11,552 domains and another 917 domains after the second scan (we did not observe any major change after the third scan). For those domains that changed their SPF results, 567 changed from *Pass* to *Fail*, 56 domains to *Neutral*, 8,792 domains to *None*, 2,344 to *Softfail*, and others to *Temperror* and *Permerror*. We also did not observe any major changes in domains with *Permerror*. Overall, after notifying CSIRTs responsible for affected domains, **0.2% (12,469 out of 6,412,322) were re-configured.**

The differences between the remediation rates of the first and second campaign are to be expected and likely caused by: 1) the importance of vulnerable domains (high-profile domains are more likely to be fixed), 2) the magnitude of vulnerable domains

(the number of vulnerable domains in the second campaign was three orders of magnitude larger). The magnitude of vulnerable resources is important since obtaining contact information at scale is highly problematic (for researchers, security companies, or CSIRTs), especially after the introduction of GDPR, and there is no alternative method suitable for large-scale notifications [177].

4.7.3 Notes on Notification Campaigns

We present below more insight into our notification campaigns and summarize major problems we encountered.

Figure 4.4 (see Appendix) shows the email template of the first notification campaign we sent to CSIRTs about vulnerable/misconfigured SPF records. Although we did not explain the problem in detail, we received many replies from the CSIRTs in the first 24 hours after sending notifications either thanking us for notifying them (we only consider manually typed emails rather than automatic replies) or with followup questions about the problem, e.g., whether we can prove it by sending a spoofed email. Figure 4.5 (see Appendix) shows one of the replies we received from one of the CSIRTs stating that they do not understand the problem and they think that the receiving MTA should be "smart enough" to handle *Permerror* responses. After providing the proof of concept, they notified the domain owners and fixed all the SPF records. On the other hand, in the second campaign, we used a more detailed email template and explained more about the problem (for each domain, we specified the reason for misconfiguration, i.e., *Permerror* or *Pass*).

Note that re-configuring domain names does not necessarily mean that the domain owners permanently solved the problem. As mentioned earlier, for 8,792 domains, the SPF result changed from *Pass* to *None*, which means that either the administrators removed the SPF record (possibly thinking that removing the record is better than setting a wrong one) or the domain just expired and was not registered anymore.

Sending large scale notifications present its own difficulties already discussed in previous work [176,177,179]. In addition to them, we encountered three major problems with sending emails to CSIRTs:

- Some countries do not have an official CSIRT to notify.
- Some CSIRTs do not have an officially published email address. Therefore, to no-

tify them, one needs to fill an online form on their websites making it impractical for large scale notifications.

- Finally, some CSIRTs changed their email addresses so that we received bounced emails.

Overall, our experience from the two notification campaigns shows that reporting vulnerabilities through CSIRTs can be effective but depends on its possible impact and magnitude of affected resources.

4.8 Related Work

In this section, we review previous work on measuring and analyzing email security extensions.

Durumeric et al. [180] measured the adoption of SMTP security extensions and their impact on end users. They studied SMTP server configurations for the Alexa top one million[11] domains and SMTP connections to and from Gmail gathered over a year. They reported the existence of a long tail of over 700,000 SMTP servers, of which only 35% successfully configure encryption, and only 1.1% specify a DMARC authentication policy.

In 2017, Durumeric [162] measured the extent of SPF and DMARC adoption for one million top domains in the Alexa list. His results showed that 40.1% of the domains have published SPF records while only 1.1% of them have valid DMARC records. Hu and Wang [163] reported similar statistics in 2018 with the results of 44.9% published SPF records and 5.1% published DMARC records showing approximately 5% of increase in one year. In their end-to-end experiment, they spoofed 30 high-profile domains and reported the ratio of emails that reached inboxes of selected email providers. We perform a similar analysis for both SPF and DMARC records but in two different phases. First, we analyze the global adoption of SPF and DMARC rules for different TLDs and then, we focus on more prominent domains (with transactional emails) including banking websites, government portals, national and international businesses as well as defensively registered domains and their subdomains. We also consider end-to-end spoofing but just as a proof of concept for our defined threat models and only for 10 high-profile domains.

[11] https://www.alexa.com/topsites

Foster et al. [168] evaluated the security extensions using a combination of measurement techniques to determine whether major providers support the Transport Layer Security (TLS) protocol [146] at each point in their email message path, and whether they support SPF and DKIM on incoming and outgoing mail. They reported that while the use of SPF is common, enforcement was limited. Scheffler et al. [175] investigated the consequence of a wrong implementation of the `check_host` function at the receiver, which lets attackers perform denial-of-service (DoS) attacks on a local DNS resolver. While our goal is not to evaluate the SPF abuse, we show that 4,349,463 domains in the global scan, 1,131 high-profile, and defensively registered domains have published SPF records that require more than 10 DNS lookups. Therefore, such misconfigured records may lead to abuse of local DNS resolvers.

Finally, Hu et al. [181] investigated the reasons behind the low adoption rates of anti-spoofing protocols. They conducted a user study involving email administrators and showed that they believe the current protocol adoption lacks the crucial mass due to the protocol defects, weak incentives, and practical deployment challenges.

4.9 Conclusion

It is paramount for high-profile domains and defensively registered domains to establish appropriate SPF and DMARC policies to reduce the chance of successful spear phishing attacks. In this chapter, we evaluate the adoption of the SPF and DMARC security extensions by domain names in two phases and analyze spoofing possibilities enabled by the absence of their rules or their misconfigurations. The results show that a large part of the domains do not correctly configure the SPF and DMARC rules, which enables attackers to successfully deliver forged emails to user inboxes. In particular, we show that for top 500 domains of 139 countries, the adoption rate of SPF and DMARC records are 65.9% and 34.3%, respectively. For banking websites, we obtain almost the same results (64.9% and 35.9%) as for the TOP500 list. However, for defensively registered domains, the results are significantly higher especially in terms of published SPF records with 95.37% adoption and 40.1% for DMARC. We also, for the first time, investigate the problem of subdomains in the anti-spoofing techniques and their possible abuse to send forged emails.

We also emulate the SPF `check_host` function not only to evaluate *Pass* and *Fail*

results but also obtain all the possible results such as *Permerror*, *None*, and *Neutral* for both domains and subdomains. The investigation shows that syntactically wrong SPF rules may break the trust-based authentication system of email service providers (e.g., Outlook and Yahoo) by allowing forged emails to land in the user inbox. To improve deployment of SPF and DMARC, we have presented a methodology for managing SPF and DMARC records and analyzing DNS logs that may prevent domain spoofing.

For remediation, we have sent the total of 238 emails to notify the CSIRTs responsible for 6,419,975 domains. Within the first two weeks after the notification campaigns, they managed to inform domain owners and re-configure SPF records of 12,629 vulnerable/misconfigured domains. More importantly, as many as 23.2% of high-profile domains were re-configured at the end. Our experience shows that disclosing vulnerabilities through CSIRTs can be effective, especially for valuable domain names. Finally, while we do not publish the scan data because of ethical concerns, we make the data available upon request to encourage reproducibility.

Acknowledgments

This work has been carried out in the framework of the COMAR project funded by SIDN, the .NL Registry and AFNIC, the .FR Registry. It was partially supported by the PrevDDoS project funded by IDEX UGA IRS and the ANR projects: the Grenoble Alpes Cybersecurity Institute CYBER@ALPS under contract ANR-15-IDEX-02, PERSYVAL-Lab under contract ANR-11-LABX-0025-01, and DiNS under contract ANR-19-CE25-0009-01.

Appendix

In this section, we present the email template of the first notification campaign we sent to CSIRTs about vulnerable/misconfigured SPF records (see Figure 4.4) and the exchange of mails with one of the CSIRTs that led to fixing the misconfigured SPF records (see Figure 4.5).

Hello,

We are writing to inform you of a misconfiguration in the Sender Policy Framework (SPF) of the domain names under your jurisdiction. This means that attackers are able to send spoofed emails on behalf of these domains.

Please find the list of vulnerable/misconfigured domains along with the corresponding SPF error in the attached file.

This vulnerability/misconfiguration has been rated as 5.4 out of 10.0, according to the scale published on the Common Vulnerability Scoring System (CVSS).
More information about the score of the vulnerability/misconfiguration can be found here:
https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N

The vulnerability/misconfiguration was brought to our notice on ${date_of_scan}.

If you have any question regarding this matter, please feel free to write us at ${our_email} referencing ${subject_of_notification}.

Sincerely,
${your_name}
${affliation}

Figure 4.4: Content of the email for the first campaign.

----------CSIRT reply----------
Hello,
Thank you for your notification.

We were looking into the reported domains and tried to reproduce your observations/thoughts.

However, we didn't come to the same conclusion.

In case of (partially) invalid SPF records like most of the reported domains are, the system is smart enough to accept the specifically mentioned IPs, but refuse everything else. I'm not sure how you think an attacker could abuse those.

Could you perhaps pick an example of the supplied list and explain what you have in mind?
----------OUR reply----------
We provided the POC by spoofing one of the domains

----------CSIRT reply----------
I got it. Despite the fact that the RFC in paragraph G.3. encourages to take special care of Permerror on checking site, it looks like most companies are just openly letting mails through. Definitely not what I expected.

We informed all the domain responsible, also for the ticket you opened through ******* (they all end up at the same place, so please be invited to use ******** in the future).

Figure 4.5: Sequence of the emails we have exchanged with one of the CSIRTs.

Chapter 5

Cybercrime After the Sunrise: A Statistical Analysis of DNS Abuse in New gTLDs

Coauthors: Maciej Korczyński, Maarten Wullink, Samaneh Tajalizadehkhoob, Giovane C.M. Moura, Arman Noroozian, Drew Bagley, and Cristian Hesselman

5.1 Introduction

Starting in 2007, The Internet Corporation for Assigned Names and Numbers (ICANN) introduced the new Generic Top-Level Domain (gTLD) program[1], which enabled hundreds of new gTLDs to enter the domain name system (DNS) since the first delegations. More than 1,900 applications for new gTLDs were filed after the process opened in 2012. To date, more than 1,200 new gTLDs have been delegated to the DNS root zone (e.g.: .nyc, .top). This expansion of the domain name space not only offers a wide range of options for consumers, but also potentially provides new avenues for cybercriminals to abuse domain names. Anticipating potential problems, ICANN has also built safeguards into the program in an attempt to mitigate the prospect of abusive, malicious, and criminal activity in these new gTLDs, such as phishing, spam, and malware distribution[2].

[1] https://gnso.icann.org/en/group-activities/inactive/2007/new-gtld-intro
[2] https://newgtlds.icann.org/en/reviews/cct/dns-abuse

In a previous study, Halvorson *et al.* [182] concluded that speculative and defensive registrations dominate the growth of registrations in new gTLDs. Their work, however, provides very little empirical information about the security of new gTLDs. In this paper, we investigate the following research question: *how do abuse rates in the new gTLDs compare to legacy gTLDs, since the implementation of the new gTLD program?* We take into account the new gTLDs as well the different parts of the industry involved: registries, registrars, and privacy/proxy service providers.

To this end, we combine multiple datasets from various sources including zone files, domain name `WHOIS` records, data obtained through active measurements, and 11 abuse feeds provided to us by 5 reputable organizations. These represent malware, phishing, and spam abuse and cover a three-year period from 2014 to 2016.

Overall, our main contributions can be summarized as follows:

- The research offers a comprehensive descriptive statistical comparison of rates of domain name abuse in new and legacy gTLDs as associated with spam, phishing, and malware distribution (§5.5.1) to evaluate joint effects of the existing anti-abuse safeguards.
- Using regression modeling we perform inferential statistical analysis to test the correlation between passively and actively measured properties of new gTLDs as predictors of abuse rates (§5.5.2).
- We analyze proportions of abusive domains across other entities relevant to abuse prevention practices, i.e. registrars and privacy/proxy providers (§5.5.3 and §5.5.4).

Our findings reveal surprising, previously unknown trends that are relevant since new gTLDs operate on the basis of different business models and history in comparison to legacy gTLDs. While patterns of abuse vary with respect to abuse type, our analysis suggests that the total number of spam domains in all gTLDs remains relatively constant. Simultaneously, the number of spam domains in new gTLDs is higher (Q4 2016) and growing. We also observe a significant decrease in the number of malicious registrations in legacy gTLDs (§5.5.1.7). Therefore, we see a new trend: attackers switching from abusing legacy to new gTLD domain name space. Our analysis of the Spamhaus blacklist also reveals that in the last quarter of 2016, new gTLDs collectively had approximately one order of magnitude higher rate of spam domains per 10,000 registrations compared to legacy gTLDs (§5.5.1.8).

This research also systematically analyzes how different structural and security-related properties of new gTLD operators influence abuse counts. Our inferential analysis reveals that abuse counts inversely correlate with the restrictiveness of registration policies (§5.5.2). The analysis of abuse across new gTLDs, registrars, and privacy/proxy service providers reveals discrete entities afflicted with significantly high concentrations of abused domains. We find new gTLDs and registrars with concentrations of blacklisted domains above 50% (§5.5.1.8 and §5.5.4). For one registrar, more than 93% of its domains were reported as abusive by SURBL.

ICANN is willing to further expand gTLDs. Therefore, it is important to understand how miscreants are using the expanded domain name space in their favor. Finally, as the presented state of the art in gTLD abuse is in clear need of improvement, we develop cases for modifying the existing safeguards and propose new ones. ICANN is currently using these results to review existing anti-abuse safeguards, evaluate their joint effects and to introduce more effective ones before an upcoming new gTLD rollout.

5.2 Background

The Internet Domain Name System (DNS) comprises one of the critical services of the Internet, mapping hosts, applications, and services from names to IP addresses [7]. ICANN [10] is the organization responsible for maintaining the Root domain namespace and its expansion with new top-level domains, in particular new gTLDs. ICANN also delegates the responsibility to maintain an authoritative source for registered domain names within a TLD to registry operators (e.g.: Verisign is the registry for `.com`). Registries, manage themselves and the domain names under their respective TLDs.

Three main entities are involved in the registration of a domain: registries (aforementioned), registrars, and registrants (so-called tripe-R). A registrant is a user or company, which in turn has to contact a registrar to register a domain name. A registrar (e.g.: GoDaddy), if affiliated with the TLD of the registrant's choice, will ask the registry to perform the registration of the requested domain.

In parallel, web hosting providers maintain server infrastructure that is used to host content for the domain. DNS providers operate authoritative DNS servers that resolve domain names to their corresponding IP addresses. Finally, WHOIS Privacy and Proxy service providers conceal certain personal data of domain name registrants.

5.2.1 Generic TLDs

The first group of generic top-level domains (gTLDs) was defined by RFC 920 [183] in October 1984 and introduced a few months later. The initial group of gTLDs (.gov, .edu, .com, .mil, .org, and .net) were distinct from country-code TLDs (ccTLDs). Until 2012, several gTLDs were approved and further introduced by ICANN, including a set of sponsored gTLDs such as .asia, .jobs, .travel, or .mobi. We refer to all gTLDs introduced before the new gTLD program initiated by ICANN in late 2013 as *legacy* gTLDs. This study analyzes a set of 18 legacy gTLDs (.aero, .asia, .biz, .cat, .com, .coop, .info, .jobs, .mobi, .museum, .name, .net, .org, .post, .pro, .tel, .travel, and .xxx), for which we were able to obtain zone files and WHOIS data, and compare them to *new gTLDs*.

5.2.2 New gTLDs

ICANN's new gTLD program began in 2012, expanding the root zone by delegating more than 1,200 new gTLDs starting in October 2013 [184]. To obtain a new gTLD, applicants are required to undergo an intensive application and evaluation process [182] that includes screening the applicants technical and financial capabilities for operating a new gTLD. Ultimately, after a new gTLD is assigned to an applicant, it will then be delegated to the root zone. Following initial delegation, each new gTLD registry is required to have a "sunrise" period of at least 30 days, during which trademark holders have an advance opportunity to register domain names corresponding to their marks, before the names are generally available to the public.

New gTLDs can be classified into four broad categories [184][3]:

- *Standard or generic gTLD*: gTLDs that are generally open for public registration, e.g. .movie, .xyz, or .family[4]
- *Geographic gTLD*: gTLDs that cover cities, states, or regions, e.g. .amsterdam or .berlin.
- *Community gTLD*: gTLDs that are restricted to a specific community, such as .thai, .radio or .pharmacy.

[3] Note that some gTLDs cross categories. For example, some community gTLDs such as .madrid are also geographic gTLDs [185].

[4] While most of these gTLDs are open to public registration, some may registries impose restrictions on the entities that can register domains.

- *Brand gTLD*: gTLDs specific to a company or a brand, such as .google or .hitachi.

In this chapter, we analyze new gTLDs that are intended for public use. Therefore, we exclude the great majority of brand gTLDs for which domains cannot be registered by regular users[5], in particular for malicious purposes. We cover new gTLDs for which registries have submitted their sunrise date information requested by ICANN. In the first quarter of 2014, there were 77 new gTLDs for which the sunrise period ended and domain names were available for public registration. For comparison, by the end of 2016 the group consisted of 522 new gTLDs.

5.2.3 Safeguards Against DNS Abuse

In preparation for the new gTLD program, ICANN sought advice from different DNS abuse and security experts. As a result of broad discussion with multiple stakeholders such as Anti-Phishing Working Group (APWG), Registry Internet Safety Group (RISG), the Security and Stability Advisory Committee (SSAC), Computer Emergency Response Teams (CERTs), members of the financial, and Internet security communities, ICANN proposed 9 safeguards [186, 187].

The first safeguard mandated that all new gTLD registry operators provide descriptions of the technical back-end services to ensure their technical competence (vetting registry operators). The second safeguard requires all new gTLD registries to implement DNSSEC at the root level. The third safeguard prohibits domain wildcarding to ensure that domains resolve for an exact match and do not redirect users for non-existent domain names. The fourth safeguard requires new gTLD registries to remove orphan glue records when it is proved that such records have been used in malicious activity. For the fifth safeguard, operators have to create and maintain "Thick WHOIS" records, i.e. complete WHOIS information from all the registrars on all domains cooresponding to a given new gTLD. New gTLD operators are also required to make their zone files available to approved requestors via the Centralized Zone Data Service (CZDS)[6]. The agreement mandates all new gTLD registry operators to document abuse contact details for registries and registrars on their websites. The agreement also obliges operators to respond to security requests to address security threats but do not define specific

[5] With a few exceptions such as .allfinanz or .forex brand gTLDs for which the sunrise period has been announced and ended.

[6] https://czds.icann.org/en

procedures for doing so. The ninth safeguard proposed to create a framework for a "high security zone verification program", however, due to a lack of consesus this safeguard has never been implemented.

The role of safeguards in the new gTLD program is critical since a broadened domain name space creates new opportunities for cybercriminals. The majority of the existing safeguards however, may not directly prevent domain abuse. For example, DNSSEC is intended to increase the security of the Internet by adding authentication to DNS resolution to prevent attacks such as DNS spoofing [67] rather than, for example, preventing legitimate domains from being hacked. We agree that making the zone files of new gTLDs open to security research may indirectly contribute to improving security of new gTLD domain space. It does not, however, prevent miscreants from registering domains for malicious purposes.

As it may be difficult to statistically measure the effects of the existing safeguards individually, we opt for a rigorous approach to assess their joint effects on domain abuse rates.

5.2.4 Related Work

Numerous studies have looked into discovering, predicting, or explaining abuse across the DNS ecosystem [39, 170, 188–192]. In addition to those, there are other studies that investigated domain re-registrations patterns and their relation with domain abuse [193–196]. For example, Lever *et al.* studied the maliciousness of domains before and after re-registration with a focus on when malicious behavior occurs. Their findings showed hundred thousands of expired domains that were maliciously re-registered [195].

When it comes to quantifying the impact of specific factors that influence security of gTLDs, in particular new gTLDs, there exists very little empirical work. Rasmussen and Aaron regularly release APWG Global phishing reports in which they examine phishing datasets collected by APWG and several other supplementary phishing feeds. Recently, they concluded that phishing in the new gTLDs is rising but is not yet as pervasive as it is in the domain space as a whole [197]. Halvorson *et al.* found that new gTLD domains are more than twice as likely as legacy TLDs to appear on a domain blacklist, within their first month of registration [182]. Vissers *et al.* studied large-scale malicious campaigns in the `.eu` TLD for a period of 14 months and observed that 80% of the malicious registrations are part of just 20 long-running campaigns. Moreover, out

of all domains operated by these campaigns, 18% never appeared on any blacklist [198].

Previous literature highlighted the importance of reliable security metrics to estimate abuse rates across network players in the domain ecosystem such as hosting providers or Autonomous Systems [40] and discussed specific factors that can influence this concentration of abuse [199, 200]. For the case of TLD operators, Korczyński *et al.* designed security metrics to measure and benchmark entire TLDs against their market characteristics [201]. They found that next to TLD size, abuse primarily correlates with domain pricing (free versus paid registrations), efforts of intermediaries (measured through the proxy of their DNSSEC deployment rate), and strict registration policies [201].

We build on the existing work in several ways. First, we analyze and compare the distribution of abuse across new and legacy gTLDs. Next, we make the first attempt to develop a comprehensive approach that can statistically quantify the impact of operator security indicators along with the structural properties of new gTLDs on DNS abuse rates.

5.3 Measurement datasets

In this section we cover six types of datasets used in this research: abuse feeds, `WHOIS` records, DNS zone files, active web scans, DNS scans, and passive registry data.

5.3.1 Abuse Feeds

To assess the prevalence of maliciously registered[7] and compromised domains[8] per gTLD and registrar, we use 11 distinct abuse feeds. These represent malware, phishing, and spam abuse and have been generously provided to us by Spamhaus [202], APWG [97], StopBadware [96], SURBL [203], the Secure Domain Foundation (SDF) [204], and CleanMX [205]. All six reputable organizations provide abused domain or URL data feeds employed in operational environments. **Spamhaus** data contains domains with low reputation collected from spam payload URLs, spam senders and sources, known spammers, phishing, virus, and malware-related websites [206]. **APGW** contains black/white listed phishing URLs submitted by accredited users through the eCrime Exchange (eCX) platform. The **StopBadware** Data Sharing Program (DSP)

[7] Domains registered by miscreants for the purpose of malicious activity
[8] Domains exploited using vulnerable web hosting

feed consists of abusive URLs shared by ESET, Fortinet, and Sophos security companies, Internet Identity, Google's Safe Browsing appeals results, the StopBadware community, and other contributors [207]. **SURBL ph** is a phishing domain blacklist comprised of data supplied by among others MailSecurity, PhishTank, OITC phishing, PhishLabs, US DHS, NATO [208]. The **SURBL jp** blacklist contains domains analyzed and categorized as spam (e.g. unsolicited) by jwSpamSpy software, traps, and participating mail servers. **SURBL ws** contains mainly spam domains from SpamAssassin, the Anti-Spam SMTP Proxy, as well as information from other data sources including internal and external trap networks. The **SURBL mw** feed contains data from multiple sources that cover malicious domains used to host malware websites, payloads or associated redirectors [208]. The **SDF** contains domains and URLs classified as phishing or malware. The domain names were queried against the Secure Domain Foundation's Luminous API which aggregates data from open source blacklist feeds and registrar suspension lists [204]. Note that unlike the other data feeds the SURBL and SDF feeds cover the 2,5-year study period between July 2014 and December 2016. Finally, **CleanMX** contains three URL blacklists identifying phishing, malware websites, as well as a "portals" category that contain defaced, spamvertized, hacked, and other types of abused websites. Table 5.1 shows the number of unique blacklisted 2^{nd}-level domain names per feed. In Appendix 5.8, we further discuss the overlap among blacklists.

Table 5.1: Overview of blacklists: unique blacklisted gTLD domain names for the StopBadware SDP, APWG, Spamhaus, SDF, CleanMX, and SURBL datasets for 2014, 2015, 2016.

Year	SB	APWG	Spamhaus	SDF	CleanMX ph	CleanMX mw
2014	403,347	60,681	1,901,970	41,094	68,523	169,237
2015	501,982	139,538	2,505,407	142,285	98,112	117,140
2016	502,579	83,215	3,944,684	110,687	138,869	149,632
Year	CleanMX pt	SURBL ph	SURBL mw	SURBL ws	SURBL jp	
2014	205,051	68,208	289,664	1,229,698	1,484,807	
2015	124,608	134,591	220,073	1,813,858	2,475,745	
2016	68,413	173,326	106,819	2,023,178	2,442,592	

Note that some of the aforementioned feeds contain data at the URL level while others at the domain level. The distinction is important from an operational level. While some domain names that appear in *URL blacklists* are registered by miscreants for malicious purposes only, the majority of domain names are compromised domains,

i.e. they were registered by legitimate users and hacked (see e.g.: phishing survey [209]). From the operational point of view blocking the domain name element of a blacklisted URL might harm legitimate operations. With this in mind, Spamhaus and other data providers maintain *blacklists of domain names* and perform extensive checks to prevent legitimate domain names from being listed. Therefore, the domain blacklists can be used by production systems to, for example, block emails that contain malicious domain names. In this paper, we refer to both domain names that appear in the domain blacklists and as part of of blacklisted URLs as "abused domains" or "blacklisted domains".

5.3.2 `WHOIS` Data

Most of the abuse feeds used for this study contain no additional domain name attributes such as registrar name or date of registration. We obtained these attributes via `WHOIS` databases covering the 3-year study period provided by Whois XML API [210] and DomainTools [211]. These databases contain `WHOIS` information for the domains of the aforementioned 18 legacy gTLDs and for the domain names of the 1,196 new gTLDs that had been delegated during our study period [212].

We extract <`domain, registrar name`> tuples from `WHOIS` data and use these in conjunction with our abuse feeds to map domain names or the domain element from abused URLs to a sponsoring registrar. The registrar name is used to determine the amount of abuse related to the registrar. We also extract the <`domain, creation date`> tuples and use these to determine if the domain has been maliciously registered or compromised.

5.3.3 DNS Zone Files

The sizes of gTLDs vary significantly. In order to provide a fair comparison criteria across gTLDs, we need to take into account their size, i.e., the number of domains registered. To do that, we processed daily zone files (containing all domains for each gTLD on a given date) for the 3-year study period. The rate of abuse, i.e. X number of blacklisted domains over the Y number of total registrations provides a more fair comparison criteria across gTLDs.

To give an idea of this difference, we show in Figure 5.1 a time series of unique domain names under legacy and new gTLDs. As can be seen, the legacy gTLDs still

account for the majority of registrations (160.9M vs 24.5M in Q4 2016).

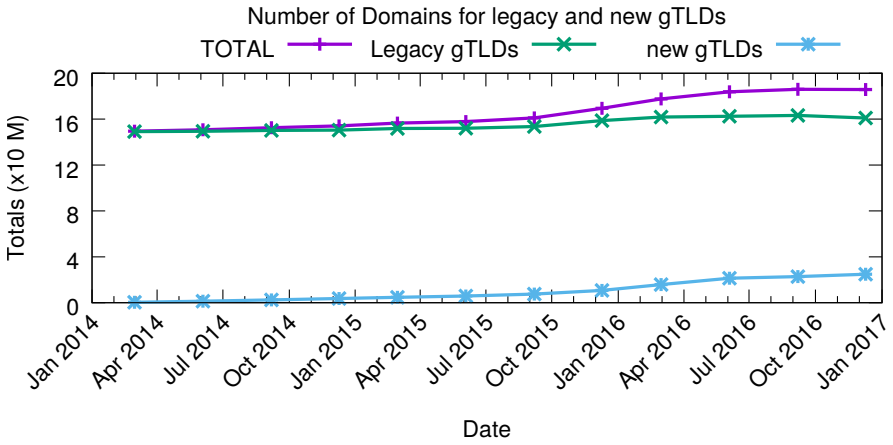


Figure 5.1: Zone file sizes for legacy and new gTLDs

We also relied upon zone files to determine the number of DNS Security Extensions (DNSSEC)-signed domains for each gTLD. One of the new gTLD program safeguards requires that all new gTLD applicants have a specific plan for DNSSEC deployment [187]. We used this data in our inferential analysis (see §5.5.2). Using regular expressions we matched `DS` records in the zone files and counted the distinct number of domains with `DS` records. The `DS` record is kept in the parent (TLD) zone and is used to prove the validity of cryptographic DNSSEC chain. Presence of a `DS` record indicates that the domain supports DNSSEC.

5.3.4 Active Web Scan

Using our web measurement platform, we crawled each new gTLD domain found in the zone files generated on May 2, 2017 (24,2M domains). We crawled these domains to determine how many are active and hosting content (see §5.4.2 for more details). The number of legacy gTLD domain names proved too voluminous to scan for this study. Therefore, we created a representative sample of 16,7M domain names (from the same date) to scan, using stratified sampling. A domain was considered non-responsive, if fetching `www.example.com` or `example.com` respectively, returned an error. If our crawler detects a redirect in either the retrieved HTML code or the HTTP headers then these redirects are followed. Any domain resulting in a crawl chain of more than 5 redirects is also marked as non-responsive.

The crawler is designed to have a minimal impact on the servers that are crawled. For this reason only the main page is retrieved. The data captured for each domain

includes the HTML code, HTTP headers and status codes. To determine if a domain is parked, the HTML code is analyzed using pattern matching to search for strings, which might indicate that the domain is for sale. The crawler also looks for URLs that are linked to known parking service providers.

5.3.5 Active DNS Scan

During the domain scan process we also queried the DNS system to retrieve the `A`, `AAAA` and `SOA` records for each domain to detect active domains serving content (see §5.4.2). The DNS crawler sends queries to a dedicated instance of the unbound DNS resolver to check whether domains resolve. Moreover, the `SOA` record is indicative of whether the primary authoritative name server for the domain is linked to a known parking services provider.

5.3.6 Passive Data for Registries

In this study, we analyzed new gTLDs whose domain names became available for public registration within the study period. The time between the delegation of a new gTLD and the end of its sunrise period might take several months[9]. Consequently, our analysis includes new gTLDs after their respective sunrise periods. This data, provided by ICANN via their public portal [212], contains 522 new gTLDs with sunrise periods ending during the timeframe of the study.

We also used a list of registry operators, their affiliates, and associated new gTLDs provided to us by ICANN. We mapped gTLDs to related registry operators regardless of which name they were operating under. We used the mapping of parent companies of registry operators and the corresponding new gTLDs in our inferential analysis as a proxy for registration practices.

Relying upon ICANN's categorizations of new generic, community, geographic, and brand gTLD registry applications, we conducted an inferential analysis on registration restrictions. We assigned registration "levels" to new gTLDs, from the least to most restricted groups: 1 generic, 2 geographic, 3 community, and 4 brand. Intuitively, while generic gTLDs are normally unrestricted and open for public registration, registration policies of community or brand gTLDs are strict and may prevent miscreants from

[9]E.g. delegation of `.zuerich`: December 25, 2014 [213], zone file seen for the first time: January 1, 2015, sunrise period termination: June 5, 2017 [212]

malicious registrations.

5.4 Methodology

5.4.1 Security Metrics

To determine the distribution of abusive activities across the gTLDs and registrars, we analyze the occurrence of *unique abused domains*. Previous research has also proposed two complementary security metrics, i.e. the number of *unique fully qualified domain names* (*FQDNs*) and *unique blacklisted URLs* aggregated by TLDs [201]. However, due to space constrains, we do not present our results for such additional metrics.

5.4.2 Size Estimate of TLDs

In order to have a fair comparison criteria, we normalized the number of reported domains from blacklists (Table 5.1) by the size of their respective TLD. We calculated the size of each gTLD by counting the number of 2^{nd}-level domains present in a zone file for each gTLD at the end of an observation period. We used zone files as they are the most accurate source to determine gTLD sizes. An alternative would be to use the ICANN monthly reports that summarize domain activity for all registered domains [214]. This would however result in an over counted gTLD size since some registrants register domain names but do not associate them with name servers.

The TLD size can also be used as an explanatory factor for the concentrations of abused domains, as indicated in the previous research [40, 200, 201]. However, it is unclear what portion of the domain names are in use and serve content. Halvorson *et al.* have shown that in 2015 as many as 16% of domain names in new gTLDs with `NS` records did not resolve [182]. Using our Web and DNS crawling platform, we performed a new scan and classified each domain as belonging to one of five groups: *i) No DNS* domains that do not resolve when queried by our DNS crawler, *ii) Parked* domains that are owned by parking services, advertisement syndicators, and advertisers. We follow the classification methodology outlined by Vissers *et al.* [172], *iii) HTTP Error* domains for which authoritative name servers return valid responses but the corresponding websites do not return an HTTP 200, *vi) Redirect* domains are redirected to a different domain, and *v) Content* domains that serve valid Web content.

Figure 5.2 shows the categorization results for all domains in the new gTLDs and

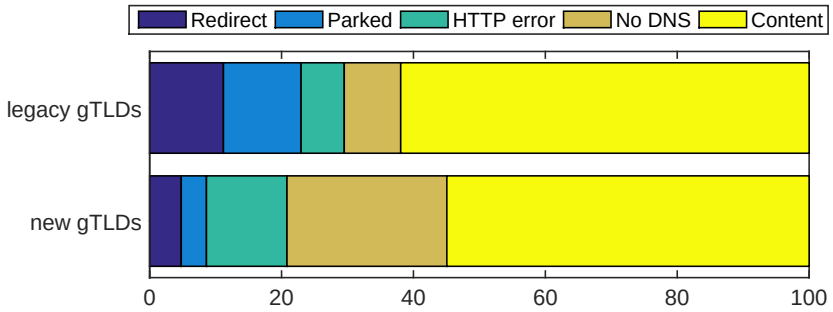


Figure 5.2: Categorization for all domains in the new TLDs and a random sample of the legacy TLDs.

a random sample of the legacy gTLDs. Interestingly, there is a significant increase in erroneous domains in the new gTLDs ("No DNS" and "HTTP Error" categories) as compared to legacy gTLDs. "No DNS" domains account for about a quarter of all domains (24.2%), whereas domains for which the corresponding websites serve an HTTP error account for another 12.2%.

Note that we use this measurement data in the inferential analysis to adjust measured TLD sizes. Intuitively, only the domains serving content are exposed to certain types of vulnerabilities and can be hacked. On the other hand, parked domains may be used to scam users or to distribute malware. One might therefore expect a positive correlation between the number of parked and maliciously registered domains.

5.4.3 Size Estimate of Registrars

Since we are interested in comparison between registrars, we calculated their sizes from the `WHOIS` data by counting the number of distinct domain names linked to each registrar name. Note that the `WHOIS` data may contain multiple name variants for a single registrar. E.g.:, GoDaddy is listed as a registrar using 52 variations, such as "GODADDY.COM, LLC", "GoDaddy.com, LLC (R91-LROR)" and "GoDaddy.com, Inc.". Therefore, we need an additional entity resolution step to group together all the different registrar name variants as a single registrar.

We also used the IANA Registrar ID, which is assigned to ICANN accredited registrars [215]. We automatically matched the list of registrar names against names found in the `WHOIS` data. Then, we manually mapped the remaining registrar variants. To determine the amount of abuse related to a registrar, we mapped each domain name found in an abuse feed to its respective registrar using the `WHOIS` records with the closest enclosing time-window.

5.4.4 Compromised Versus Maliciously Registered Domains

Miscreants can both register or compromise and abuse legitimate domains. To distinguish between compromised and maliciously registered domains, we build on three heuristics previously used in domain abuse surveys (e.g. phishing survey by Aaron and Rasmussen [197]). More specifically, we label a domain as maliciously registered if it was involved in criminal activity within a relatively short time after its registration or if it contains a brand name or a misspelled variant of brand name. We refer the reader to Appendix 5.9 for more details on the methodology used in our study.

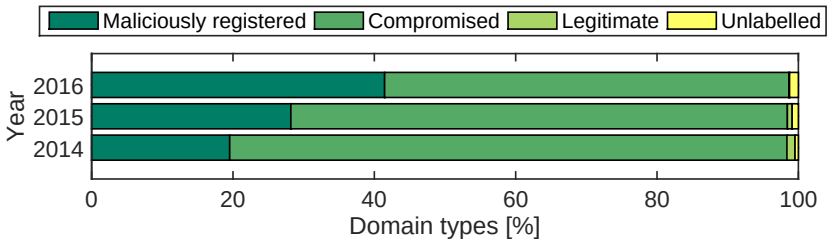


Figure 5.3: Categorization results: the fraction of maliciously registered, compromised, legitimate, and unlabelled domains for **APWG** feed.

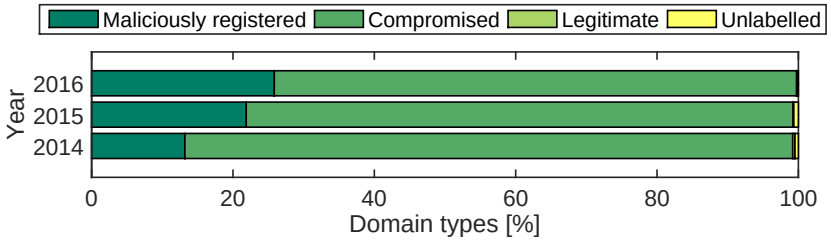


Figure 5.4: Categorization results: the fraction of maliciously registered, compromised, legitimate, and unlabelled domains for **StopBadware** feed.

Figure 5.3 and Figure 5.4 show the categorization of domains blacklisted by APWG and StopBadware respectively during the study period (2014, 2015, and 2016). Note that up to 1.1% of all domains submitted to the APWG have been pre-filtered based on the maintained list of domains corresponding to legitimate services and labeled as "legitimate". For comparison, we have excluded less than 0.3% of the StopBadware domains. A previous study showed that domains of legitimate services are often misused by miscreants to distribute malware or used in phishing campaigns [201]. However, some may also represent legitimate domains that were incorrectly blacklisted.

The results indicate that 78.8% of abused phishing and 86% of malware domains (listed on URL blacklists in 2014) were compromised by criminals (see Figure 5.3 and

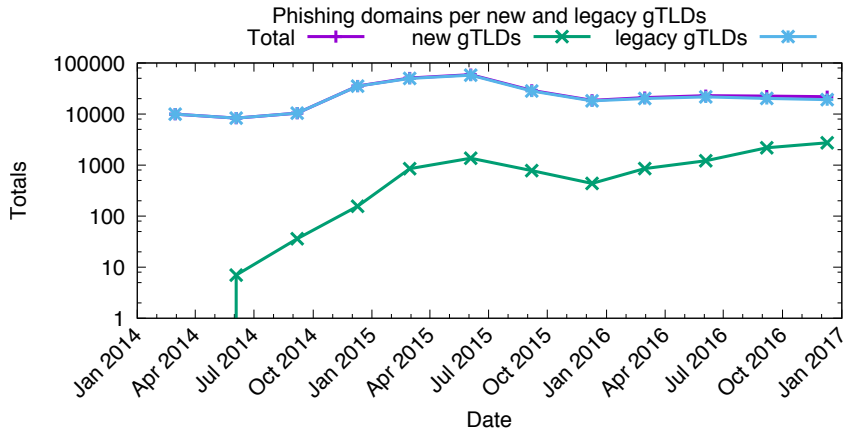


Figure 5.5: Time series of counts of phishing domains in **legacy** gTLD, **new** gTLDs, and **all** gTLDs (Total) based on the Anti-Phishing Working Group feed (APWG, 2014-2016, Table 5.1). Log scale on y axis.

Figure 5.4). In 2016, those percentages were smaller: 57.2% and 73.9% of phishing and malware domains were labeled as compromised. Although domains listed in URL blacklists are predominantly compromised, their number has been gradually decreasing. Instead, miscreants are registering domain names more often. We find that 19.5%, 28.2%, 41.5% and 13.2%, 21.9%, 25.8% (in 2014, 2015, and 2016) of all phishing and malware domains respectively were presumably maliciously registered by miscreants. This trend suggests a shift in the behavior of miscreants that over time seem to prefer registering rather than compromising legitimate domains.

5.5 Results

5.5.1 TLD Reputation

5.5.1.1 Phishing Abuse

Figure 5.5 plots a time series of the number of phishing domains for new gTLDs, legacy gTLDs, and a "Total" number for our 2014–2016 study period based on data from the APWG feed. We aggregate phishing incidents on a quarterly basis and present counts using a logarithmic scale. We observe that the total number of phishing domains (purple line) overlaps largely with the number of phishing domains in legacy gTLDs. This phenomena is due to the disproportionate market share of domain names registered in legacy gTLDs. While the number of abused domains remains relatively constant in legacy gTLDs, we observe a clear upward trend in the absolute number of phishing

domains in new gTLDs. We observe similar trends in SURBL phishing and CleanMX phishing datasets (which have been ommitted due to space contrains).

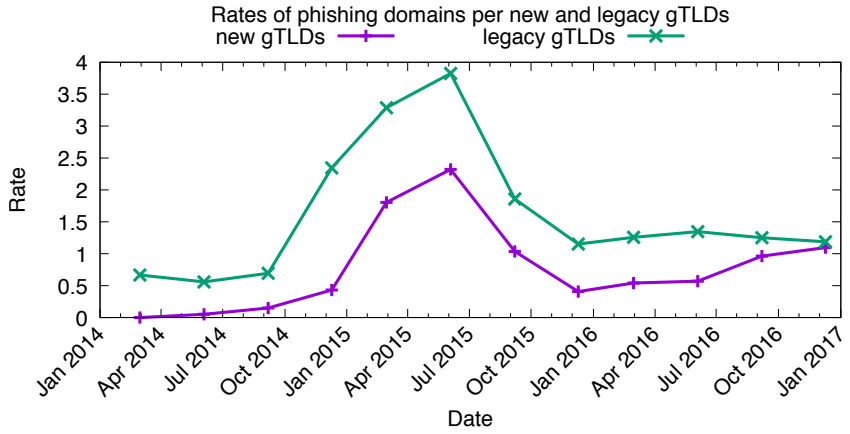


Figure 5.6: Time series of abuse rates of phishing domains in **legacy** gTLDs and **new** gTLDs based on the APWG feed (2014-2016). $Rate = 10,000 * \#blacklisted\ domains/\#all\ domains$.

5.5.1.2 Normalized Phishing Counts

As previously discussed, reliable reputation metrics must account for market shares (i.e. size) as larger market players may experience a higher amount of domain abuse. Figure 5.6 shows a time series of abuse rates of phishing domains for legacy gTLDs and new gTLDs based on the APWG feed (due to space limitation we do not present figures related to abused CleanMX phishing and SURBL phishing domains).

Here, abuse rates are presented on a linear scale. E.g., in the second quarter of 2015 the domain abuse rate for legacy gTLDs is equal to 3.82503. This means that, on average, legacy gTLDs had 3.8 blacklisted phishing domains per 10,000 registered domains. Our results suggest phishing abuse rates in legacy and new gTLDs to be converging towards similar values over time and were almost equal the end of 2016.

5.5.1.3 Phishing: Compromised vs Maliciously Registered

Up to this point, our descriptive statistical analysis of phishing abuse rates in the new and legacy gTLDs has conflated compromised and maliciously registered domains. Now, we compare abuse rates for these two types, separately.

Figure 5.7 plots abuse rates for compromised phishing domains within legacy gTLDs and new gTLDs, based on the APWG feed over time. The curves corresponding to all blacklisted phishing domains and compromised phishing domains of legacy gTLDs (cf.

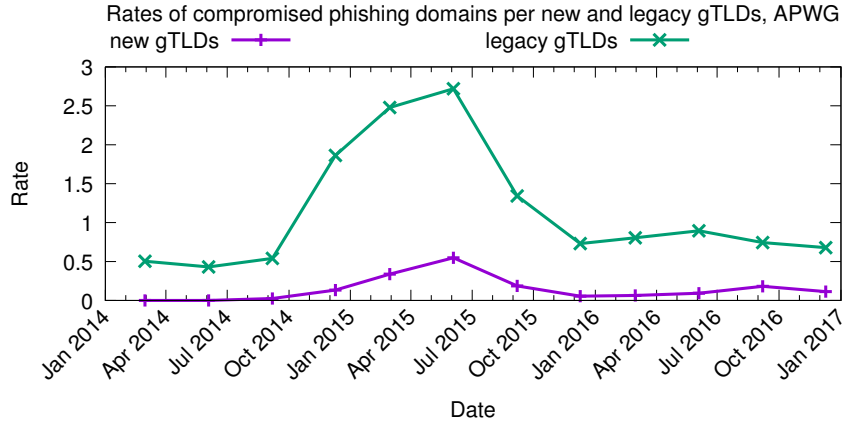


Figure 5.7: Time series of abuse rates of **compromised** phishing domains in **legacy** gTLDs and **new** gTLDs based on the APWG feed (2014-2016). $Rate = 10,000 * \#compromised\ domains/\#all\ domains$.

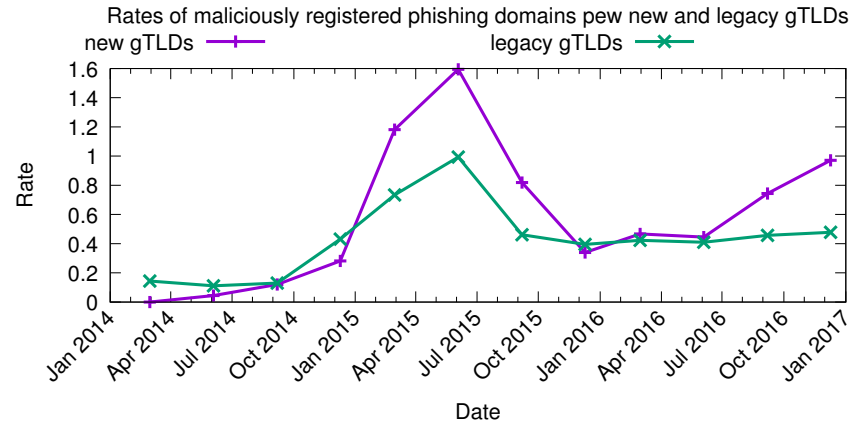


Figure 5.8: Time series of abuse rates of **maliciously** registered phishing domains in **legacy** gTLDs and **new** gTLDs based on the APWG feed. $Rate = 10,000 * \#maliciously\ registered\ domains/\#all\ domains$.

Figure 5.6 and Figure 5.7) follow a similar pattern due to a disproportionate concentration of compromised domains in legacy gTLDs.

Figure 5.8 on the other hand, shows abuse rates for maliciously registered phishing domains in the legacy and new gTLDs in APWG feed over time. When comparing the rates of all blacklisted domains of new gTLDs with rates of maliciously registered domains (cf. Figure 5.6 and Figure 5.8), we conclude that (despite higher relative concentrations of compromised domains in legacy gTLDs) miscreants more frequently choose to maliciously register domain names using one of the new gTLDs.

Moreover, we observe relatively higher rates of maliciously registered domains in new gTLDs in the first three quarters of 2015. We find 616 abused new gTLD domains. We observe 182 and 111 abused `.work` and `.xyz` domains, respectively. Manual inspection

indicates that the majority of .work domains were registered by the same person: 150 domains were registered on the same day using the same registrant information, the same registrar, and the domain names were composed of similar strings.

Attackers often seem able to maliciously register strings containing trademarked words. Manual analysis of maliciously registered domains in the fourth quarter of 2015 revealed 88 abused .top domains 75 out of which contain the words: Apple, iCloud, iPhone, their combinations, or misspelled variants of these strings suggesting that they may have been all used in the same phishing campaign against users of Apple Inc. products.

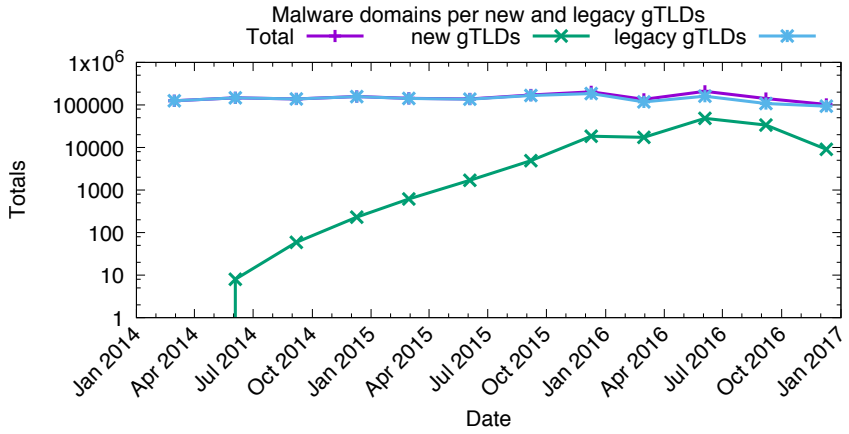


Figure 5.9: Time series of counts of malware domains in **legacy** gTLD, **new** gTLDs, and **all** gTLDs (Total) based on the StopBadware DSP feed.

5.5.1.4 Malware Reputation

Having examined phishing abuse, we now analyze the malware related activity.

Figure 5.9 presents a time series of the number of malware domains in legacy gTLD, new gTLDs, and a "Total" based on the StopBadware feed between 2014 and 2016. Similar to phishing abuse, the total number of malware incidents in all gTLDs is mainly driven by incidents in legacy gTLDs (88.6%). We observe that the number of abused malware domains in legacy gTLDs remains relatively constant, whereas a growing trend in the number of malware domains in new gTLDs is clearly visible. SURBL mw and CleanMX malware datasets (not presented due to space limitation) confirm this observed trend.

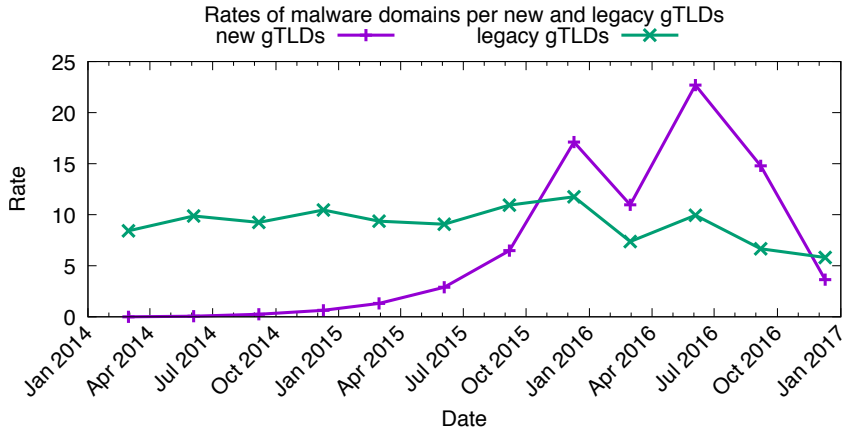


Figure 5.10: Time series of abuse rates of **malware domains** in legacy gTLDs and new gTLDs based on the StopBadware DSP feed (2014-2016). $Rate = 10,000 * (\#blacklisted\ domains/\#all\ domains)$.

5.5.1.5 Normalized Malware Counts

We now account for gTLD market shares by constructing a time series of abuse rates of malware domains in legacy and new gTLDs based on the StopBadware feed (see Figure 5.10). As before, the abuse rates are presented on a linear scale. Here, we observed an exponential growth of malware domain abuse rates in the new gTLDs up to the first quarter of 2016. Differences between malware abuse rates in legacy and new gTLDs is the most prominent in the second quarter of 2016. While legacy gTLDs collectively had a malware-domains-per-10,000 rate of 9.9, the new gTLDs experienced a rate of 22.7. In absolute terms, malware domains in new gTLDs constitute 23% of all gTLD domains blacklisted by StopBadware during this period. SURBL and CleanMX malware datasets confirm the growing trend in terms of the malware rates in new gTLDs in comparison to legacy gTLDs.

5.5.1.6 Malware: Compromised vs Maliciously Registered

To distill factors that drive higher abuse rates in new gTLDs, in our analysis, we will differentiate between maliciously registered and compromised domains as we did for phishing abuse. Figure 5.11 and Figure 5.12 plot time series of abuse rates of compromised and maliciously registered malware domains, respectively, in legacy and new gTLDs. The results suggests that similar to phishing, malware abuse rates in legacy gTLDs are mainly driven by compromised domains (cf. Figure 5.10 and Figure 5.11). As expected, the malware abuse rates for new gTLDs are driven by maliciously registered

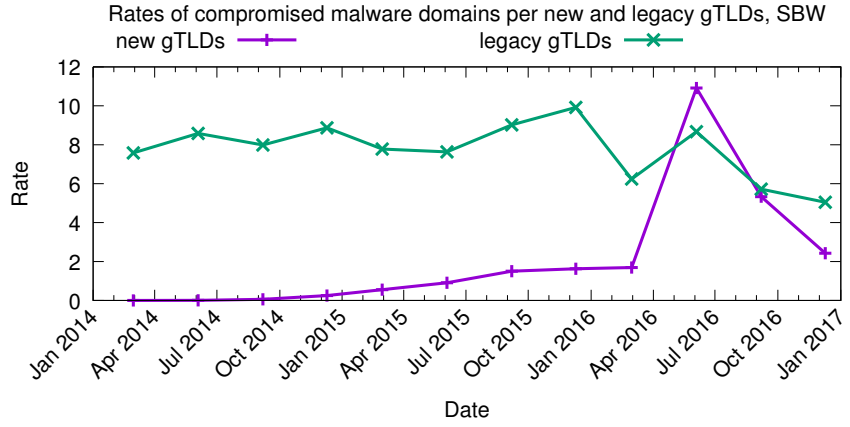


Figure 5.11: Time series of abuse rates of **compromised malware domains** in legacy gTLDs and new gTLDs based on the StopBadware DSP feed. $Rate = 10,000 * \#compromised\ domains/\#all\ domains$.

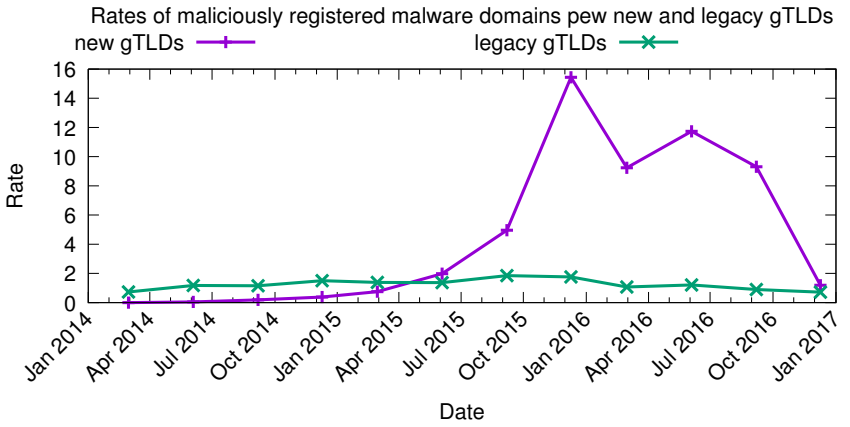


Figure 5.12: Time series of abuse rates of **maliciously registered malware domains** in legacy gTLDs and new gTLDs based on StopBadware DSP. $Rate = 10,000 * \#maliciously\ registered\ domains/\#all\ domains$.

domains (cf. Figure 5.10 and Figure 5.12).

Manual analysis of maliciously registered domains reveals distinctive common patterns in domain names. For example, we find 9,376 `.link` domains of which 9,256 were created in the first quarter of 2016 and 9,253 were registered through the Alpnames Limited registrar. 8,381 of all `.link` domains were registered using two registrar names only. Moreover, 8,205 and 1,027 were composed of 5 and 6 randomly generated characters, respectively. We created a user account with Alpnames Limited and tested bulk domain registration options. In fact, it is possible to randomly generate up to 2,000 domains at once from the selection of 27 new gTLDs using different patterns like letters, time, cities, zip codes, etc.

Finally, note that the registries of the most abused new gTLDs such as `.win`, `.loan`,

.top, and .link compete on price, and in 2016 their registration prices were occasionally below US $1, which was lower than the registration fee of a .com domain.

5.5.1.7 Spam Reputation

The results of the spam activity in the new and legacy gTLDs reveal very surprising trends. Due to space limitation, we only present our analysis of the Spamhaus feed. Note that Spamhaus provides *domain* rather than *URL* blacklists, which means that the great majority of listed domains are maliciously registered. Figure 5.13 presents a time series for the number of spam domains observed in legacy gTLDs, new gTLDs, and the total number of spam domains. While we observed an upward trend in the number of *phishing* and *malware* domains in new gTLDs, in contrast the absolute number of malicious *spam* domains in new gTLDs was actually higher than in legacy gTLDs. Note that the total number of spam incidents in all gTLDs is relatively constant and in the Q4 2016 is mainly driven by incidents in new gTLDs (58.8%). Figure 5.19 and Figure 5.20 (see section 5.10), presenting spam domains in legacy and new gTLDs for SURBL ws and SURBL jp spam datasets confirm this observed trend. The results suggest an alarming trend that miscreants seem to be switching from abusing legacy to new gTLDs when it comes to spam domains.

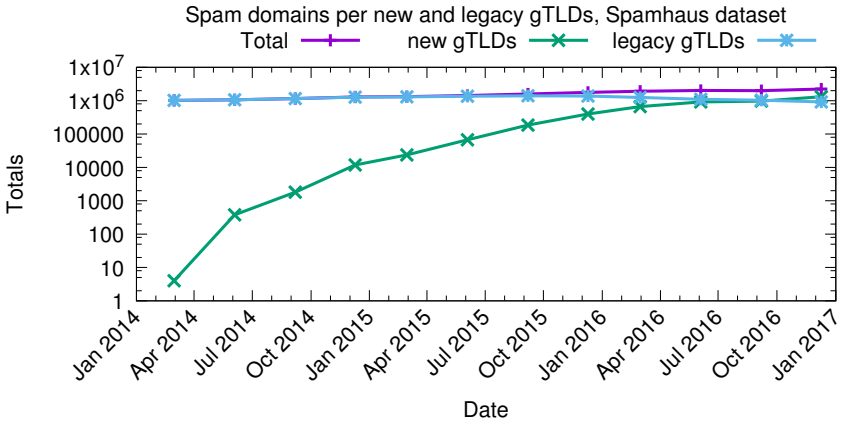


Figure 5.13: Time series of counts of blacklisted domains in **legacy** gTLD, **new** gTLDs, and **all** gTLDs (Total) based on the **Spamhaus** feed.

5.5.1.8 Normalized Spam Counts

Figure 5.14 plots a time series of spam domain abuse rates for legacy gTLDs and new gTLDs based on the Spamhaus feed. As expected, the difference between spam abuse

rates in legacy and new gTLDs is quite prominent. While legacy gTLDs collectively had a spam-domains-per-10,000 rate of 56.9, in the last quarter of 2016, the new gTLDs experienced a rate of 526.6–which is almost one order of magnitude higher. When comparing abuse rates based on our SURBL jp and SURBL ws spam feeds in the same period we observed a spam-domains-per-10,000 rates of 46.6 and 26 for legacy gTLDs, whereas for new gTLDs the spam-domains-per-10,000 rates are 286.3, and 265.2, respectively.

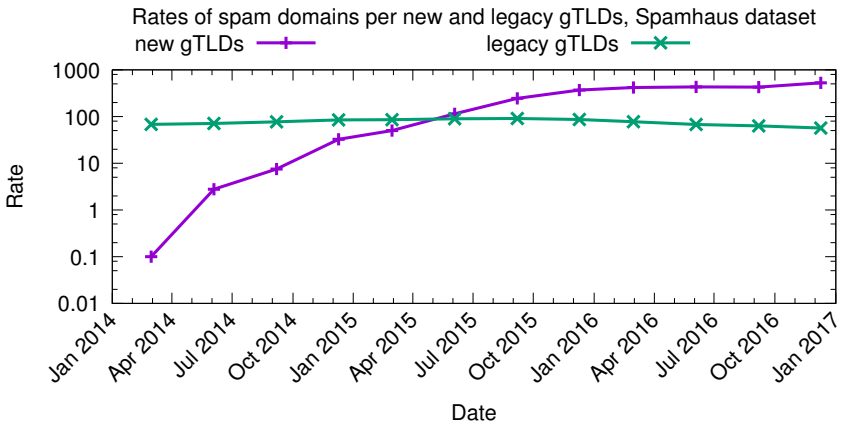


Figure 5.14: Time series of abuse rates of blacklisted domains in **legacy** gTLDs and **new** gTLDs based on the **Spamhaus** feed (2014-2016). $Rate = 10,000 * \#blacklisted\ domains/\#all\ domains$.

Table 5.2 (see the Appendix section) lists the top 10 new gTLDs with the highest relative concentrations of blacklisted domains for selected feeds in the fourth quarter of 2016. For example, spam-domains-per-10,000 registration rates calculated using the Spamhaus feed for `.science`, `.stream`, and `.study` are equal to 5,154, 4,756 and 3,343, respectively. In other words, as many as 51.5%, 47.6% and 33.4% of all domains in the corresponding zones were abused by cybercriminals and blacklisted by Spamhaus. Note that our results clearly indicate that the problem is not caused by just a few abused new gTLDs. As many as 15 most abused new gTLDs had spam-domains-per-10,000 registration rates calculated using Spamhaus feed higher than 1,000 at the end of 2016. Does this problem affect all new gTLDs? No. Our analysis of Spamhaus and SURBL blacklists reveals that approximately 32% and 36% of all new gTLDs available for registration did not experience a single incident in Q4 2016.

To conclude, while the number of abused domains in legacy gTLDs seem to remain relatively constant over time (or are decrasing), new gTLDs that underwent rigorous security analysis by ICANN are much more frequently affected by phishing, malware,

and especially spam activities. Despite the new safeguards a number of new gTLDs are more susceptible to DNS abuse in comparison to legacy gTLDs. Given these observations, we systematically analyze the potential factors driving DNS abuse in new gTLDs.

5.5.2 Inferential Analysis of Abuse in New gTLDs

Previous work used regression analysis to study the impact of factors that influence the variation of abuse counts across networks of different intermediaries such as hosting providers [200] or TLDs [201]. Examples of such factors or more specifically intermediary properties are size, pricing, domain popularity index, or security effort [200,201]. In this section, we aim to analyze and quantify the relationship between the collected new gTLD properties (independent variables), and abuse counts (dependent variable), at the level of gTLDs. In other words, we use regression analysis to examine the amount of variance that gTLD properties can collectively explain, out of the total observed variance in the abuse counts.

Our regression models in Table 5.3 are built using the datasets explained in §5.3.1. We model the number of abused domains as a dependent variable (i.e. blacklisted domains or domain name elements of blacklisted URLs) using negative binomial[10] generalized linear model (GLM) with a Log link function. Depending on the model, we use the total number of abused domains or treat maliciously registered and compromised domains separately (details follow later). The independent variables in the models are the following properties of new gTLDs: *"new gTLD size"*: number of domains in TLD, *"Parked"*: number of parked domains, *"No DNS"*: number of domains that do not resolve, *"HTTP Error"*: number of domains for which corresponding websites return an HTTP error, *"DNSSEC"*: number of DNSSEC-signed domains, *"Type"*: an integer corresponding to the type of new gTLD, from least to most restricted group: 1 generic, 2 geographic, 3 community, and 4 brand, *"Registry"*: name of the registry operator that the TLD is operating under.

Table 5.3 in the Appendix section contains the summary of the regression models, i.e., the estimated coefficients, and their significance levels together with the goodness-of-fit measures such as the maximum Log likelihood, θ values and minimum Akaike

[10] We choose negative binomial over Poisson due to the over-dispersion (unequal mean and variance) in our data.

information criterion (AIC) value (for more details, we refer the reader to the relevant literature). Note that the presented models are chosen from a stepwise addition of the variables into a baseline model with a single explanatory variable. Each column of the table contains a regression model for one of the abuse feeds with the count of abuse being the dependent variable.

The results in Table 5.3 are very consistent among all the analyzed abuse feeds. While all types of abuse show a positive and statistically significant correlation between the new gTLD size and abuse counts, the coefficients are very weak. We suspect that this is because the majority of abused domains in the new gTLDs are maliciously registered rather than compromised.

As expected, two variables indicating the number of domains that do not serve valid Web content to their users, i.e. "No DNS" and "HTTP Error" show a weak negative significant relationship with abuse counts. That means, the more domains labelled as "No DNS" and/or "HTTP Error", the less abused domains. Those two variables also correspond to the count of compromised domains rather than maliciously registered counts.

Moreover, the number of parked domains in new gTLDs plays a weak positive and statistically significant role in explaining the variance in phishing and malware domains. The more parked domains in a new gTLD, the more abused domains. This is to be expected as landing pages of parked domains may serve malware on a large scale. Note that the coefficients are very small. For example, if we hold the other independent variables constant and increase the number of parked domains by one unit (which is the equivalent to multiplying the number of parked domains of a gTLD by 10 since it is in the log_{10} scale), the number of phishing domains in APWG is multiplied by $e^{0.0003} = 1.0003$.

Previous research found a negative significant relation between the DNSSEC deployment and the count of phishing domains [201]. The authors used DNSSEC deployment as a proxy for the security efforts of both ccTLDs and gTLDs. In our analysis we test the relationship between the number of DNSSEC-signed domains and abuse counts using various types of blacklists for new gTLDs. Note that ICANN requires each new gTLD to demonstrate a plan for DNSSEC deployment to ensure integrity and utility of registry information. Therefore, in our analysis, the number of DNSSEC-signed domains cannot serve as a proxy for registry efforts and obviously it does not prevent

malicious registrations. One may suspect that attackers could be interested in deploying DNSSEC and signing their maliciously registered domains. Although it is not clear if that is the case, we indeed observe a weak but positive and statistically significant correlation between the number of DNNSEC-signed domains and the number of abused domains.

The regression results consistently show a negative correlation between the "Type" variable reflecting strict registrations and the count of phishing domains. In fact, in comparison to other variables, the obtained coefficients indicate the strongest statistically significant negative correlation for APWG, CleanMX phishing, and SURBL phishing datasets: -0.54, -0.4, and -0.76, respectively (see Table 5.3). Note that for all other considered datasets, in particular malware, we also observe negative but not statistically significant correlations. When we consider separately maliciously registered and compromised domains (models not presented due to space limitation) the "Type" of new gTLD plays a significant role in explaining phishing abuse counts only for malicious registrations. Again, the results are intuitive. For example, it is much easier to register domains in the `.top` *standard* gTLD than it is for the `.pharmacy` *community* gTLD, for which the registration policy restricts the sale of domains to legitimate pharmacies only.

We also considered other models that contain "Registry" as a fixed effect to capture systematic differences in the policies of registries across new gTLDs such as pricing, bulk registration options, etc. Interestingly, our results indicate that none of the registry operators have a statistically significant effect on the abuse counts.

5.5.3 Privacy and Proxy Services

In this section we present the results of an analysis to determine if there is a difference in the usage of WHOIS Privacy and Proxy services for abused domains in legacy gTLDs and new gTLDs. WHOIS Privacy and Proxy services are designed to conceal certain personal data of domain name registrants who use them. In practice this works by replacing the registrant information in `WHOIS` with the information of the WHOIS Privacy and Proxy service.

There are many legitimate reasons why someone may want to conceal possession of a domain name. The usage of a WHOIS Privacy and Proxy services by itself alone is, therefore not a reliable single indicator of malicious activity. A previous study by

National Physical Laboratories [216] did however find that a significant portion of abusive domains use Privacy and Proxy services.

There are numerous WHOIS Privacy and Proxy services available, which can be used by domain owners. In section 5.11, we describe the methodology used in this study to identify commonly used WHOIS Privacy and Proxy services.

To get an indication of how common WHOIS Privacy and Proxy service usage is, we aggregated all domains from the `WHOIS` data by their create date. This shows us the number of newly added domains per month for legacy and new gTLDs. After checking how many of these domains were using a Privacy and Proxy service when the domain was registered, we calculated what percentage of the total number of newly registered domains is using a Privacy and Proxy service (see Figure 5.15). We find that for legacy gTLDs the usage is stable with a mean of 24%, and a standard deviation of 1.6. For new gTLDs the usage is generally below that of legacy gTLDs with a mean of 18% and a standard deviation of 9.3, which is visualized by the larger spikes and the increase to above the level of legacy gTLDs near the end of the study period.

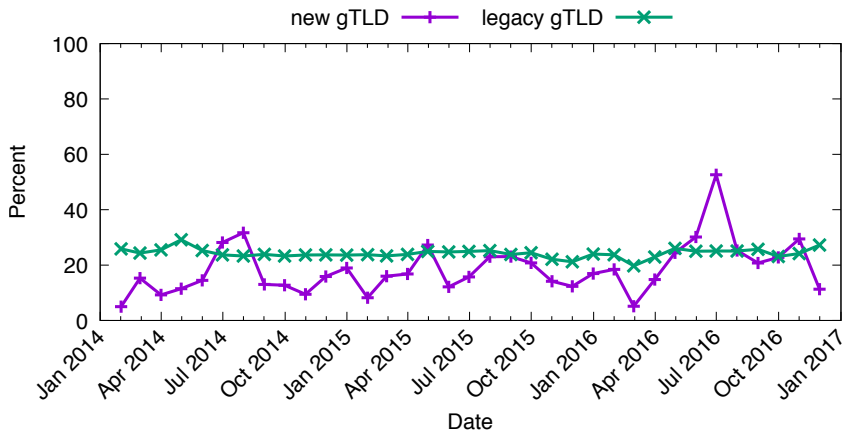


Figure 5.15: Usage percentage of Privacy and Proxy services for newly registered domains

Figure 5.16 shows the percentage of all newly created domains using Privacy and Proxy service, that have been reported to the Spamhaus or SURBL blacklist on or after the registration date. We have chosen to use Spamhaus and SURBL for this figure because these blacklists mainly contain maliciously registered domains. Here again, just as seen in Figure 5.15, we find that the variability for the new gTLDs is higher than compared to the legacy gTLDs.

For each blacklist used in this study we analysed the proportion of domains that were using a Privacy and Proxy service at the time the domain was found to be abusive

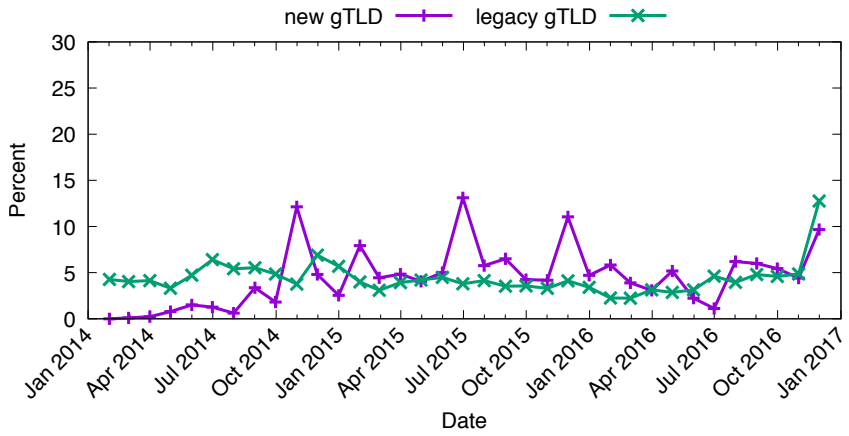


Figure 5.16: Percentage of abusive newly registered domains using Privacy and Proxy services

and included in the blacklist. Here again, we make a distinction between legacy and new gTLD domains.

For all SURBL feeds combined in 2016 the mean usage per month of privacy and proxy services by abusive domains in new gTLD observed is 5,874, with a standard deviation of 1,984, while for legacy gTLDs the mean usage per month is 21,744 with a standard deviation of 9,475. For Spamhaus the 2016 new gTLDs mean usage per month is 8,951 with a standard deviation of 2,892, while for legacy gTLDs the mean usage per month is 16,569 with a standard deviation of 3,843.

In the SURBL data we find 2 large peaks (see Figure 5.17) of abusive new gTLD domains using Privacy and Proxy services. Both of these peaks are driven by the `.xyz`, `.click` and `.link` new gTLDs. We attempted to find peaks in new registration that correspond to the two peaks seen in Figure 5.17. In the 7-15 day period leading up to a peak we do see an increase in the number of new registrations for the `.xyz`, `.click` and `.link` new gTLDs with the same registrar. However, we do not find strong evidence that the malicious registrations belong to a single or multiple campaigns using WHOIS Privacy and Proxy services.

The analysis of the use of WHOIS Privacy and Proxy service leads us to conclude that the usage of a WHOIS Privacy and Proxy services by itself is not a reliable indicator of malicious activity. Apart from the peaks, the usage of Privacy and Proxy services for abusive domains is not that high (see e.g. Figure 5.17). The usage of Privacy and Proxy seems to be higher in legacy gTLDs.

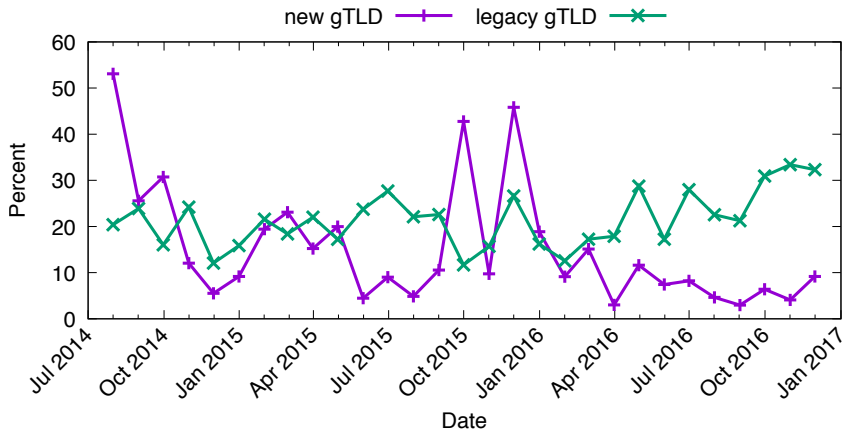


Figure 5.17: Usage of Privacy and Proxy services for abusive domains, reported by SURBL

5.5.4 Registrar Reputation

Here we present the distribution of abused domains across ICANN accredited registrars. For each registrar we find how many (#Incidents) can be attributed to the registrar and the total number of domains sponsored by that registrar (#Domains). We then calculate what proportion (Percentage) of all domains managed by the registrar is reported as abusive by a blacklist (see e.g. Table 5.4 in the Appendix section). An outlier with a relatively high rate compared to its peers may be caused by registrar-specific policies or operational practices.

Note, sinkholing of confiscated abusive domains or preventive registration of botnet C&C infrastructure domains is a common practice and special registrars have been created for this purpose e.g. "Afilias Special Projects" or "Verisign Security and Stability". These registrars have high numbers of abuse and have been filtered out during the analysis because they are not regular registrars.

Our analysis reveals that "Nanjing Imperiosus Technology Co. Ltd." is an outlier: over 93% of its domains are reported as abusive by SURBL (35,502, with a total number of 38,025 under its management) and 78% by Spamhaus (see Table 5.4). Figure 5.18 shows that both blacklists have marked domains managed by this registrar as abusive starting from early 2016. Starting from November 2016 we see a sharp decline in domains reported by Spamhaus and SURBL. This can be explained by the fact that ICANN has terminated the registrar accreditation [217] for this registrar, as it was determined that the registrar was in breach of the RAA [218]. Termination of the RAA had an effect on the amount of abuse linked to this registrar.

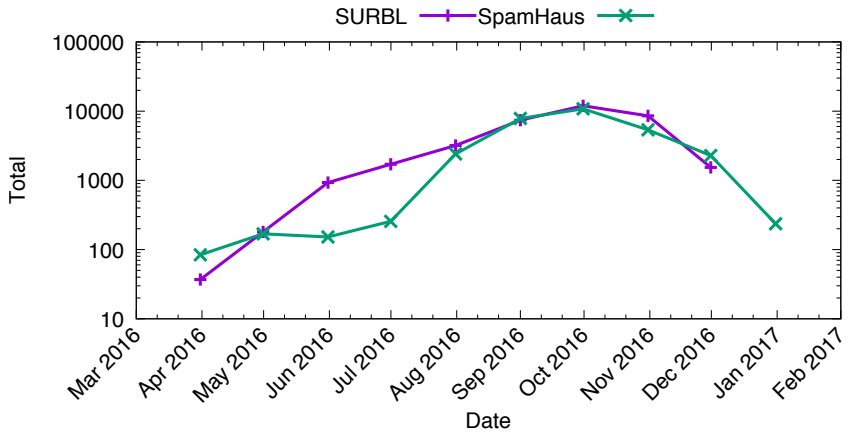


Figure 5.18: Abusive domains managed by Nanjing Imperiosus Technology

Alpnames Limited is another registrar that suffers from a high volume of abusive new gTLD domains reported by both Spamhaus and SURBL. The SURBL feed shows 2 distinctive peaks with a high number of abuse reports in 2016. After more detailed analysis, we find that these peaks correspond with 103,758 reports of abusive domains in the `.top` gTLD in August 2016. In October 2016 we find another peak, which is caused by 120,669 reports of abusive domains in the `.science` gTLD. In 2016 Alpnames did have promotions for domains using the `.science` gTLD for US $1 or less. We did not find corresponding peaks in the size of the `.top` and `.science` zone files, indicating the abusive domains have been registered over a longer period of time.

5.6 New Anti-Abuse Safeguards

Our results indicate that the implementation of the 9 anti-abuse safeguards have not effectively prevented domain name abuse in new gTLDs in comparison to legacy TLDs. Our findings, therefore, beg the question of whether more effective safeguards could be implemented by ICANN before the upcoming new gTLD rollout.

Our results suggest that lesser strict registration policies, low registration pricing, and the possibility of bulk domain name registration lower barriers to abuse. In addition, we observe that some of the more specific safeguards (e.g. DNSSEC deployment and prohibition of wildcarding) do not to raise barriers enough to prevent abuse. Yet, we cannot for example expect registries and registrars to raise registration prices to reduce abuse levels as this might be in conflict with their economic incentives. Alternatively, registries and registrars with disproportionately higher concentrations of abused

resources could be penalized while those with relatively lower concentrations could be financially rewarded, for example through lowered ICANN fees, to align incentives towards raising abuse barriers. This would also incentivize intermediaries to develop their own anti-abuse best practices while balancing their anti-abuse policies against their economic incentives and allow for self-regulation.

Our analysis of domain abuse across new gTLDs revealed that some distinct entities are (or have been) afflicted with significantly hight concentrations of abused resources. We observed large concentrations of blacklisted domains associated with Nanjing Imperiosus Technology in early 2016. ICANN has terminated its registrar accreditation in this case in early 2017. Yet at the time of writing this paper, registry operators of the most abused new gTLD (e.g. `.science`, `.stream` or `.racing`) still remain ICANN-accredited. Accreditation terminations may be effective penalizing factors.

That being said, existing safeguards mostly concentrate on individual complaints (e.g. removing orphan glue records) rather than on security reputation metrics. An alternative more effective path forward could be to introduce continuous monitoring of abuse rates (including that of domain resellers) and employing enforcement mechanisms such as immediate accreditation termination if the concentrations of abused domains are persistent and exceed certain levels.

Note that all above-mentioned proposals are currently under consideration by the ICANN community for upcoming new gTLDs rollout.

5.7 Conclusions

Since its inception, the new gTLD program has led to more than 1,200 strings being delegated in the root DNS zone, which greatly expanded the domain name space and increased consumer choice. We presented in this chapter the first comprehensive study comparing the rates of malicious and abusive behavior in the new and legacy gTLDs. To that end, we employed datasets from many sources, including zone files, domain `WHOIS` information, data obtained through our active measurements, and heterogeneous blacklists representing malware, phishing, and spam.

While the number of abused domains in legacy gTLDs seem to stay relatively constant over time (or in some cases decreasing), new gTLDs that underwent rigorous application and evaluation process by ICANN are more frequently affected by phishing, malware, and especially spam activities.

The systematic investigation of the relation between structural and security-related properties of new gTLD operators, and abuse counts has shown that the number of domains in the new gTLDs, number of parked, and DNSSEC-signed domains play a statistically significant but weak role in explaining the differences in abuse counts among different new gTLDs. Low domain registration prices, unrestrictive registration practices, a variety of other registration options such as WHOIS privacy, registration in bulk and finally the increased availability of domain names decrease barriers to abuse and seem to make some new gTLDs very attractive for miscreants.

Taken together, our findings indicate that the existing safeguards do not prevent domain name abuse and therefore we further develop cases for modifying the existing safeguards and propose new ones, which we extensively discussed with the ICANN community.

Acknowledgements

This study was commissioned by the Competition, Consumer Trust, and Consumer Choice Review Team with the support of ICANN. We would like to thank ICANN, DomainTools, Whois XML API, Spamhaus, SURBL, StopBadware, CleanMX, Secure Domain Foundation, Anti-Phishing Working Group for providing access to their data. Authors also thank Roland van Rijswijk for his help in obtaining additional domain data and anonymous reviewers for their constructive and valuable comments.

Appendix

5.8 Overlap Among Blacklists

To determine the overlap among our blacklsits, we present their pairwise intersections as a matrix in Figure 5.21, after extracting unique domain names from each data feed. Note that darker shades of grey represent larger overlaps among compared feeds. For example, the overlap between Spamhaus and SURBL ws indicates that they have 2,257,450 domain names in common within the observation period. This overlap constitutes 37% of the Spamhaus feed. In comparison, 2,257,450 domain names represent 64% of the SURBL ws feed. This is to be expected as both blacklists contain the same type of abuse, i.e. spam. The rightmost column indicates the absolute number and the

percentage of samples that each blacklist has in common with all other feeds combined. For instance, the overlap between Spamhaus and all other blacklists is equal to 3,054,837 and indicates that as many as 51% of all domains blacklisted by Spamhaus are blacklisted by at least one other organization. Combined, these blacklists provide a comprehensive overview of domain name abuse for various criminal purposes.

5.9 Method to Distinguish Between Compromised and Maliciously Registered Domains

We flag a domain name as malicious if it is blacklisted within 3 months after its registration. Aaron and Rasmussen have recently examined the delay between the time when phishing domains were initially registered and when they were ultimately used in attacks [197]. Their analysis indicates that miscreants tend to age the malicious domains they register to ensure a higher reputation score from security organizations. They concluded that the great majority of the domains used for phishing were maliciously registered within three months before they were used in an attack. To estimate the time between original registration and blacklisting, we analyze domain `WHOIS` information and extract the domain *creation date*. According to the Registrar Accreditation Agreement (RAA) [218], the creation date of the domain registration cannot be changed as long as the domain does not expire.

Furthermore, Aaron and Rasmussen identified 783 unique organizations used as phishing targets in 2015 and 679 in 2016, among which the most popular ones were PayPal, Yahoo!, Apple, and Taobao [197]. We used this information to create a list of keywords that the attackers may incorporate in maliciously registered domain names. As the great majority of phishing attacks target the most popular organizations, we extracted 300 keywords of the most popular domains according to their Alexa ranking and we labelled each blacklisted domain as maliciously registered if it contained an extracted string or its misspelled version. For example, `0paypalpayment.com` would be labelled as malicious as it contains the string "paypal". To test if the domain contains a misspelled keyword, we first remove all digits from a domain name and split the resulting string into words with the "–" character. We compute the Levenshtein edit distances between the predefined keywords and a set of words derived from a domain name. If any Levenshtein edit distance is smaller than 2, we label the domain as maliciously

registered.

Note that from the categorization process we exclude a list of 11,075 domains of legitimate services that tend to be misused by miscreants. These represent a separate, third group of domains that are neither maliciously registered nor hacked (i.e. third party domains). For example, `bit.ly` – a domain used by a legitimate URL shortener service – could be used by an attacker to create a malicious URL (e.g. `bit.ly/dcsahy`) that may further be used to redirect a legitimate user to a phishing website. In fact, previous research shows that miscreants extensively abuse a variety of services with good reputations, affecting not only the reputation of those services, but of entire TLDs [201]. The list is composed of the 10,000 most popular domains according to their Alexa ranking and our own, manually maintained lists of domains of legitimate services (332 domains of URL shorteners and 840 domains of free hosting providers).

5.10 Blacklisted Spam Domains in Legacy gTLD and New gTLDs Based on the SURBL Feeds.

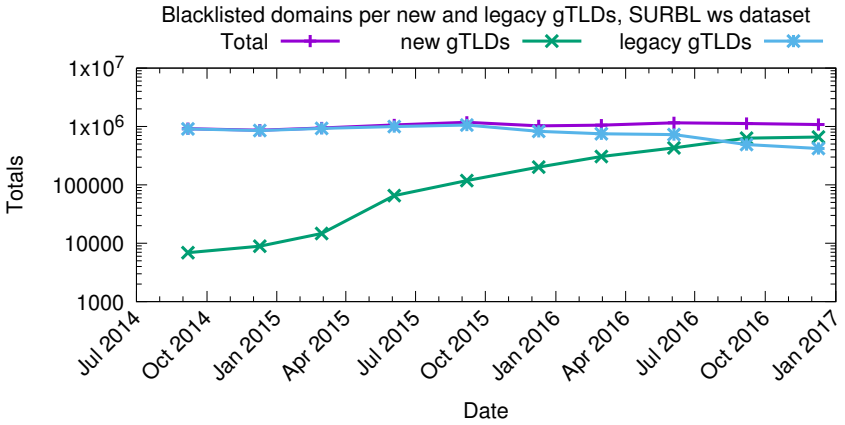


Figure 5.19: Time series of counts of blacklisted Spam domains in **legacy** gTLD, **new** gTLDs, and **all** gTLDs (Total) based on the **SURBL ws** feed.

5.11 Method to Identify WHOIS Privacy and Proxy Services

To identify the most commonly used WHOIS Privacy and Proxy services we used the following methodology: *i)* Using the `WHOIS` data, we aggregated all distinct domains

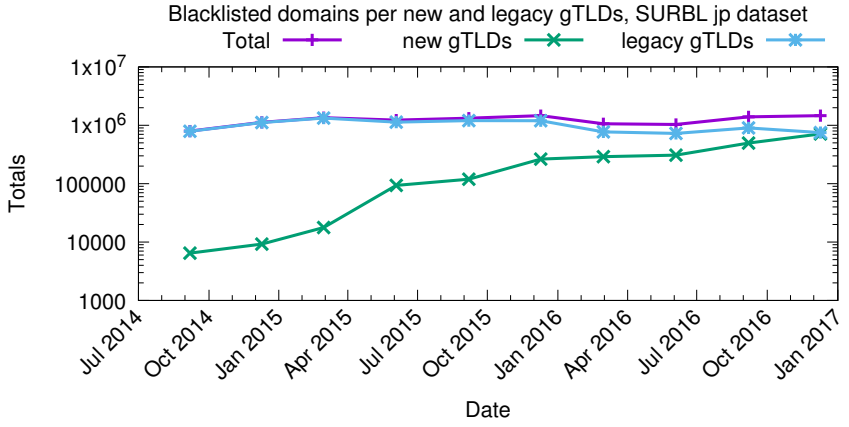


Figure 5.20: Time series of counts of blacklisted Spam domains in **legacy** gTLD, **new** gTLDs, and **all** gTLDs (Total) based on the **SURBL jp** feed.

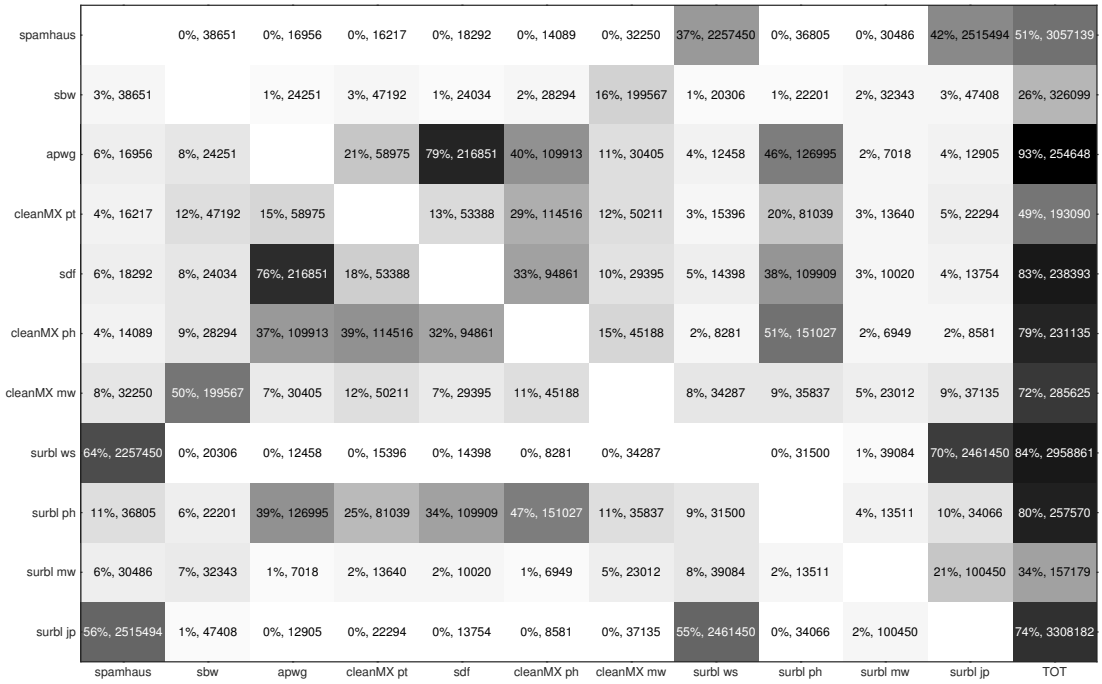


Figure 5.21: Pairwise overlap of feeds with unique domains (2014-2016)

by "registrant name" and "registrant organization" attributes and created a list with the top 5,000 registrants. *ii)* A keyword search on the top 5,000 "registrant name" and "registrant organization" attributes, trying to match any registrant with keywords such as: "privacy", "proxy", "protect", "private", "whois" etc. *iii)* A manual inspection of the suspect "registrant name" and "registrant organization" attributes to decide if the registrant is a Privacy and Proxy service (when this was not immediately clear from the name itself we used an Internet search to find additional information). Using the above described method we identified 570 "registrant name" and "registrant organizations"

Table 5.2: Top 10 new gTLDs with the highest relative concentration of blacklisted domains for StopBadware SDP, APWG, Spamhaus, SDF, and SURBL datasets (fourth quarter of 2016). $Rate = 10,000 * \#blacklisted\ domains/\#all\ domains$.

StopBadware			APWG			Spamhaus			SDF		
TLD	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**
TOYS	32	78	LIMITED	31	66	SCIENCE	117,782	5,154	SUPPORT	510	294
TRADE	221	15	SUPPORT	43	24	STREAM	18,543	4,756	TECH	4,409	158
TAFTAR	1	11	CENTER	72	22	STUDY	1,118	3,343	ONLINE	4,179	83
WANG	1,086	11	CREDITCARD	1	13	DOWNLOAD	16,399	2,016	LIMITED	15	32
JUEGOS	1	9	SERVICES	24	10	CLICK	20,713	1,814	REVIEW	161	24
TOP	3,830	8	ONLINE	417	8	TOP	736,339	1,705	CLAIMS	3	19
MOE	5	8	MOE	5	8	GDN	45,547	1,602	PRESS	91	19
CAB	3	7	HOST	32	7	TRADE	23,581	1,521	FURNITURE	4	18
PICS	10	7	LEASE	1	6	REVIEW	9415	1,318	WEBSITE	298	15
TATTOO	2	7	REPORT	3	6	ACCOUNTANT	6,722	1,279	CREDITCARD	1	13
SURBL ph			**SURBL mw**			**SURBL ws**			**SURBL jp**		
TLD	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**	**TLD**	**# Domains**	**Rate**
LIMITED	51	109	FOOTBALL	7	16	RACING	51,443	3,812	SCIENCE	152,719	6,683
SUPPORT	82	46	TOP	5,066	11	DOWNLOAD	21,515	2,645	CLICK	27,871	2,441
CENTER	93	29	RIP	1	5	ACCOUNTANT	10,543	2,007	GDN	50,940	1,792
SERVICES	61	25	BID	200	3	REVIEW	12,615	1,766	STREAM	6,033	1,547
CRICKET	57	22	DENTIST	1	3	GDN	49,427	1,739	LINK	39,764	1,238
ONLINE	903	16	LGBT	1	3	FAITH	5,540	1,301	REVIEW	8,705	1,219
WEBSITE	318	14	ACCOUNTANT	11	2	TRADE	19,330	1,247	CRICKET	2,468	993
REPORT	7	14	CAB	1	2	CLICK	13,270	1,162	TRADE	14,535	937
HOST	65	13	SUPPORT	5	2	STREAM	4,406	1,130	FAITH	3,130	735
CREDITCARD	1	13	POKER	1	2	DATE	1,3851	999	TOP	285,488	661

Table 5.3: Negative Binomial GLM for count of abused domains per new gTLD

	Dependent variable:						
	apwg	sbw	cmx ph	cmx pt	cmx mw	surbl ph	surbl mw
	(1)	(2)	(3)	(4)	(5)	(6)	(7)
New gTLD size	0.00002***	0.00001***	0.00002***	0.00003***	0.00001***	0.00002***	0.00002***
	(0.00001)	(0.00000)	(0.00001)	(0.00001)	(0.00000)	(0.00001)	(0.00001)
Parked	0.0003***	0.0001***	0.0002***	0.00003	0.0001***	0.0002***	0.00001
	(0.00004)	(0.00003)	(0.00003)	(0.00004)	(0.00003)	(0.00004)	(0.00004)
DNSSEC	0.00001***	0.00002***	0.00002***	0.00002***	0.00001***	0.00002***	0.00002***
	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)
No DNS	−0.00004***	−0.00003***	−0.00005***	−0.00005***	−0.00002***	−0.00004***	−0.00004***
	(0.00001)	(0.00001)	(0.00001)	(0.00001)	(0.00000)	(0.00001)	(0.00001)
HTTP Error	−0.00002	−0.00004***	−0.0001***	−0.00003*	−0.00004***	−0.0001***	−0.0001***
	(0.00002)	(0.00001)	(0.00001)	(0.00002)	(0.00001)	(0.00002)	(0.00002)
Type	−0.540**	−0.150	−0.400**	−0.120	−0.190	−0.760***	−0.170
	(0.220)	(0.120)	(0.180)	(0.170)	(0.160)	(0.190)	(0.220)
Constant	−0.630**	−0.390**	−0.960***	−1.200***	−1.600***	0.330	−2.200***
	(0.280)	(0.170)	(0.230)	(0.230)	(0.220)	(0.230)	(0.290)
Observations	521	521	521	521	521	521	521
Log Likelihood	−566.000	−792.000	−508.000	−546.000	−392.000	−786.000	−284.000
θ	0.140***	0.330***	0.240***	0.200***	0.470***	0.190***	0.240***
	(0.017)	(0.035)	(0.034)	(0.024)	(0.087)	(0.019)	(0.051)
AIC	1,149.000	1,600.000	1,031.000	1,109.000	800.000	1,588.000	583.000

Note: *p<0.1; **p<0.05; ***p<0.01
Standard errors in brackets

attribute combinations used by WHOIS Privacy and Proxy services.

Each blacklist abuse incident contains metadata such as the date when the domain was added to the blacklist. We used this date to identify the correct historical `WHOIS` record for an abused domain. By comparing the "registrant name" and "registrant

organization" attributes from the domain `WHOIS` record to the list of known WHOIS Privacy and Proxy services, we are able to correctly identify abusive domains that were using a WHOIS Privacy and Proxy service at the time the domain was added to a blacklist.

Table 5.4: SURBL top10 percentage between blacklisted new and legacy gTLD domains (#Incidents) and total number of registrar gTLD domains (#Domains).

#	new gTLD registrar	#Domains	#Incidents	Percent	Legacy gTLD registrar	#Domains	#Incidents	Percent
1	Nanjing Imperiosus Technology	38,025	35,502	93.36	HOAPDI INC.	141	126	89.36
2	Intracom Middle East FZE	20,640	11,255	54.53	asia registry r2-asia (700000)	1,379	598	43.36
3	Dot Holding Inc.	153	76	49.67	Nanjing Imperiosus Technology	35,309	10,834	30.68
4	Alpnames Limited	3,028,011	751,748	24.83	Paknic (Private) Limited	10,525	3,083	29.29
5	Todaynic.com, Inc.	329,399	69,404	21.07	OwnRegistrar, Inc.	22,188	5,238	23.61
6	Web Werks India Pvt. Ltd	785	146	18.6	Eranet International Limited	6,109	1,339	21.92
7	GMO Internet, Inc. d/b/a Onamae	1,734,775	295,641	17.04	BR domain Inc. dba namegear.co	847	158	18.65
8	TLD Registrar Solutions Ltd.	163,988	24,700	15.06	Netlynx Inc.	17,612	3,030	17.2
9	Xiamen Nawang Technology, Ltd	282,925	42,089	14.88	AFRIREGISTER S.A.	1,551	266	17.15
10	Instra Corporation Pty Ltd.	77,642	6,200	7.99	GMO Internet, Inc. d/b/a Onamae	7,306,312	1,177,886	16.12

Chapter 6

COMAR: Classification of Compromised versus Maliciously Registered Domains

Coauthors: Sourena Maroofi, Maciej Korczyński, Cristian Hesselman, Benoit Ampeau, and Andrzej Duda

6.1 Introduction

Domain names play an important role in almost all types of cybercriminal activities. Miscreants tent to use domains in various attack scenarios such as phishing (e.g., to collect sensitive information) or spam campaigns, or as part of command and control (C&C) services with algorithmically generated domain names (AGDs). In all these cases, the involved domains are either solely registered for malicious purposes (which we refer to as *malicious* for simplicity) or registered for legitimate reasons but have been compromised at some time to serve malicious content (we refer to these domains as *compromised*).

One common way to fight malicious activities is to build domain blacklists so a security system can check whether a domain exists on the blacklist and decide on how to treat the incoming traffic related to that domain [38]. However, this method is effective when the blacklist only contains the malicious domains because if it includes the compromised ones, the legitimate services associated with the domains may be interrupted and cause financial loss.

At the time of registration, each domain has two possible states: either it is registered for a malicious purpose or a legitimate one. Then, when the domain is active, there are three possible states: (1) *Benign*: the incoming traffic from the domain is benign and can be passed to users safely, (2) *Malicious*: the traffic related to the domain should be considered as malicious and treated differently (e.g., blocked), and (3) *Compromised*: an attacker leverages an arbitrary vulnerability to upload malicious content, e.g., a phishing page. In this way, while the legitimate website is likely to continue serving benign content to its customers, the attacker benefits from the good reputation of the website to conduct her phishing attack. Therefore, the traffic related to the domain can be either malicious or benign.

The existing domain name reputation systems only consider the first two states. They can detect malicious domains either at registration (e.g., PREDATOR [219]) or after they exhibit malicious behavior (e.g., EXPOSURE [39]). However, none of them can detect compromised domains due to two major problems: (1) there is no such state as *compromised* at the registration time, and (2) compromised domains may exhibit the same behavior as malicious domains while they are benign and abused to serve malicious content. In this regard, domain reputation systems may detect a compromised domain as malicious and blacklist it [38]. While this method successfully prohibits malicious traffic, it also blocks the traffic to the legitimate part of the compromised domain. If such a system identifies a compromised domain as benign, it helps attackers achieve their goals. Therefore, in both cases, the decision on the state of the domain may cause collateral damage. For this reason, a complementary system is required to work along with domain reputation systems to differentiate the compromised domains from the malicious ones.

Apart from creating effective domain blacklists, distinguishing compromised from malicious domains is also important for intermediaries involved in the domain name registration and deployment process. When confronted with a malicious URL, it is critical to assess the registrant's intention for registering the underlying domain since the mitigation action could be different if the registration is for malicious purposes or not. Regarding Top-Level Domain (TLD) registries, one appropriate action for malicious domains is domain delisting, i.e., removing the name from the zone file and changing its status to *hold* to completely deactivate it [220]. Another appropriate action is to block access to the domain (*domain blocking*) or redirect the traffic of the domain to

another server under the control of authorized entities (also known as domain *sinkholing*), which can be done by registrars. The latter is a popular and widely used technique to identify the victims infected by malware and to reduce its spread [221].

Taking appropriate action against blacklisted domains is also important for hosting providers since hosting malicious content can adversely affect their reputation [222–225]. Canali et al. studied the reaction of hosting providers when confronted with compromised websites [226]. They showed that in more than 50% of the cases, the reaction of the hosting providers was to suspend (or terminate) users' accounts. For large providers, which may receive hundreds of abuse notifications every day, it is not feasible to manually investigate each case. Therefore, there is a need for a system that can help hosting providers identify the compromised domains and differentiate them from the malicious ones for taking appropriate actions.

Distinguishing between malicious and compromised domains may also lead to revealing the profit-maximizing behavior of attackers. For example, there has been anecdotal evidence indicating that miscreants choose to abuse registrars that offer low domain registration prices [11,227–229]. However, no study has systematically proved this conjecture mainly because the existing URL blacklists conflate compromised and malicious domains. One attacker may indeed prefer lower registration prices but, others may choose to abuse a registrar that offers specific payment methods or a free API allowing for domain registration in bulk. On the other hand, registrars might offer cheap domains but, to prevent domain abuse, perform additional checks to confirm the identity of registrants.

In this chapter, we propose **COMAR** (Classification of COmpromised versus MALiciously Registered Domains), a system capable of differentiating compromised (and misused) domains from the malicious ones to 1) create more effective domain blacklists, 2) help registries, registrars, and hosting providers to take appropriate mitigation actions depending on the abuse type, and 3) gain better insights into the attackers' behavior for choosing candidate domains to hack and intermediaries to abuse.

We thoroughly study the domain life cycle to understand the intentions of both miscreants and benign users and determine the relationship between each step and its associated features. We use OpenPhish [230], PhishTank [231], Anti-Phishing Working Group (APWG) [232], and URLhaus [233] as our initial URL blacklist resources, but the system is not only limited to phishing or malware feeds. Our results illustrate

that COMAR achieves high classification accuracy by leveraging only *publicly available* data without relying on any privileged resources like historical WHOIS or passive DNS traffic. We also show how it is possible to compensate for the lack of domain creation time if there is no access to WHOIS information.

In summary, we make the following contributions:

- We develop a system to classify domains exhibiting malicious behavior as either compromised or maliciously registered by *only* using publicly available and readily accessible resources, and achieve 97% accuracy with 2.5% of false positives.
- We leverage 38 features to identify the state of a domain, 14 of which are new and have not been used in previous work.
- We introduce a new method to estimate the domain creation time in cases there is no access to WHOIS information, which outperforms standard statistical methods in filling missing values.
- We show that content-based features are the most important ones in representing the domain status.

6.2 Domain Life Cycle

To understand better the intentions of both malicious actors and benign users for registering and maintaining a domain name, we need to thoroughly inspect the domain life cycle and determine the relationship between each step and its associated features. In this way, we can capture the characteristics of the benign but compromised and malicious domain registration. We divide the domain life cycle into four phases as follows:

L1. Choosing the domain name. In this phase, both miscreants and benign users try to register an appropriate domain name based on their needs. Benign users tend to choose easier to remember, meaningful domain names related to the service provided by the domain. Malicious actors with the purpose of a phishing attack in mind, may try to choose a deceptive name to lure benign users and steal their personal information (e.g., facebook-account.support). In the case of malware C&C panels, miscreants may choose the names that can be generated by the malware family as part of a domain generation algorithm (DGA). These domain names are likely to be long and meaningless

(to increase the chance of availability). We expect spammers to use domains that contain keywords of the targeted service to effectively persuade users to click on the link to increase the click rates and search engine ranks (e.g., earn-bitcoin.biz). The knowledge we gain from this phase can help us to build appropriate lexical features related to the characteristics of the domain name.

L2. Registration of the domain name. A user (registrant) registers a domain either through a registrar or a reseller by paying the registration fee. The registration period can be between one to ten years depending on the registrars and registries (although shorter registration periods also exist [234]). In this phase, malicious actors tend to choose less expensive (or free) TLDs to maximize their profit [11, 227–229]. The name of the registrar, domain creation, and expiration dates are stored by registrars and registries as part of WHOIS information. The registrant's information, i.e., the registrant name, address, phone number, are often obscured and not publicly available due to the European General Data Protection Regulation (GDPR) [178]. The COMAR system uses the public part of the WHOIS data as well as TLD-related registration features such as retail domain pricing to discriminate between malicious and compromised domains.

L3. DNS record configuration. After the domain name registration phase, DNS records should be set up to allow the discovery of the services associated with the domain. Each resource record provides information about the service behind the domain name. For example, the DNS 'A' record gives the IP address of the server providing the content for that domain (sometimes, the 'A' record points to a reverse proxy server, responsible for fetching the content from the backend server and delivering it to end-users). The 'MX' records point to mail servers whereas 'DMARC' and SPF 'TXT' records are for giving the email domain owners the ability to protect their domain from unauthorized use. Passive DNS datasets (e.g., Farsight Security [235]) come from monitoring DNS responses and extracting DNS information. For legitimate domains, we expect more stability and *availability* of DNS records while for malicious domains, we expect frequent changes or unavailability of some records (e.g., 'TXT', 'MX', or 'DMARC'). COMAR uses the monitoring approach of passive DNS to construct a feature set, but it *does not* rely on passive DNS datasets since they are not always publicly available. We also use active DNS features by querying blacklisted domains.

L4. Service deployment process and its activity period. This step consists of all the ac-

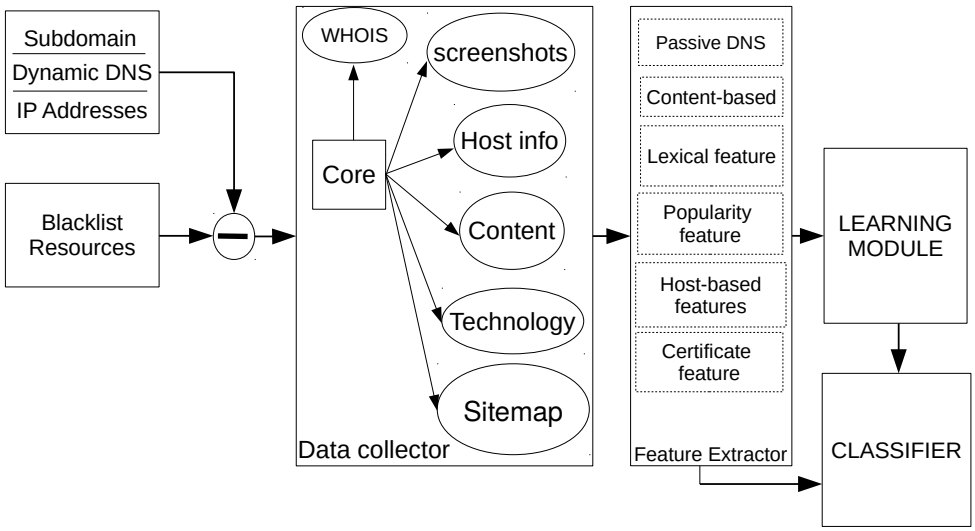


Figure 6.1: COMAR system structure.

tivities to set up the necessary infrastructure for the (legitimate or malicious) service offered by the domain. The activities may include setting up a web server, deploying the application to manage the web content, or ordering a Transport Layer Security (TLS) [146] certificate for the domain name to build trust of the service visitors. We expect that legitimate domain owners put the effort in creating content to increase user interest and therefore, the website popularity, i.e., the amount of web traffic the site receives. Miscreants may or may not take the effort of setting up real websites depending on the type of abuse. We also expect to observe more (vulnerable) libraries and technologies to build a legitimate website, which is not required for the correct operation of malicious domains. In this phase, COMAR collects data mainly through a crawl of blacklisted domains and extracts host-based, popularity, and its most important content-based features.

6.3 Methodology

Our system comprises three main modules: 1) data collector, 2) feature extractor, and 3) learning and classification modules. Figure 6.1 presents its structure.

The data collector module gathers data related to the domains derived from URL blacklists. The feature extractor module derives features from the collected data. It can be further used to support efforts of manual labeling domains as maliciously registered or compromised. The learning module takes the labeled data on an input to build a

classifier using an appropriate supervised learning technique. Finally, the classification module uses the extracted features and the generated model to classify unlabelled domains derived from URL blacklists in real-time.

6.3.1 Data Collector Module

We use OpenPhish, PhishTank, APWG, and URLhaus as our initial blacklist resources, but the system is not limited to these URL feeds and can use other types of blacklists on input.

The system downloads URL blacklists every 5 minutes to one hour (depending on the blacklist) to get the newly blacklisted URLs. Some URLs are already not operational by the time they are downloaded (domains are taken down or websites are suspended). Some URLs do not contain domain names and use IPv4 addresses instead, whereas some of them use free subdomain services or dynamic DNS services. We use the private part of the public suffix list [236] to exclude dynamic DNS and free subdomain services from further analysis. For each remaining newly appeared URL in the blacklist, we collect the following information:

Technology information. We define technology information as frameworks and libraries used to build websites (both client-side libraries like JQuery and WordPress, and server-side technologies like PHP or ASP programming languages). To extract such data, we use the Wappalyzer [237] signature list. For each signature in the list, we search in (a) the URL, (b) HTTP headers, and (c) page content to extract all the libraries and tools used to build the website.

Page content. For each domain name, we download the corresponding homepage for further analysis and extracting features. To catch the real content of the domains, which are behind the reverse proxy service (e.g., Cloudflare) with the anti-DDoS feature enabled, we emulate the behavior of a real browser to solve the JavaScript anti-DDoS challenge [238] by using a headless version of the Firefox and Selenium browsers.

Sitemap structure. We further extract all the hyperlinks on the homepage and generate the tree structure of the domain name. For professionally designed websites, the sitemap is often stored in the root directory of the website. However, most of the compromised websites are not well designed, whereas malicious domains rarely have a sitemap file (even if they do, they are not trustworthy). Therefore, we develop our crawler to generate a sitemap for domains. For example, Figure 6.2 shows the website

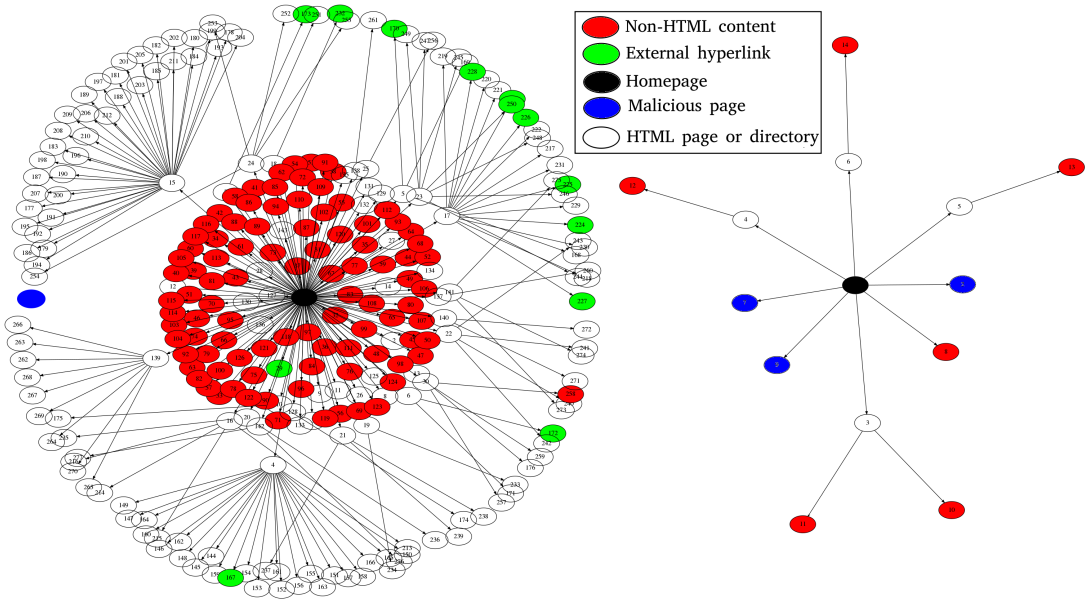


Figure 6.2: Website structure of a compromised (left) and a malicious domain (right) with the depth level of 3.

Table 6.1: Features and their characteristics. Feature types are binary (B) or continuous (C). The availability column shows the availability of features as highly available (high), medium, or low. The source column shows the features defined by us (*new*) or appeared in previous work.

Feature#	Type	Availability	Source
(1) - (3)	B	High	[197], [239], [11]
(4)	B	High	new
(5)	C	High	[240]
(6)	C	High	[39]
(7)	C	High	[241]
(8)	B	High	[242]
(9)	B	High	new
(10)	C	Medium	[223]
(11)	B	High	new

Feature#	Type	Availability	Source
(12),(13)	C	High	new
(14)-(16)	B	Medium	new
(17)	B	Medium	[241]
(18)	C	Medium	new
(19)	C	Medium	new
(20)-(24)	C	High	[243]
(25)	B	High	new
(26)	B	Medium	[219]
(27)	C	Medium	[197]

Feature#	Type	Availability	Source
(28)	C	Medium	[244]
(29)-(31)	B	High	[64]
(32)	B	High	[64]
(33)	C	High	[245]
(34)	B	High	new
(35)	C	Low	new
(36)	B	Low	new
(37)	B	High	[241]
(38)	B	Medium	[246]

structure of two sample domains with 3 levels of depth for compromised (left) and malicious (right) domains. The black node in the center of the image is the homepage of the domain name. The green nodes are the links to external domain names, the red nodes

are the links to non-HTML data types such as PDF or ZIP files, whereas the white nodes are either HTML pages (leaf) or directories (non-leaf). The blue node shows the malicious page. Having this graph, we can extract various information about the website. For example, the number of internal links to pages with different HTML content is higher in the compromised domain compared to the maliciously registered domain because compromised domains have legitimate parts for their users. More importantly, most of the time, there is no connection between the phishing page and the homepage in compromised domains since malicious actors do not tend to change the homepage of the compromised domain. Malicious domains have often a connection between the homepage (if there is one) and the malicious page.

DNS resource records. For each domain, we actively collect the 'A', 'AAAA', 'NS', 'TXT', 'SOA', 'DMARC', and 'MX' resource records. Then, using the Maxmind database [247], we convert the 'A' record to the country code and the autonomous system number (ASN) for further use. We also extract the sender policy framework (SPF) [248] rule from the 'TXT' record if available.

Host information. The host information module is responsible for collecting all the host information related to the input domain (and a possible subdomain) at the time of blacklisting. This information includes the TLS certificates of the domain, the HTTP headers of the web server, the AS number, and its related organization name.

WHOIS data. We collect and parse the WHOIS data, however, we only use the domain creation date in our features. Since this field is not available as part of the WHOIS data for all TLDs, we estimate the missing value based other features (see Section 6.3.4 for more details).

Screenshots. The lifespan of the blacklisted URLs is short [249]. Therefore, for each domain, we save the screenshots of the homepage as well as the malicious URL (and a subdomain if there exists any) for further manual analysis and labeling of domains in case the website has been taken down by registrars, hosting providers, or miscreants.

6.3.2 Features

The feature extractor module extracts features from the collected data. It operates along with the data collector in a real-time manner to convert plain data into features. In total, we extract 38 features divided into seven main categories (feature set F_1 through F_7) as presented below:

1. Lexical features (F_1)
2. Ranking system and popularity features (F_2)
3. Passive DNS features (F_3)
4. Content-based features (F_4)
5. WHOIS and TLD-based features (F_5)
6. TLS certificate features (F_6)
7. Active DNS features (F_7)

Table 6.1 shows the characteristics of each feature along with their availability, types (B: binary or C: continuous), and if they appeared in previous work or are defined by us.

Table 6.2: Lexical features used in maliciously registered domain names.

Domain name	Attack type	Lexical features
paypala.com	Phishing	(1) (2) (3)
suportaccount-services.com	Phishing	(4) (5)
3lf4vlxegj1luy6kbs.com	AGD (Rovnix)	(6)
erdoypf-inr.net	AGD (Redyms)	(5)
applid.appsgr.girtrusgirs.com	Phishing	(7)

6.3.2.1 Lexical features

They are the features extracted from the registered domain (e.g., *example.com*), the subdomain (e.g., *sub.example.com*) as well as the path part of the URL.

Famous brand name in the domain name ($\mathbf{f_1}$). We have identified 231 brand names mostly targeted by attackers in phishing attacks (e.g., PayPal, Amazon, Yahoo, or Gmail). We have created a list of keywords by manually inspecting phishing pages and the corresponding domain names. If the domain name consists of one of these words, it is an indication of maliciousness.

Misspelled target brand name in the domain name ($\mathbf{f_2}$). We use dnstwister [250] to generate possible similar domain names for each of 231 brand names and compare them with the domain name to check the existence of these words. We also consider internationalized domain names and convert the unicode characters to their look alike ASCII equivalent to cover homograph attacks.

Levenshtein distance of the domain name and targets ($\mathbf{f_3}$). We calculate the Levenshtein distance (LD) between the domain name and every 231 targets on our list. We choose $LD = 1$ as the threshold as proposed by Korczyński et al. [11].

Special words but not brand names in the domain name ($\mathbf{f_4}$). Some specific words (e.g., verification, account, support) are not brand names but, based on our word frequency analysis, miscreants tend to use them as part of the domain name to lure victims to enter their credentials. We split the domain name into a word list using the hyphen character. For each word in the domain name, we look for a complete or partial match of that word and our predefined list of 28 keywords. For example, for the domain name 'supportacc-paypal.com', we have one brand name match (i.e., 'paypal') and one special word match (i.e., 'support').

Number of hyphens in the domain name ($\mathbf{f_5}$). The only special character that can be used in a domain name is hyphen ('-'). Both phishing [240] and algorithmically generated domain names (e.g., Redyms malware [251]) tend to use hyphens as part of the domain name.

Digit ratio ($\mathbf{f_6}$). AGDs tend to have more digits than legitimate domain names [39]. This feature is more suitable for domains generated by malware families.

Level of subdomains ($\mathbf{f_7}$). As miscreants control the DNS records of the malicious domains[1], they can create as many subdomains as necessary for a successful attack [241].

Presence of a brand name in the path part of URL ($\mathbf{f_8}$). For compromised domains for which attackers generally do not have access to the domain zone file to create new subdomains, the only way for the malicious actors to use the target brand name is to include it as a part of URL.

Presence of the dot character in the path part of URL ($\mathbf{f_9}$). By manual analysis of blacklisted URLs, we have observed that some malicious actors tend to use the dot character ('.') before file or directory names, for example: *https://masseffect.co.za/.lilman/login.php?cmd=submit*, which may allow the attacker to deceive an unskilled administrator who may not notice the hidden malicious content on the compromised system.

For features $\mathbf{f_1}$-$\mathbf{f_6}$, we only consider the domain name part of the blacklisted URL. Feature $\mathbf{f_7}$ considers the subdomain, whereas $\mathbf{f_8}$ and $\mathbf{f_9}$ are only related to the path part of the URL. Table 6.2 shows the use of selected features in various types of maliciously

[1] In some types of attacks like zone poisoning [252] or domain shadowing [253], it is still possible for miscreants to change almost any DNS record of a benign domain and generate arbitrary subdomains.

registered domains.

6.3.2.2 Content-based features

The ultimate goal of domains is to identify a website or a web service that serve content to their customers in various forms. While it is not trivial to examine the content validity, yet it is feasible to extract informative content-based features.

Content length ($\mathbf{f_{10}}$). Malicious domains tend to have less content [223]. For this feature, we only consider the content length of the homepage for each domain part of the blacklisted URL. If there is no index page for that domain (i.e., default directory listing page of the web server), or the web server returns any HTTP code other than the success code (e.g., 404 not found or 403 not authorized), we consider the length to be zero.

Number of used technologies ($\mathbf{f_{11}}$). Using different frameworks and libraries in building a website needs time and effort. The more different technologies used in creating a website, the more time spent on the development. Therefore, we consider the number of used technologies as an indication of the domain being benign. We crawl websites to fingerprint software using unique words and patterns found in the source code. We derive the fingerprints and signatures used in Wappalyzer [254].

Vulnerable technology ($\mathbf{f_{12}}$). It is a binary feature that indicates whether the website uses a technology with at least one known vulnerability. For example, 271 known vulnerabilities have been found in the WordPress content management system (CMS), including themes and plugins that enable the attackers to upload an arbitrary file to the server [255]. Other familiar technologies with known vulnerabilities are *Joomla* or *Drupal* CMSes, and *Magenta*, *PrestaShop*, or *DotNetNuke* frameworks. The intuition is that if a website uses one of these CMSes, frameworks, modules, or libraries, then there is more chance to get compromised. To obtain the list of technologies with at least one reported vulnerability, we use the exploit [256] and vulnerability databases [257].

Number of internal working hyperlinks ($\mathbf{f_{13}}$). The website with some content is not always benign since miscreants may create fake content on the website so that it looks legitimate. The easiest way for malicious actors is to clone the content of a legitimate website. For each internal hyperlink in the homepage, i.e., a page belonging to the same domain, we fetch the content (only HTML content not files) and calculate the fuzzy hash as proposed by Kornblum [258] to make sure all the pages are not the

same and then count the number of unique hashes as the number of working internal hyperlinks.

Content-related domain name ($\mathbf{f_{14}}$). This feature defines the relationship between the content of the homepage and the domain name itself. We extract the meaningful words (based on a dictionary) of the domain name and search for those words in the visible text of the homepage. It is a binary feature with the values 1 (at least one match between a word from the domain name and a related word in the textual content) or 0 otherwise. Another approach would be to use the 'Google trends' service but it is paid and difficult to use on a larger scale.

Presence of the index page ($\mathbf{f_{15}}$). It is common for attackers to upload their files to the web server and just use them without appropriate configuration. In this case, if they forget to upload an appropriate index page (e.g., index.html, index.php, or index.asp depending on the server and server-side programming language), the default behavior of the most web servers (e.g., NGINX or Apache) is to list the directory content. One possibility is that the attacker could remove the index page from the compromised domain but it leads to immediate reaction of the webmaster.

Presence of the default index page ($\mathbf{f_{16}}$). The index pages (homepages) of some domains are the default sites deployed by the registrars, hosting providers, or resellers after the domain name registration process is complete. Resellers often offer free software installation plans like WordPress or Joomla CMSes along with hosting plans. The whole process of installing a CMS on the server takes a few minutes. Sometimes, attackers leverage these free plans to make the domain looks more legitimate. For each domain name, we compare the content of the index page with our pre-defined list of default pages from familiar CMSes and default control panels to check whether the home page is a default page or not.

Using page redirection ($\mathbf{f_{17}}$). Homepage redirection and web cloaking [259] are two common methods among attackers to conceal their malicious intention by displaying benign contents to web crawlers and bots. Regarding homepage redirection, when users try to visit the homepage of the malicious domain, they will be redirected to a benign website. In case of phishing, the redirection is mostly to the real website of the phishing target (e.g., the real Bank of America website). In case of malware domains, it can be a random website like google.com. To distinguish between page redirection and web cloaking, we crawl each malicious URL with the Selenium browser and with the Python

requests library. We set this feature to true if the destination URL of the homepage requested using both the browser emulation and the *requests* library shows the same domain name but it is different from the domain name of the blacklisted URL.

Number of external hyperlinks ($\mathbf{f_{18}}$). This feature works the same way as *internal working hyperlinks* but counts the number of hyperlinks that refer to external domains. Sometimes miscreants, especially in phishing attacks, tend to clone the target website to perform a more successful attack. In these cases, the cloned website often has hyperlinks to the real target. Using this feature, we can capture such fake content.

6.3.2.3 Passive DNS features

Passive DNS has become an industry-standard tool for more than a decade. It can give us insights into how the behavior of a domain changes over time (e.g., changing the IP address) and how popular the domain name was in the past. Although the features based on passive DNS data proved to be significant, they can be compensated by other features without lowering accuracy. Therefore, the absence of this feature set does not affect the classification results.

First passive DNS query before the blacklist time ($\mathbf{f_{19}}$). The number of days between the first occurrence of a passive DNS query (for 'A' or 'NS' records) and the blacklist time. This feature provides the estimation of the age of the domain in terms of usage and not only with respect to the registration.

Passive DNS queries ($\mathbf{f_{20}}$-$\mathbf{f_{24}}$). The number of queries for each resource record before appearing in the blacklist resources (i.e., 'A', 'AAAA', 'NS', 'MX', 'TXT' records). For example, the higher the number of observed 'MX' queries, the higher the chance that the domain has an active mail service.

6.3.2.4 Active DNS features

We extract the following features from DNS data queried shortly after the blacklisting time.

Presence of the sender policy framework (SPF) ($\mathbf{f_{25}}$). 'TXT' records are used (among others) for setting SPF rules [248], domain message authentication reporting and conformance (DMARC) rules [260], and in some cases for domain ownership verification by third-party services (like Google App verification). The presence of SPF for a specific domain can be considered as an indication of legitimacy. For example, a

domain owner for whom protection against email spoofing is important would set an appropriate SPF rule in the 'TXT' record [261]. Nevertheless, the malicious actors may also set up SPF rules to increase domain reputation.

Self-resolving name server ($\mathbf{f_{26}}$). Miscreants may use self-resolving name servers i.e., name servers responsible for resolving their own domain names (e.g., *ns1.domain.com* for resolving *domain.com*) [219], whereas legitimate users tend to use the default DNS resolvers of their DNS service providers.

6.3.2.5 WHOIS features

Due to the introduction of GDPR and our requirement that the proposed method should only depend on publicly available data sources, we only derive the domain creation date from WHOIS and propose the following feature:

Domain age ($\mathbf{f_{27}}$). The older the domain name, the higher the chance to be legitimate. However, according to the 2016 APWG report [197], some miscreants age registered domains waiting weeks or sometimes months before using them. In this way, they can gain reputation for the domain and bypass the detection methods that work based on the registration date. However, according to the report, the number of such domains is low because maintaining a domain name for a long time needs extra effort and money, not always possible for attackers. We use the time lapse between the domain registration and the blacklist dates.

6.3.2.6 Ranking system and popularity features

This feature set consists of 8 features related to search engine results, the Internet Archive [262], and domain name popularity in different ranking systems.

Search engine results ($\mathbf{f_{28}}$). The number of results returned by the Bing search engine for 'site:example.com' queries. The higher the number of results, more popular the domain is. We do not consider Yahoo and Google search engine results because although they are free, with publicly available APIs, the number of requests per day is limited. For example, at the time of writing, the Google custom search engine only allows 100 queries per day. While the Bing search engine is not free (we used the trial version), the price ($3/1000 requests [263]) is low compared to its equivalent alternatives.

Top ranking websites ($\mathbf{f_{29}}$-$\mathbf{f_{32}}$). The presence of the domain name in the Alexa

[264], Majestic [265], Quantcast [266], and Umbrella [267] top 1 Million website and domain ranking lists. While we could merge features $\mathbf{f_{29}}$-$\mathbf{f_{32}}$ into a single one based on the Tranco list [64], each of these ranking systems uses its own metrics to calculate domain popularity and therefore, captures different characteristics. We only consider the presence of a domain in such lists as a sign of its popularity.

Wayback Machine ($\mathbf{f_{33}}$). The Internet Archive project started in 1996 by archiving the Internet itself. The sources of the captures come from different plans of the project, e.g., capturing Alexa top domains, domains that have at least one link from different domains that the Wayback Machine already captured at least one time, and several more plans covering the most part of the Internet [262]. We consider the high number of captures as a sign of benignness for domains.

6.3.2.7 TLD-related features

Chosen TLD is not random among miscreants [11]. They tend to use TLDs based on some factors like the TLD price. We extract two features related to TLD.

TLD maliciousness index ($\mathbf{f_{34}}$). It is a number greater than or equal to zero corresponding to the proportion of abused to all registered domains for each TLD introduced by Spamhaus [268].

TLD price ($\mathbf{f_{35}}$). The price of domains is very important among miscreants since they want to maximize their profit by minimizing the costs. For example, free TLDs (i.e., those provided by Freenom) are among the most common TLDs used in phishing attacks [197, 227].

6.3.2.8 TLS certificate features

COMAR uses three features related to TLS certificates.

TLS certificate price ($\mathbf{f_{36}}$). The purpose of making a TLS certificate available free of cost was to make access to HTTPS available for all websites [269], which means that miscreants can also benefit from it. By using a TLS certificate, attackers can make their attacks look more legitimate (e.g., by showing the green lock in the address bar of the browsers). Free TLS certificates do not either require their owners to provide any personal information. Therefore, the conjecture is that miscreants would prefer to choose free TLS certificates rather than the paid ones.

Presence of TLS certificate ($\mathbf{f_{37}}$). Although the report published by Phishlab [270]

shows that almost half of the phishing websites were hosted on domains with an active TLS certificate, we can still leverage this feature since our analysis is not limited to only phishing attacks.

Valid TLS certificate ($\mathbf{f_{38}}$). Trusted but expired TLS certificates or those issued by untrusted certificate authorities (CAs) trigger an error (or a warning) in most standard browsers (e.g., Chrome or Firefox). This behavior may alert victims about an attack. Therefore, most of the phishing URLs are either HTTP or HTTPS with valid certificates. However, for websites that are used in malware spread or domains for C&C panels, the victims are not humans but infected machines. Having a TLS certificate let the infected machines to communicate with their hosts (e.g., bot masters) securely regardless of the validity of the certificates [246]. For each domain, with a TLS certificate, we define a binary feature indicating whether the certificate is valid or not.

6.3.3 Further Notes on Features

So far, we have introduced 38 features in 7 categories. There are some aspects to consider regarding these features.

- Not all features are available for domain names but some features rely on the presence of other ones. For example, all the content-related features are available if there is a homepage available for the domain name. As another example, a TLS certificate price solely relies on the existence of a TLS certificate (i.e., the domain should be HTTPS-enabled). Such dependency enables suitable handling of missing values discussed below in Section 6.3.4.

- For each type of domain abuse, only a specific set of features may be related to that type. For example, lexical features (more specifically, URL-based features) are mostly used in phishing attacks and are not relevant to algorithmically generated domain names. However, we apply all the features in the classifier and let the classifier decide the relevance of each feature. Then, by interpreting the results, we can choose the appropriate feature set for each type of domain abuse.

- Another important aspect of feature engineering is feature evasion, i.e., how robust each feature is against manipulation. In Appendix 6.10, we discuss potential evasion strategies and how difficult they are for attackers to deploy.

6.3.4 Handling Missing Values

Ideally, the classifier operates on a complete ground-truth dataset without missing values. However, in practice, it is not always possible to collect all the features due to several reasons. Two important considerations regarding missing values are their types as explained by Little and Rubin [271], and the reasons for the absence of data. For example, one important feature in our set is the domain age, which depends on the availability of the registration date. However, it is not always feasible to parse the WHOIS data [193]. Some registries do not provide the registration date as part of WHOIS information or WHOIS data at all (e.g., Freenom registries for .ml, .tk, .ga, or the German registry for .de). Therefore, the lack of the registration date means losing important information, which may result in misclassification. The common strategy to fill missing values is to use statistical methods such as the mean (or median) of the feature. However, the mean and median values may lead to biased results since each sample in the dataset (i.e., each registered domain name) is independent of other samples [272]. Another way to fill missing values is to estimate the best value based on the available evidence. In the case of the registration date, although we cannot find the exact date, we can use the earliest day we observed the domain name in the wild. We use the following formula:

$$\min_{date}\{wayback\_machine, SSL\_certificate, first\_pDNS\} \tag{6.1}$$

with respect to the following constraints:

- Regarding passive DNS, we consider the first seen 'A' (or 'NS') record that matches the 'A' (or 'NS') record of the domain name before the time it was submitted to one of the blacklist resources. The justification comes from the possibility that a domain name was registered by someone before, then re-registered by another user (miscreant) and misused.
- Regarding TLS certificates, we use Certificate Transparency logs to retrieve all the previous certificates of the domain (and subdomains, if any) and extract the issuance date of the oldest one that matches the certificate of the domain name before the blacklisting time.

In this way, we obtain the earliest date the domain appeared in the 1) Wayback Machine

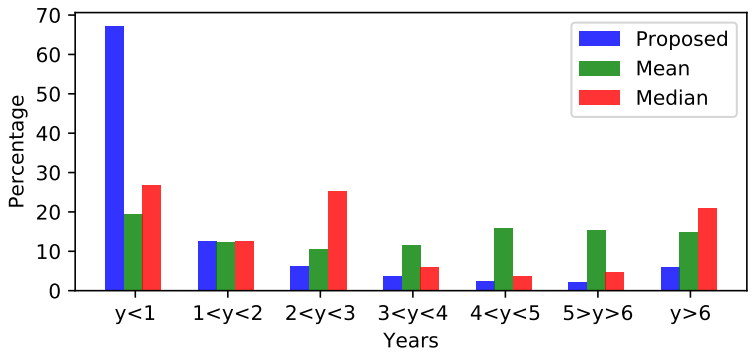


Figure 6.3: Proportion of the domains vs. the difference between real registration dates and the estimated ones using the proposed method, mean, and median approaches.

[273], 2) Certificate Transparency [274, 275], and 3) the passive DNS database [235], which ensures that the real domain registration date is earlier than (or equal to) our estimated value.

Figure 6.3 shows the proportion of the domains for which the difference between the real registration dates and the estimated ones using the proposed method, mean, and median approaches is less than 1 year, between 1 and 2 years, and so on. As the ground-truth data, we use 10,000 domains with different TLDs with known registration dates. For approximately 67% of the domains, the difference is less than one year, while for the mean and median, the result is less than 30%. Furthermore, filling the registration date with the mean for a specific TLD requires to have at least partially the data for that TLD, while for some TLDs (e.g., .ml, .tk), the responsible registries do not provide registration dates at all.

Apart from the registration date, there may be some more missing values in our feature set. For the *'TLD maliciousness index'*, whenever we do not have the data, we fill the value with zero. For *content-related* features, we send requests to domains using the headless version of Selinium and Firefox browsers to mimic user-oriented actions. If we do not get any response from the server, we can assume that the domain has no content to offer to visitors. Therefore, we consider *'content length'*, the *'internal hyperlink'*, and *'external hyperlink'* features as zero. However, in some rare cases, it is possible that the attack type is location-based that either serves the content to specific IP addresses or serves different contents to different IP addresses [276]. In this case, due to our limited resources, we may not be able to fetch the real content. Concerning the TLS certificate price, our approach is to create a binary feature, paid vs. free. However, for

some certificate authorities, there is no clear cut boundary between these two options. For example, Comodo CA [277] (also known as Sectigo) offers both free and paid TLS certificates for domains. For these CAs, we consider the validity period of the certificate. If the validity period is less than three months, then we consider the certificate as free.

In Section 6.6, we compare handling of missing values, data availability, and usage limitations of previously proposed methods with COMAR.

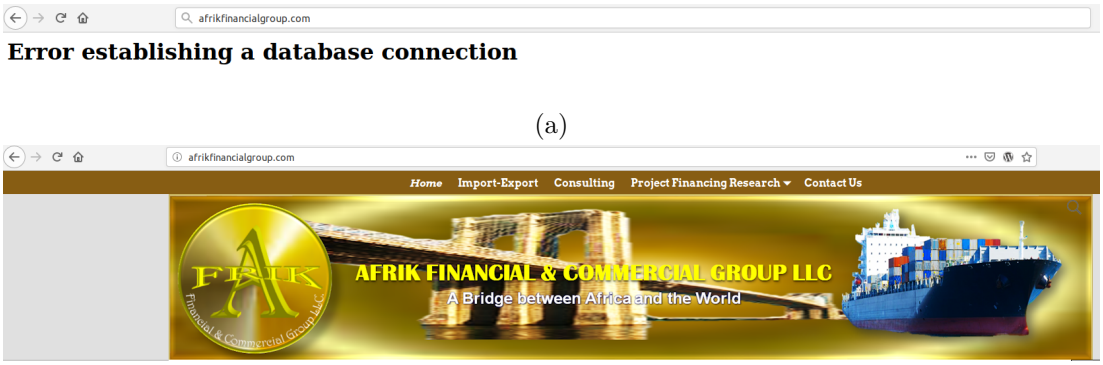
6.4 Experimental Results

In this section, we provide the details of the phishing and malware ground-truth datasets and describe our method to classify compromised and maliciously registered domains.

6.4.1 Ground-Truth Datasets

We have collected 41,002 URLs from four blacklists. Figure 6.8 in Appendix 6.9 shows the number of collected URLs for each blacklist and the overlap between them—it is only the number of working (live) URLs at the time of crawling (March to July, 2019), after removing URL shorteners, free subdomain services, and inactive URLs. Then, we have created two ground-truth datasets from the subset of collected URLs with: 1) URLs from phishing blacklists (APWG, PhishTank, and OpenPhish) and 2) URLs from malware distribution blacklist (URLhaus).

We start with labeling URLs by manually visiting the homepage of the domain and investigating its content and functionality. It is not always trivial even for a human to decide if a domain name is compromised or a malicious one. For instance, it is easy to label 'pyplcompte.fr' (without any homepage and one URL to a fake PayPal login page) as malicious while for 'afrikfinancialgroup.com', the domain name does not contain any suspicious word and the registration time is 2017 but looking at the homepage of the domain, there is only a database connection error description (Figure 6.4a). The error can be the result of an attack or it can be just a simple message to fill the homepage of the maliciously registered domain. To be certain that the chosen label is correct, we re-visit each domain manually after a period of 10 days and check the homepage and the presence of the malicious URL again (the hypothesis is that a 10 day period is long enough for a webmaster to notice that the website is defaced). If the homepage is fixed after 10 days (see Figure 6.4b), we consider the domain as compromised.



(a)

(b)

Figure 6.4: (a) The homepage of the 'afrikfinancialgroup.com' captured in the first scan showing a database connection error. (b) The homepage of the same domain name revisited after 10 days.

We have manually labeled domains as either 1) maliciously registered, 2) compromised, 3) subdomain/free service, or 4) false positive. Although the data collector module automatically excludes free subdomain services, still some of them, which were not in our predefined subdomains list, appeared in the labeling process. After removing subdomain services and false positives (i.e., URLs mistakenly blacklisted) the final datasets consist of 1,321 domains from phishing blacklists and 1,008 malware domains from URLhaus. The proportion of the phishing dataset is 58% malicious - 42% compromised and for the malware dataset 57% compromised - 43% malicious.

6.4.2 Classifier

We use two classification methods: 1) Logistic Regression and 2) Random Forest. We apply each method separately on the malware and phishing datasets. We choose the methods because of their characteristics. Logistic regression is a machine learning algorithm that works on linearly separable data and uses the combination of the weighted input features to predict the output class. It is a parametric method known for its efficiency, low computational resources, and interpretability. However, feature engineering plays an important role with respect to its performance. On the other hand, the random forest is a non-parametric machine learning algorithm capable of training a non-linear model based on the input samples. Generally, it does not need any feature transformation or any assumption about the underlying mapping function. With a sufficient number of training samples, it may result in a better performance model compared to logistic regression [278]. As for evaluation metrics, we use accuracy, precision, recall,

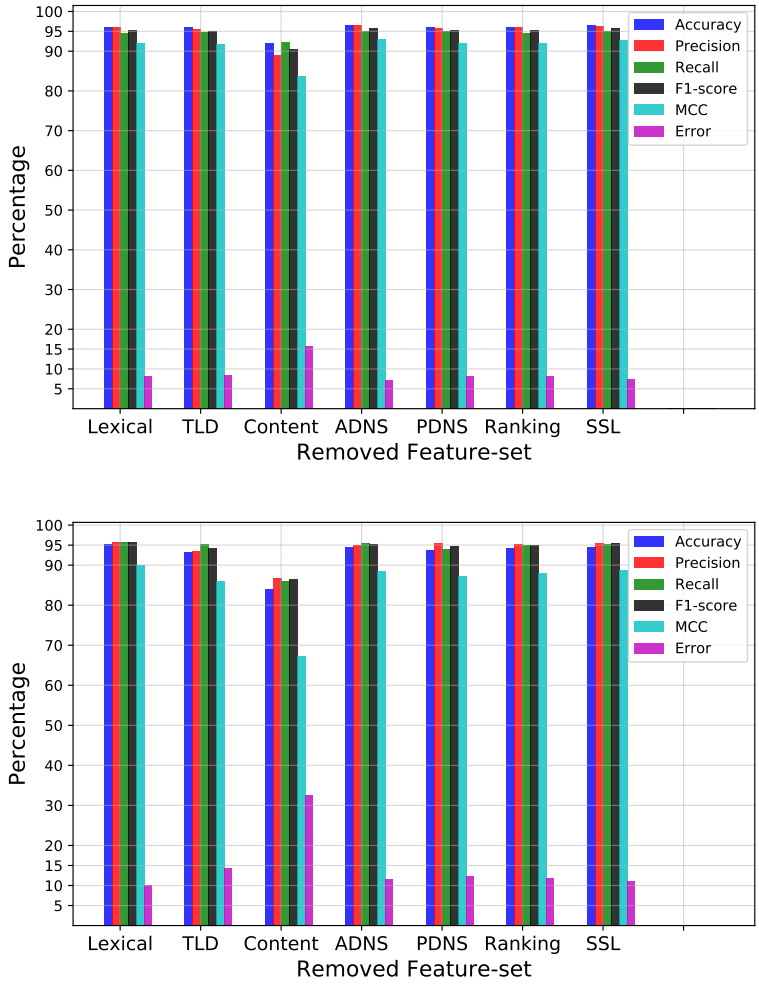


Figure 6.5: Evaluation metrics of phishing (top) and malware (bottom) datasets using logistic regression.

F1-score, and Matthews correlation coefficient (MCC) defined in Appendix 6.8. We use the MCC metric since our datasets are not completely balanced and we also need to consider false positives and false negatives in the final results of the classifier.

Table 6.3 shows the results of the random forest (RF) and logistic regression (LR) classifiers for phishing and malware datasets. We can notice that the classification results of the random forest are slightly better than logistic regression for both datasets. However, we use logistic regression to describe the data and explain the relationship between input features and output classes since it can produce interpretable coefficients.

Figure 6.5 shows the classification results by applying logistic regression on the phishing and malware datasets, and eliminating one feature set at a time. We set the maximum number of iterations to 10,000, using 10-fold cross-validation to evaluate the

Table 6.3: Evaluation of the Random Forest (RF), Logistic Regression (LR), and the APWG method on phishing and malware datasets.

Method	DB	Acc	Precision	Recall	F1	MCC
RF	Phish	97%	95%	97%	96%	0.93
LR	Phish	96.5%	96.59%	95%	95.7%	0.92
APWG	Phish	85%	82%	93%	88%	0.69
RF	Mal	96%	97%	96%	97%	0.92
LR	Mal	94.5%	95.6%	95.2%	95.4%	0.89

algorithm and ridge regularization to create a less complex model and avoid overfitting. The classification error is the sum of false positives and false negatives. A false positive refers to the malicious domains misclassified as compromised and a false negative refers to the compromised domains misclassified as malicious ones. We can observe that removing content-based features can severely affect the results of the classifier both in phishing and malware datasets, and increase the classification error up to 16% and 30%, respectively. On the other hand, removing the *passive DNS* feature set has almost no effect on the final results (Acc: 96.14%, Precision: 95.78%, Recall: 94.91%, F1: 95.34%, MCC: 0.92 for phishing datasets). Moreover, content-based features are more important for malware samples than phishing. The reason is that most of the maliciously registered domains related to malware spreading or C&C panels have no content in their homepages.

6.5 Evaluation of the Results

In this section, we first compare our results with the simple approach used in the 2016 APWG phishing survey [197] to distinguish between malicious and compromised domains. Then, we study the features extracted and used in the classification process. We analyze the 'strength' of each feature (i.e., how it is related to each output class) and select those with the highest impact on the classification results. This section provide a better insight into how we can select the features to create a more effective classifier. We also present three case studies that may influence the classification results.

6.5.1 Comparing COMAR with APWG Method

In the 2016 global APWG phishing survey [197], Aaron and Rasmussen used a simple set of heuristics to distinguish maliciously registered from compromised domains in

phishing attacks. They considered a domain to be malicious if it was reported within a very short time after registration and/or contained a brand name and/or was registered in a batch or there existed a pattern indicating common ownership or intent. Since we do not have access to the registrant's information in the WHOIS data to detect batch registration or any pattern of common intent, we use only the first two conditions to evaluate the APWG method on our ground-truth data. The report did not specify the exact meaning of the 'very short time of being registered', so we chose three months as it is used in the previous study [11]. If the domain has appeared in a blacklist in less than three months of its registration time, or if it has a famous brand name/string in its name, we consider it as a malicious one otherwise it is categorized as compromised.

Table 6.3 shows the classification results of the APWG method. Although the accuracy of the result is relatively high (85%), the false-positive rate is also very high (27%), which results in low MCC (69%). The reason for the high false-positive rate is that the method is unable to detect malicious domains that were registered more than three months before blacklisting and that have no famous brand name or a misleading string as part of the domain name.

In general, there are three limitations of this and other methods that use the registration date as the main feature for classification. As discussed in Section 6.3.4, the registration date is not always available for all TLDs. Therefore, the evaluation is limited to TLDs with the registration date available as part of the WHOIS data. The second drawback is that identifying patterns or evidence of bulk registrations need registration information such as the registrant's name and the address no longer publicly available [178]. Finally, the third caveat of using this heuristic is the fact that it does not consider legitimate domains compromised in the first few months or even days after registration.

Figure 6.6 shows the partial cumulative distribution of the compromised domains after their registration date. We collect the data of hacked websites for 18,810 domains from various resources like accounts of the hacker groups on Facebook, Twitter, and hacking forums for 2 months. The results show that 12% of the domains get compromised in the first three months of their registration, and approximately 32% get hacked in the first year after registration probably because of the lack of appropriate configurations or because the website is still under active development. These types of domains may lead to false-negative results (classifying newly registered benign but

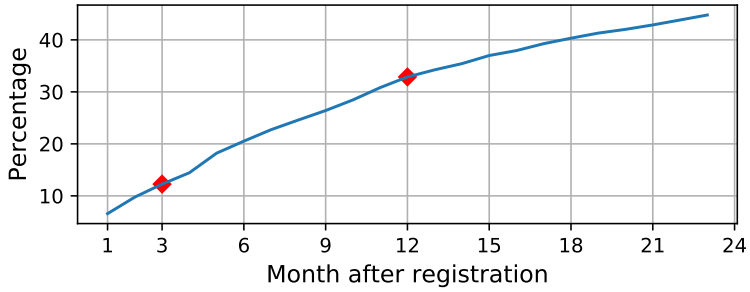


Figure 6.6: Partial cumulative distribution of the compromised domains after registration.

compromised domains as maliciously registered). However, COMAR does not suffer from these limitations since it does not heavily rely on the registration dates (only one feature out of the other 38 proposed ones), and we estimate the missing values of the domain registration dates for TLDs that do not provide the WHOIS data.

6.5.2 Feature Analysis

By using logistic regression, we can measure how important individual features are to the overall performance. Table 6.4 and 6.5 show the logistic regression weights for 24 most significant features. We use the L2 norm regularization to keep the weights small to avoid overfitting and reduce model complexity. Moreover, small weights help us making sure that one feature with a large value cannot heavily affect the final classifier result. We also use log transformation for some features (e.g., *'number of Bing result'*) to increase the linearity between the input features and the output class. The sign of each coefficient shows the relationship between the feature and the *compromised* output class. For example, the *'TLD_maliciousness_index'* feature has a negative relationship with the *compromised* (and a positive relationship with the *malicious*) output class. Therefore, a higher maliciousness index of TLD indicates a higher probability of a domain being maliciously registered rather than compromised.

We can observe that the *'number of internal hyperlinks'*, *'number of Bing search results'*, *'number of technologies'*, and *'content length'* features are in the top five strongest features indicating compromised domains for both malware and phishing datasets. The '*number of internal hyperlinks*', '*number of technologies*', and '*content length*' features are content-based and capture the effort the owner (legitimate or malicious) put into creating a fully-featured website. The results support the conjecture that attackers spend less time to deploy a fully-functional website with rich content since it is time

Table 6.4: Logistic regression coefficients of the significant features for the phishing dataset.

#	Feature	Category	Weights
1	$f_{\text{number of internal hyperlink}}$	Content-based	1.88
2	$f_{\text{number of technology used}}$	Content-based	1.28
3	$f_{\text{Bing search result}}$	Ranking	1.26
4	$f_{\text{content length}}$	Content-based	0.98
5	$f_{\text{first PDNS before blacklist}}$	Passive DNS	0.78
6	$f_{\text{number of PDNS MX}}$	Passive DNS	0.56
7	$f_{\text{TLD maliciousness index}}$	TLD-based	-0.56
8	$f_{\text{domain aging}}$	WHOIS-based	0.49
9	$f_{\text{using redirection}}$	Content-based	-0.46
10	$f_{\text{has vulnerable tech}}$	Content-based	0.41
11	$f_{\text{presence of index page}}$	Content-based	0.39
12	$f_{\text{wayback machine captured}}$	Ranking	0.30
13	$f_{\text{URL has famous brand name}}$	Lexical	0.28
14	$f_{\text{is content related}}$	Content-based	0.21
15	$f_{\text{special word in domain name}}$	Lexical	-0.18
16	$f_{\text{number of external hyperlink}}$	Content-based	-0.17
17	$f_{\text{using HTTPS}}$	SSL-based	0.15
18	$f_{\text{using brand name in domain name}}$	Lexical	-0.12
19	$f_{\text{presence of default homepage}}$	Content-based	-0.10
20	$f_{\text{has SPF}}$	Active DNS	-0.07
21	$f_{\text{self-resolve NS}}$	Active DNS	-0.05
22	$f_{\text{presence in quantcast}}$	Ranking	0.03
23	$f_{\text{using misspelled brand name}}$	Lexical	0.03
24	$f_{\text{presence in umbrella}}$	Ranking	0.02

consuming. Content-based features play an important role in the classification: 5 out of 10 most significant features are content-based. The '*number of Bing search results*' is related to domain popularity, which reflects the conjecture that malicious domains are less popular than compromised domains since they have legitimate traffic generated by benign users.

Another interesting feature is *'number of external hyperlinks'* with different signs for phishing and malware datasets probably because phishers tend to copy the entire HTML code of the target website to create the exact look and feel experience, and most of the time, the cloned HTML code contains hyperlinks related to different pages of the target website. On the other hand, malware domains (e.g., algorithmically generated) often have less (or no) content, which leads to less (or no) external hyperlinks. Therefore, in this case, having an external hyperlink is the indication of a compromised domain.

Table 6.5: Logistic regression coefficients of the significant features for the malware dataset.

#	Feature	Category	Weights
1	$f_{\text{number of technology used}}$	Content-based	0.87
2	$f_{\text{number of internal hyperlink}}$	Content-based	0.84
3	$f_{\text{content length}}$	Content-based	0.82
4	$f_{\text{Bing search result}}$	Ranking	0.74
5	$f_{\text{TLD maliciousness index}}$	TLD-based	-0.72
6	$f_{\text{number of PDNS MX}}$	Passive DNS	0.50
7	$f_{\text{wayback machine captured}}$	Ranking	0.50
8	$f_{\text{presence of index page}}$	Content-based	0.19
9	$f_{\text{number of external hyperlink}}$	Content-based	0.18
10	$f_{\text{domain aging}}$	WHOIS-based	0.16
11	$f_{\text{has vulnerable tech}}$	Content-based	0.14
12	$f_{\text{presence of default homepage}}$	Content-based	-0.14
13	$f_{\text{self-resolve NS}}$	Active DNS	-0.13
14	$f_{\text{is content related}}$	Content-based	0.11
15	$f_{\text{presence in umbrella}}$	Ranking	0.05
16	$f_{\text{using HTTPS}}$	SSL-based	0.05
17	$f_{\text{first PDNS before blacklist}}$	Passive DNS	0.04
18	$f_{\text{using redirection}}$	Content-based	0.04
19	$f_{\text{URL has famous brand name}}$	Lexical	0.02
20	$f_{\text{using brand name in domain name}}$	Lexical	0.01
21	$f_{\text{presence in quantcast}}$	Ranking	0.01
22	$f_{\text{using misspelled brand name}}$	Lexical	0.01
23	$f_{\text{special word in domain name}}$	Lexical	0.0
24	$f_{\text{has SPF}}$	Active DNS	0.0

For URL-based features (e.g., *'URL has famous brand name'*), we observe a significant decrease from phishing dataset to malware dataset because URL-based features are mostly related to phishing attacks. For example, the weight of *'URL has famous brand name'* is 0.28 for phishing while it is 0.02 for the malware dataset.

Considering ranking and popularity features, the presence of the domain name in four ranking websites (i.e., Alexa, Quantcast, Majestic, and Umbrella) has a weak association with the output class in favor of compromised domains in both datasets. Although these features are less significant and it is not difficult to manipulate these ranking lists [64], still using these features combined with others can provide more accurate results.

The *'presence of HTTPS'* feature has a small weight in the phishing dataset (0.15) and near zero (0.05) for the malware dataset, which is not surprising since more than

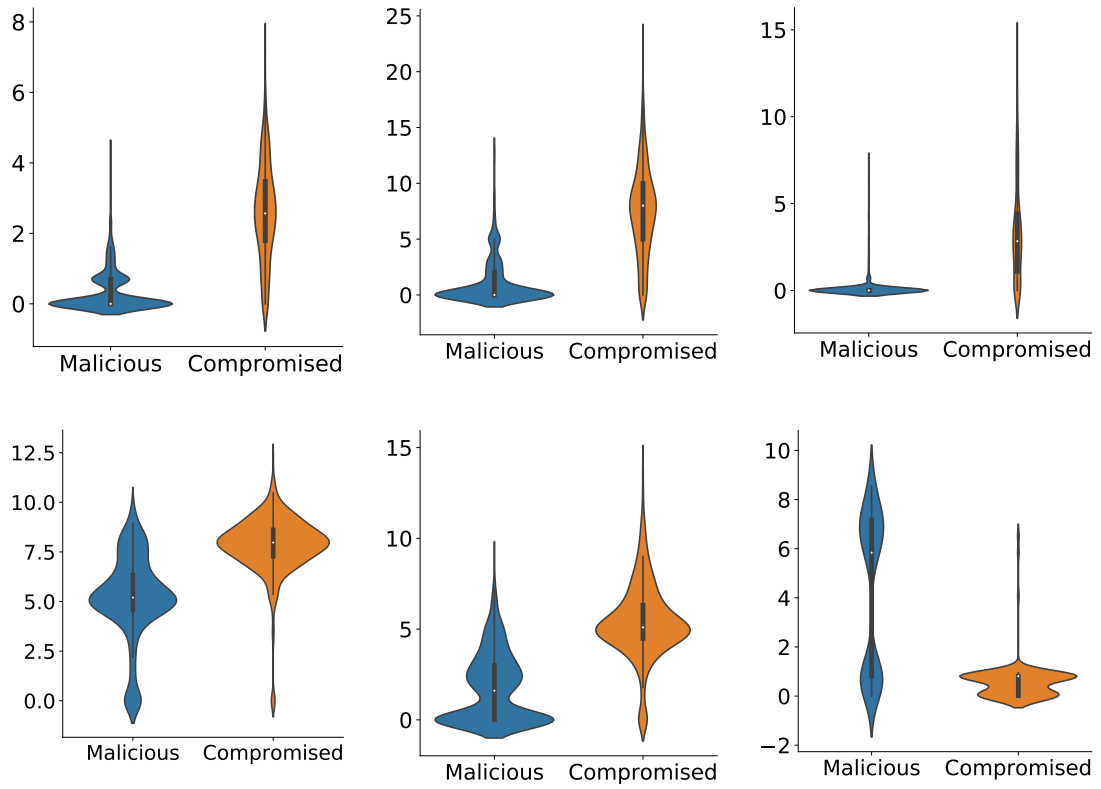


Figure 6.7: Distribution of the *'internal hyperlink'** (top-left), *'number of technologies'* (top-middle), *'Bing search results'** (top-right), *'content length'** (bottom-left), *'number of passive DNS MX'** (bottom-middle), and *'TLD maliciousness index'* (bottom-right) features in phishing datasets. Y-axis is log-transform for *.

58% of the phishing attacks used TLS certificates in the first quarter of 2019 according to the phishing activity report [279]. Therefore, the presence of a TLS certificate cannot be considered as a strong feature to distinguish malicious and compromised domains due to the popularity of using TLS certificates among both attackers and legitimate users.

Figure 6.7 shows the distribution of six selected features for each output class for the phishing dataset. For better representation of the distribution, we use logarithmic scales for the *'Bing search result'*, *'number of passive DNS for MX'*, *'content length'*, and *'number of internal hyperlink'* features. For example, in Figure 6.7 (bottom-left) the average length of the homepage content for compromised domains is greater than for maliciously registered domains. Looking at Table 6.4, the weight of the *'content length'* feature is 0.98 in favor of compromised domains, which means that more content on the homepage is an important characteristic of the benign but compromised domains.

6.5.3 Case Studies

In this section, we present three case studies that may influence the classification results. The first one is related to website defacement when an attacker changes the visual appearance of a website by replacing the index page of the domain. To the best of our knowledge, the second case presents a new technique observed in phishing attacks for the first time. Finally, the third case is related to domain dropcatching in which attackers register expired benign domains to take advantage of their residual trust.

6.5.3.1 Case 1: Homepage Defacement

It concerns a compromised domain name registered back in 2017 but detected by COMAR as malicious. We manually investigated the results, visited the homepage of the domain, and compared it with the data and screenshots from the data collection process. When we found the domain name in the OpenPhish blacklist, the homepage of the domain name was defaced and the content replaced by following HTML code:

```
<html><head></head>
    <body>dddddddd</body>
</html>
```

The COMAR classifier uses 9 content-based features (as explained in Section 6.3.2.2). With the replaced homepage, COMAR was not able to extract features effectively, therefore, misclassified the domain as malicious (with the probability of 67.1% in favor of the malicious class).

As a matter of fact, this result is one of the drawbacks of the content-based features. If we do not fetch the real content of the domain for any reason, the classification results are uncertain. However, homepage defacement is very rare since attackers tend to keep the homepage of the compromised domains as intact as possible to avoid early detection by the website owners.

6.5.3.2 Case 2: New Anti-Phishing Evasion Technique

Phishers always look for new techniques to extend the lifetime of the phishing pages by evading anti-phishing bots and detection systems. One of the best ways to do so is to use defense techniques like page redirection, web-cloaking or server-side techniques like filtering famous user agents like *googlebots* or known scanners IP addresses [280].

As mentioned in Section 6.4.1, we scan each URL and domain twice, within ten days in between, to make sure that the state of both URL and the domain in the labeling process is correct. During our scan, we noticed an URL labeled as safe by Google safe browsing in both scans. Since the URL was in the Phishtank blacklist, which is a community based URL blacklist based on user reports, we had to manually check it to avoid a false positive. By visiting the URL, we noticed that the attacker used Google CAPTCHA to hide the real content of the malicious page. Therefore, even the browser emulation technique was not able to fetch the real phishing content unless a human solves the CAPTCHA manually. Figure 6.9 in Appendix 6.11 shows the website homepage, the phishing page protected by Google CAPTCHA, and a fake PayPal login page for phishing the user's credentials. Although COMAR classified correctly the domain as compromised (since we do not use any feature related to the content of the phishing URL), any phishing (fraud) detection system based on the content of the phishing URL cannot probably automatically fetch the page content. This is the first time we observe an evasion strategy using one of the strongest counter-attack techniques (CAPTCHA). Using techniques like CAPTCHA by phishing attackers may raise a serious challenge to security vendors in detection of malicious pages.

6.5.3.3 Case 3: Domain Dropcatching

Domain dropcatching is the practice of registering a domain name once it is expired and released for new registration [281]. In this process, it is possible for miscreants to register an already expired benign domain name and inherit its residual trust. Miramirkhani et al. [281] showed that approximately 10% of the dropped domains are picked and registered by attackers for malicious purposes. The problem with these domains is that while they should be treated as newly registered domains (as they are), some of the features will match the original registration leading to misclassification of the domains as compromised. The feature sets concerned by drop-caught domains are TLS certificate, passive DNS, and ranking and popularity features. To show the effect of domain dropcatching, we compared the domain registration date with the date related to the first observed DNS query in DNSDB and the first captured page in the Wayback machine only for domains manually labeled as malicious. Whenever the date of the first captured page in the Wayback machine or first seen DNS query is older than the real registration date, we consider the domain as a drop-caught one. In this way, we found

7 samples in our dataset, 6 of them correctly classified as malicious.

Then, we applied the classifier two times on the samples: first, by removing passive DNS features (since they are affected by dropcaching and COMAR does not heavily rely on them) and then, by removing content-based features (since they can be relatively easily evaded and may affect classification). COMAR misclassified 1 and 2 samples (out of 7 samples) in the first and second experiment, respectively. While the number of samples is not enough to evaluate the generalizability of the method in the context of domain dropcaching, we assume that the benign history of the domain may mislead the classifier. We believe that this situation can be worse when attackers clone the content of the original website using the Wayback machine (we have not observed such a case in our dataset).

To reduce the negative impact of the drop-caught domains on the classifier, we could improve the Bing search engine result feature ($\mathbf{f_{28}}$) by only retrieving the results for a specific time slots i.e., after the registration date. Regarding the Wayback machine ($\mathbf{f_{33}}$), we already count only the number of captured pages after the domain registration date. However, passive DNS features and the TLS feature set are still heavily affected by the benign history of the domain and in the worst case scenario, attackers could also consider bypassing content-based features by cloning the content of the original website.

6.6 Related Work

Detecting malicious activity from URLs. Several authors proposed techniques in this category, which makes it one of the most prevalent research topic in the field. The main purpose of these methods is to detect phishing pages and malware C&C panels using machine learning techniques. In case of phishing attacks, Jain and Gupta, for example, proposed a machine learning approach that uses a set of 20 features to identify the input URL as malicious or legitimate [282]. Tian et al. proposed a combination of visual and content-based features to detect phishing attacks [283]. Their assumption is that even if attackers can evade content-based features by using obfuscation techniques, the final appearance of the phishing page should be the same as the target to persuade users to enter their credentials. Tan et al. proposed a phishing detection technique using lexical, URL-based, and content-based features combined with the Google search engine

results to detect phishing URLs [284]. However, in a large scale detection system, it is not feasible to use the Google search engine due to the limitation of the number of requests [285]. COMAR uses some of the features from the above mentioned systems but the primary goal of COMAR is not to detect the malicious content of the URL since we create the domain classification system on top of already blacklisted URLs.

Detecting maliciously registered domains. Several effective methods have been proposed in this category. Although the ultimate goal of these methods is not the same as in COMAR, it might still be possible to apply some of the techniques on each domain in the URL blacklists and potentially identify the malicious ones. NOTOS [38] is a reputation system based on passive DNS queries to rank input domains. It extracts 41 features in three categories: 1) network-based, 2) zone-based, and 3) evidence-based features. Except for two features related to the lexical characteristics of the domain name itself, all other ones are derived from the IP address associated with the domain. NOTOS calculates the reputation of IP addresses, networks, and autonomous systems. Therefore, if the domain is behind a reverse proxy system (e.g., CloudFlare [286]). NOTOS is unable to capture the true IP address and instead, it calculates the reputation of the network related to the reverse proxy rather than the reputation of the true network that hosts the domain. Another limitation is that it needs a large passive DNS dataset to perform well. COMAR does not rely on passive DNS queries and even by excluding passive DNS features, it can still obtain high accuracy with low false positive rate. PREDATOR [219] is a proactive recognition method to detect maliciously registered domains at the time of registration. It uses lexical features, IP-based features, and batch registration patterns to identify malicious domains. PREDATOR suffers from the same limitation as NOTOS in confronting reverse proxies. It also heavily uses WHOIS information and historical WHOIS data, which makes it only practical at registries that have access to such data. Le Pochat et al. proposed an automated method for classifying maliciously registered, algorithmically generated domain names and benign ones that accidentally collide with AGDs, within the constraints of the real-world takedown context of the Avalanche botnets [26]. MENTOR [287] is a system designed to remove benign domains from a blacklist of C&C domains. Both COMAR and MENTOR look for features related to the benign parts of the domains. While the goal of COMAR is to use these features to identify a domain as compromised, the goal of MENTOR is to distinguish benign domains (that are not abused) from malicious ones. One important

caveat of MENTOR is the training and testing datasets. The authors used top 500 domains in the Alexa ranking list as the benign dataset. To form the malicious dataset, they used domains from various blacklists double-checked with the Google safe browsing (GSB) system. However, top 500 domains in the Alexa list are professionally designed, well-structured websites, which make them inappropriate to be used as fair samples of the benign domains in the wild. Moreover, for the malicious training set, if a domain is labeled by GSB as *'not safe'*, it does not necessarily mean that the domain name is completely malicious, since the goal of the GSB system is to detect malicious content (also hosted on benign but compromised domains) rather than malicious domains.

Detecting malicious activity on compromised domains. The main purpose of these methods is to detect malicious activity on compromised domains. Rao and Pais proposed a technique based on Google search engine queries to detect phishing activity [288]. Apart from the limitation of the number of queries, during the manual labelling of the domains in our dataset, we observed that most of the compromised domains are low ranked websites and many of them had been compromised in the first month of their registration and never got indexed by search engines. Corona et al. [289] proposed 11 content-based features along with image similarity combined using a fusion classifier to detect phishing URLs on compromised websites. We leverage some of their features in our work. However, our ultimate goal is not to detect phishing URLs but to classify domains as maliciously registered or compromised ones. Le Page et al. [245] proposed a method to classify maliciously registered domains and compromised ones. They used 15 features in three categories of lexical (5 features), domain name popularity (3 features), and 7 features related to the Internet Archive. Their results show that features derived from the Internet Archive perform the best among all features. However, relying heavily on the Internet Archive may lead to generate feature vectors with a considerable number of missing values since there is, high likely, no archive history for newly registered domains compromised in a short period after their registration.

6.7 Conclusion and Future Work

In this chapter, we present COMAR, a system capable of distinguishing maliciously registered from compromised domains. COMAR leverages publicly available data and makes classification decisions based on the extracted features. Registries, registrars,

and hosting providers can use it to decide on appropriate mitigation actions for each domain with malicious content. It can also serve as an effective tool for creating domain blacklists from the existing URL ones.

We show that the content-based features are the most effective in capturing the 'amount of benignness' of domains during their life cycles. We examine features regarding their robustness and the possible ways attackers can bypass them. High cost and effort for attackers complicates the evasion from COMAR and may therefore discourage malicious actors.

We introduce a new technique to compensate missing values in the 'domain registration date' field of the WHOIS data that outperforms the existing methods. We also show that approximately 12% of the domains get compromised in the first three months of their registration, which suggests that domain reputation systems based on the domain age cannot distinguish maliciously registered from compromised domains with high accuracy.

We plan to deploy COMAR at two European registry operators: SIDN (.nl domains) and AFNIC (.fr domains) and set up an early notification system to contact the owners of compromised domains and domain registrars for maliciously registered domains.

We also plan to correlate the concentrations of maliciously registered domains with a specific registration policy (prices, available payment methods, etc.) at the time of the domain creation. We intend to systematically distill a set of registration features preferred by attackers and analyze individual campaigns as well as long-term trends.

Acknowledgments

We thank: the anonymous reviewers and Thymen Wabeke (SIDN Labs), Pierre-Aymeric Masse (AFNIC), Paul Vixie (Farsight Security) for their valuable feedback; Anti-Phishing Working Group, OpenPhish, PhishTank, URLhaus for providing access to their URL blacklists; Farsight Security for sharing DNSDB, and the DNSDB data contributors. This work has been carried out in the framework of the COMAR project funded by SIDN, the .NL Registry and AFNIC, the .FR Registry. It was partially supported by the ANR projects: the Grenoble Alpes Cybersecurity Institute CYBER@ALPS under contract ANR-15-IDEX-02, PERSYVAL-Lab under contract ANR-11-LABX-0025-01, and DiNS under contract ANR-19-CE25-0009-01.

Appendix

6.8 Evaluation Metrics

We use the following metrics to evaluate our machine learning algorithms.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.2}$$

$$Precission = \frac{TP}{TP + FP} \tag{6.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.4}$$

$$F_1 - score = \frac{2TP}{2TP + FP + FN} \tag{6.5}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{6.6}$$

where *TP*, *TN*, *FP*, *FN* are the number of true positives, true negatives, false positives and false negatives, respectively. Compromised domains are considered positive and maliciously registered domains negative. Accuracy is the ratio of the number of correct predictions to the total number of input samples. Precision means the percentage of relevant results. Recall refers to the percentage of total relevant results correctly classified by the algorithm. The F1 score is the harmonic mean of precision and recall.

The Matthews correlation coefficient (MCC) [290] is a measure of the quality of binary classification. The return value of MCC is between -1 and +1 which +1 represents a perfect prediction, 0 means random prediction and -1 means total disagreement between the predictions and true labels. The advantages of MCC over accuracy and F1-score is that it considers the size as well as the imbalance of dataset. Most importantly, MCC takes into account true and false positives and negatives (all the entries of the confusion matrix not only true-positives and true-negatives).

6.9 Phishing and Malware Datasets

Figure 6.8 shows the Venn diagram of the collected URLs from a) URLhaus, b) APWG, c) OpenPhish, d) Phishtank, and the overlap between them.

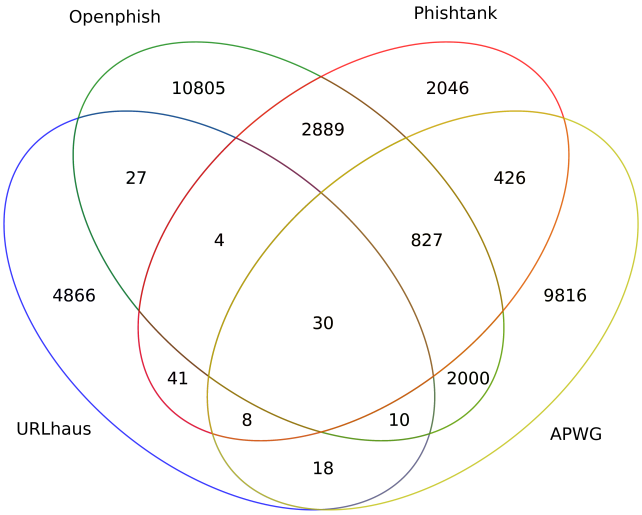


Figure 6.8: Venn diagram of the collected URLs from four blacklists.

6.10 Evasion Techniques

In Section 6.1, we discussed the appropriate mitigation actions for compromised and maliciously registered domains by different intermediaries. For malicious domains, one recommended action is to take down the domain or suspend the hosting service related to that domain. This action may generate extra costs for malicious actors (losing the domain name or the hosting service), which makes it a good reason to avoid their domain being classified as maliciously registered. However, manipulating COMAR features also requires extra effort. In this section, we examine the possibility of feature evasion and associated costs. We take into account i) the amount of money the attackers should pay to bypass a specific feature, ii) the amount of time the attacker should spend to evade each feature and, iii) the necessary skills the attacker should have to bypass a feature.

Generally, it is safe to consider external features as more difficult to evade compared to the features under the control of the attacker. For example, manipulating search engine results, the Wayback Machine as well as passive DNS data are more difficult compared to content-based or lexical features in case of maliciously registered domains. However, it does not necessarily make external features completely bulletproof against manipulation. Furthermore, any feature with a one-time cost (either in terms of time or money) for the attacker cannot be considered as robust.

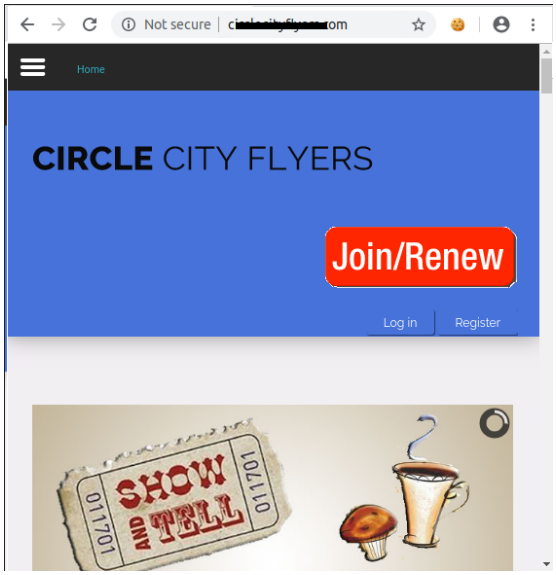
Content-based features. In Section 6.5.2, we show that content-based features are

among the best ones, yet most available in our set. Through this feature set, we exploit the *benignness* of the domain by analyzing 1) the length of the generated content on the homepage of the domain, 2) the relationship between the homepage and other (possible) pages related to that domain (i.e., the number of internal and external hyperlinks), 3) the amount of effort required by the domain owner to design a professional websites (i.e., the number of technologies that are used to create the website), and 4) the number of technologies prone to attacks. We now consider possible evasion techniques the attacker can use to bypass content-related features.

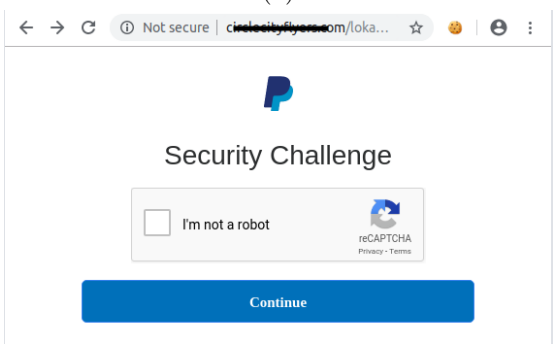
1. **Content length and hyperlinks**. To bypass the content length feature ($\mathbf{f_{10}}$), the attacker needs to generate lengthy content either manually (which is not feasible in large-scale attacks) or automatically through third-party applications. The same methodology can be applied to features related to internal and external hyperlinks (i.e., $\mathbf{f_{13}}$ and $\mathbf{f_{18}}$). Wang et al. [291], studied the effectiveness of black hat search engine optimization (SEO) campaigns to evaluate the possibility of manipulating search engine results for specific keywords by generating fake contents and leveraging various linking strategies. This method can be used to evade features related to the content length and hyperlinks but it requires a fair amount of effort and costs not always available for the attacker.

2. **Technology-related features**. As mentioned in Section 6.3, we use Wappalyzer to enumerate the technologies used by the domain owner to design the website. Wappalyzer is a fast, free, easy to use, signature-based tool able to extract the used technologies by partial string and regular expression matching. Unfortunately, it is also easy to evade. For example, using PHP as a server-side programming language, the default name for the session ID stored as a cookie in the client machine is '*PHPSESSID*'. Wappalyzer uses this name to decide if the server-side code is PHP or not. Therefore, it is possible to mislead Wappalyzer and force it to make a wrong decision on the server-side language just by changing one keyword in cookies. However, decisions can be made using more advanced techniques e.g., hash-based fingerprinting [292].

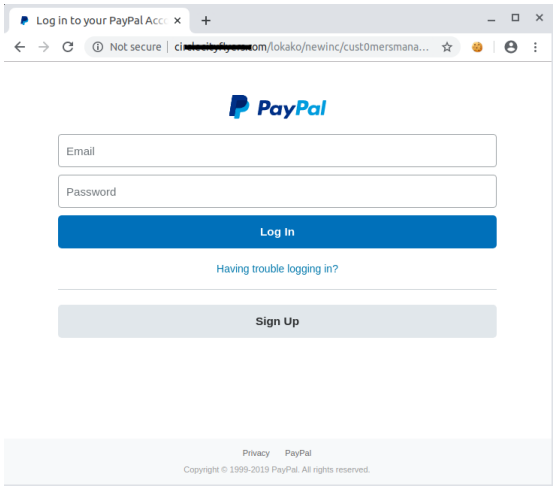
Overall, to evade content-based features, the miscreant must establish a fully-functional website with different content and hyperlinks related to the domain name itself either manually, which takes **time**, or automatically, which impose additional



(a)



(b)



(c)

Figure 6.9: Home page of the compromised domain (a), Google reCAPTCHA with a fake PayPal logo (b), and a fake PayPal login page for phishing user's credentials.

costs on them.

Ranking and popularity features. Manipulating features in this category is not completely under control of the attacker as it represents an external feature. However, it is feasible for a sufficient amount of time and effort.

1. Regarding the Wayback Machine, the attacker can manually submit URLs related to their domains to the Internet Archive project [293].

2. Regarding the Bing search engine results, using SEO techniques (e.g., black hat SEO as explained earlier [294]), it is possible to increase the number of indexed pages for each domain name in search engines.

3. Regarding top ranking websites (e.g., Alexa ranking system), previous research shown that it feasible to manipulate them [64].

However, the cost of evading *'ranking and popularity'* features is related to the **expertise** and **amount of time** the attacker should spend to make her domains as popular as it is necessary to evade the COMAR classifier.

TLD and WHOIS features. COMAR uses one WHOIS-based feature (i.e., *'domain age'*) and two TLD-based features (i.e., *'TLD maliciousness index'* and *'TLD price'*).

1. To evade the *'domain age'* feature, the attacker should register domains long time before using them since we use the number of years before blacklisting, which imposes costs in terms of **money** on the attacker as she needs to register or re-new the domain for a period of a few years to evade this feature.

2. *'TLD maliciousness index'* is another strong feature of COMAR to decide on the state of a domain. One of the factors affecting the value of this feature is pricing. Cheap TLDs (or the free ones) have a higher 'maliciousness' value compared to the expensive TLDs [11, 268]. A higher value of the maliciousness index increases the chance that the domain name is maliciously registered. Therefore, to avoid being detected by the COMAR classifier, the attackers should register domains with TLD suffixes with low maliciousness values, which means they should pay more **money**.

Lexical, passive, and active DNS features.

1. Lexical features are relatively easy to evade. For malware distributors, the domain name is not important since the victims that download the malicious content from the website are not humans but infected machines. However, for phishers, the choice of the domain name is relatively important to conduct a successful attack. For example, `insta-support.com` is more appealing to lure Instagram users compared to the name that has no indication of Instagram.

2. Regarding active DNS features, it is feasible to setup a mail server and/or define, for example, SPF rules in 'TXT' records. However, for attacks performed at a larger scale, the process needs automation.

3. Passive DNS features are the most difficult to evade as the sensors are distributed all around the world. Attackers are not aware of their locations and even if they were, it is not trivial to inject a large number of DNS packets as the monitoring sensors are placed above the local recursive resolvers.

6.11 Captcha Evasion Technique

Figure 6.9 shows a compromised website hosting a phishing page protected by Google CAPTCHA to prevent anti-phishing bots from accessing the malicious page content (details in Section 6.5.3.2).

Chapter 7

A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints

Coauthors: Victor Le Pochat, Tim Van hamme, Sourena Maroofi, Tom Van Goethem, Davy Preuveneers, Andrzej Duda, Wouter Joosen and Maciej Korczyński

7.1 Introduction

On November 30, 2016, a global consortium of law enforcement agencies and Internet stakeholders completed a four-year investigation aimed at dismantling the Avalanche infrastructure [295], which has been called "the world's largest and most sophisticated cybercriminal syndicate law enforcement has encountered" [296]. For seven years, this 'bulletproof hosting service' [297] offered services to cybercriminal operations through a 'crime-as-a-service' model [296], fully managing all technical aspects of carrying out malware attacks, phishing, and spam campaigns. It supported a botnet of a massive scale: Avalanche was responsible for two thirds of all phishing attacks in the second half of 2009 [298], and ultimately affected victims in over 180 countries with estimations of its monetary impact reaching hundreds of millions of euros worldwide [299]. The takedown operation in 2016 was supported by authorities from 30 countries and culminated in five arrests, 260 servers being taken offline and the suspension of over 800,000 domains [295].

As part of this dismantling, a large domain takedown effort sought to disable the

botnet's communication infrastructure. This effort targets the large sets of domains that the malware families of Avalanche generate through *domain generation algorithms* (DGAs). Through this 'domain fluxing' [300], infected hosts attempt to contact all generated domains, whereas the botnet master only needs to register one to continue operating the malware, decreasing the likelihood of blacklisting and takedown. However, as security researchers have reverse-engineered several of these DGAs [300], law enforcement is able to identify upfront which domains the malware will try, after which these can be blocked or seized. Over four yearly iterations of the Avalanche takedown, more than 4.3 million domains were thus prevented from being abused, making it the largest domain takedown so far [301].

Previous work related to DGAs focused on detecting *malicious* domains in regular traffic, relying on strong indicators of *ongoing* malware activity, to discover new malware families or find infected hosts inside a network [302–304]. In this paper, we address the orthogonal issue that the Avalanche takedown faces: given – presumably malicious – DGA domains that will be generated in the future and should *proactively* be taken down, we seek to detect those that accidentally collide with *benign* domains. In particular, we assess how we can effectively support law enforcement investigators with an automated domain classification to inform the appropriate takedown action in a real-world use case. This reduces the extensive manual effort previously invested in this classification, while still maintaining the high accuracy required in such a sensitive operation. Taking down benign domains may cause prejudiced service interruption and harm their owners. At the same time, we have to guarantee that no malicious domain is left untouched, as this would allow malicious actors to target infected users once again.

We are the first to develop an approach that can be used to effectively identify the domains registered with malicious intent, within the constraints of a real-world takedown operation. First, *bulk patterns* no longer apply, both for domains that are benign (due to the accidental uncoordinated collisions) and malicious (due to the low number of required domains). Second, as the takedown is *proactive*, we cannot search for malicious activity (any ongoing activity would mean that infected machines are implicated in actual attacks and defeat the proactive purpose of the takedown). Third, we *cannot actively contact domains* so that the takedown can occur stealthily (otherwise attackers could evade detection and undermine the takedown). Instead, we rely on capturing more generic differences in how benign and DGA-generated malicious domains

are registered and operated.

We design a machine learning-based model for classifying benign and malicious domains, and we evaluate it on ground truth from the 2017 and 2018 iterations. Using a human-in-the-loop approach that combines automated classification and manual investigation targeted at the most difficult domains, we achieve an accuracy of 97.6% for the real-world Avalanche use case, ensuring high correctness while still vastly reducing manual effort: in the 2019 iteration, our approach reduced this effort by 76.9%. However, we go beyond reporting this metric with an extensive analysis of the benefits and limitations brought by the machine learning approach as well as the real-world setting. We provide an interpretation for the factors that impact the decisions of the model, giving insight into how the owners of benign and malicious domains behave differently and how the model uses this information to make decisions. These insights can help law enforcement in their choices regarding the acceptable performance and reliability of the model.

Malware creators increasingly employ techniques that make the takedown of their command and control infrastructure more complex, and the scale of malicious operations continually increases. Further automation of the takedown process with our classifier of malicious and benign domains can support law enforcement in coping with the increased complexity. However, we need to carefully design, evaluate, and analyze such an approach to cope with the constraints of a real-world application as to avoid any adverse effect on the legitimacy of the operation. This enables law enforcement to continue disrupting malware infrastructure and protecting potential victims.
In summary, our contributions are the following:

- We assess to what extent an automated approach can assist law enforcement investigators in correctly detecting the collisions with benign domains among registered domains implicated in the Avalanche takedown, without the ability to rely on bulk malicious registrations, ongoing malware activity or actively collected traffic.

- We develop a technique where we complement a machine learning model with targeted manual labeling of the most informative and difficult domains, to maintain performance across multiple takedown iterations while still vastly reducing the required manual investigative effort.

- We evaluate how well this approach performs and transfers for the 2017 and 2018 takedowns: we obtain an accuracy of 97.6%. The predictions of our model were used in the 2019 takedown, and we find a subsequent reduction in manual investigative effort of 76.9%.
- We critically examine the factors that impact the performance and decision-making process of our model. We find that time-based features are the most important ones, which at the same time are the most costly to evade. In terms of data set availability, WHOIS data greatly improves accuracy, which shows its importance for conducting effective cybercrime investigations.

7.2 Background

7.2.1 Domain generation algorithms

Machines in a botnet such as Avalanche communicate with the malicious actor through command and control (C&C) servers. Early malware hard coded the domain names or IP addresses of their C&C servers, so it was easy to obtain this information and either blacklist the servers or even take over the corresponding infrastructure (by pointing for instance the domains to 'safe' IP addresses and/or having hosting providers take C&C servers down), effectively stopping the malware from further malicious operation [305]. Malware has therefore evolved from hard coding the C&C server information to dynamically creating or updating it.

One technique of this dynamic approach is 'domain fluxing', in which domain generation algorithms (DGAs) create up to thousands of algorithmically generated domains (AGDs) every day [300]. The malware will then attempt to contact these domains and ignore the unavailable ones: the botnet owner therefore only needs to set up one of the generated domains to host a C&C server [305]. Avalanche combined this technique with 'fast fluxing', in which compromised machines hosting a proxy to the C&C server as well as the corresponding DNS entries of the AGDs rapidly switch [306], thus further evading blacklisting and takedown [295].

DGAs take as seeds parameters known to both the malware owner and the infected host, so that they both generate the same set of domains [300, 305]. These parameters such as the length of domains, top-level domains (TLDs) to use, or seeds for pseudo random number generators can be hard coded. More complex algorithms may depend

Table 7.1: Examples of domains generated by Avalanche DGAs.

	Domain	Malware	Validity
1	0a85rcbe2wb5n5fkni4i4y[.]com	CoreBot	Jan 21, 2018
2	researchmadness[.]com	Matsnu	Jan 28-31, 2018
3	arbres[.]com	Nymaim	Mar 9, 2018
4	sixt[.]com	Nymaim	always

on time: one of the inputs to the DGA is then the current time, either from the system clock or retrieved from a common source (e.g., GET requests to legitimate sites [307]). In this way, the DGA creates domains having a certain *validity period*: the time frame during which the seed timestamps make the DGA generate that domain, which the infected machines then attempt to reach. For Avalanche malware families, these validity periods range from 1 day (e.g. Nymaim) to indefinitely (e.g. Tiny Banker).

We can further distinguish between deterministic DGAs that know all parameters upfront and non-deterministic DGAs that know some parameters only at the time of generating the domains: e.g., the DGA of the Bedep family uses exchange rates as seeds [308]. Avalanche did not use any non-deterministic DGAs so for successfully reverse-engineered DGAs [300, 309], we can generate all potential AGDs ahead of their validity, by varying the timestamp that serves as input to the DGA.

Table 7.1 lists example names generated by DGAs, from malware hosted by Avalanche. While Example 1 appears random (a long name with many digits and no discernible words), certain DGAs generate names that look much more like legitimate domains. Example 2 shows a name generated based on a word list yielding domains that may correspond to a regular domain name. Example 3 shows a short yet randomly generated name for which there is a high probability of generating either a valid word or a plausible abbreviation. These last two examples have a high probability of generating domains that collide with existing benign domains.

Finally, certain malware families alter domain resolution on the infected host, generating traffic to hard-coded and otherwise benign domains that actually resolve to malicious IP addresses to circumvent domain-based filters [310]. While these domains are not algorithmically generated, they are present in malware code and traffic and must therefore also be classified as part of the takedown operation, to distinguish them from other hard-coded and actually malicious domains. Example 4 is one such instance

using the domain of the Sixt car rental site. We include these domains in our classification, but for brevity, we refer to all domains to be classified as the 'registered DGA domains'.

7.2.2 Taking down the Avalanche infrastructure

The perpetrators behind the Avalanche infrastructure offered two services for rent by cyber criminals: registering domain names as well as hosting a layered network of proxy servers through which malware actors could control infected hosts and exfiltrate stolen data [309]. Avalanche thereby supported the operation of 21 malware families [311], controlling a botnet of an estimated one million machines at the time of takedown [309].

Prosecutors completed the first iteration of the takedown in November 2016, where the whole infrastructure was dismantled through arrests, server seizures, and domain name takedowns [295]. For the latter, the first iteration targeted live C&C domains, but also those that would be generated by the DGAs in the coming year, preemptively blocking these to prevent Avalanche from respawning. This effort has been repeated every year since, as in January 2020 infected machines on over two million IPs still contacted the Avalanche network [312], highlighting the potential damage if Avalanche were to respawn.

Coupled with the large number of malware families and the extensive amount of domains that these DGAs generate, this results in a large number of DGA domains to be processed. For the three yearly iterations from 2016 to 2018, this amounts to around 850,000 domains per year [301, 311], while the 2019 iteration looks ahead five years and therefore treats almost 2 million domains: this means more than 4.3 million targeted domains have been processed in total. For the DGA domains in the Avalanche takedown, law enforcement took one of three actions on the takedown date [313]:

- *Block registration:* for a not yet registered domain, the TLD registry blocks registration. This is the case for the vast majority of domains.
- *Seize domain:* for a domain registered by a seemingly malicious actor, it is seized from the original owner and 'sinkholed', i.e. it is redirected to servers of the Shadowserver Foundation. Optionally, domains are also transferred to the "Registrar of Last Resort". Through sinkholing, law enforcement can then track how many and which infected hosts attempt to contact the domains [312] and aid in mitigation

Table 7.2: Number of benign and malicious domains per iteration. *: according to our classification.

	2017	2018	2019–2024*
Benign	1397	1014	4945
Malicious	1145	402	1053
Classified	2542	1416	5998
Sinkholed	1177	594	2293
Total	3719	2010	8291

through notifications to network operators and infected users [314]. Domain *seizures* require a legal procedure such as a court order, while organizations could also *request* a takedown through a 'takedown notice' [315].

- *No action:* for a domain registered by a seemingly benign actor (including domains sinkholed by other security organizations), no action is taken by law enforcement and the domain remains with its original owner.

7.3 Problem statement

7.3.1 Making accurate takedown decisions

The aim of the Avalanche takedown is to prevent the botnet owners from interacting with infected machines by blocking access to the required domains that the DGAs will generate in the year following the takedown. However, as these DGAs may generate labels that collide with benign sites, performing a blanket takedown of all generated domains would harm legitimate websites. For Avalanche, public prosecutors therefore first had to manually classify domains into benign and malicious: as shown in Table 7.2, they had to determine an appropriate action for a few thousand registered DGA domains each year.

For registered domains, an incorrect decision may have unintended adverse effects [315, 316]. In case of the seizure of a benign domain, its legitimate owner can no longer provide its service to end users. Owners may experience lengthy downtime, as challenging an illegitimate seizure and regaining the domain can be an opaque and difficult process [315, 317]; it appears that this also holds for Avalanche domains [318, 319].

Conversely, not preemptively seizing a malicious domain allows the botnet to respawn and continue its malicious operation: as the takedown does not remove the malware

Table 7.3: Overview of goals and strategies for the differentiation of benign and malware/DGA domains.

Context/Detection goal	Individual patterns	Proactive analysis	No active connections	Related work
Active malware domains within regular traffic	✗	✗	✓	[303, 322, 323]
Likely DGA domains within regular traffic	✗	✗	✓	[324–326]
Future malicious domains at registration	✗	✓	✓	[327–329]
Benign domains within known malware domains	✓	✗	✗	[330]
Benign domains within future DGA domains	✓	✓	✓	*Our work*

from infected machines, these will continue to establish contact with DGA domains. Once the botnet owners can obtain such a domain, the attackers can launch new attacks or spread malware to additional hosts. The takedown efforts, intended to permanently stop the malware, are then effectively spoiled.

Manually classifying all DGA domains is a resource- and time-consuming process, where due to 'decision fatigue' [320,321], the mental effort in making repetitive decisions could lead to biases. Given the severe consequences of incorrect classifications, our goal is to develop an automated approach to the classification of DGA domains that performs with high accuracy, in order to relieve human investigators from manual effort as much as possible. At the same time, this does not preclude a manual review of those domains that are the hardest to classify or that could have the most significant effects. In the analysis of our approach in Section 7.5, we quantify how such a union of automated and manual classification can still lead to a significant reduction in required effort. Through such a reduction in manual effort and time, we can ensure the correctness of takedown decisions, thereby minimizing negative effects on website owners as well as end users.

7.3.2 Constraints for distinguishing malicious and benign domains

While our base goal is to distinguish malicious and benign domains, we cannot use previously proposed solutions as they rely on certain indicators that would not work for the Avalanche use case. Concretely, these indicators no longer hold for malicious domains (e.g. bulk registration), cannot be observed by us (e.g. detecting malware activity), or are counterproductive (e.g. alerting the attacker). Table 7.3 summarizes how the different contexts, goals and strategies of previous works do not fully satisfy

our requirements.

The reason is that the assumptions made in previous work no longer hold due to a different balance between malicious and benign domains: instead of detecting domains with clear malicious behavior among a (large) set of regular traffic, we assume that domains are malicious (they would be contacted by malware) and need to detect benign domains (i.e. accidental collisions). While in previous approaches, domains that do not exhibit strong indicators of maliciousness (offered by the former) are benign, the absence of such indicators in our use case means that we may not make such an assumption, and makes those previous approaches ineffective for Avalanche.

We translate these unique characteristics of the Avalanche takedown into three constraints. First, we need to take the characteristics of benign domains into account as well, by developing appropriate features that capture *individual differences in registration and configuration*. Second, as we cannot leverage ongoing malware activity itself, we need to develop features that allow for a *proactive analysis*. Third, attackers may not evade or detect data collection, so we may *not make any active connections* to domains in order to remain stealthy. In this section, we elaborate on these challenges and differences that make previous approaches ineffective for our use case.

7.3.2.0.1 Individual registration and configuration patterns Previous work often assumes that specific (bulk) patterns in the setup of domains indicates maliciousness.

For example, PREDATOR [328] relies on the observation that in order to evade blacklisting, malicious spam domains are registered in bulk (over 50% in groups of ten or more at one registrar in five minute intervals), causing these temporal clusters to be similar in infrastructure, lexical composition and life-cycle stage. In a similar spirit, Premadoma [329] relies on similarities in registrant data and the prevalence of malicious domains at specific facilitators (such as registrars) to detect sustained large-scale malicious campaigns. However, these patterns are no longer usable for our set of domains. Attackers only need to register one of the domains that the DGA outputs at a given time, so they no longer need to register domains in bulk, as is necessary for spam domains, also reducing the likelihood that they share e.g. registrars. Figure 7.1 confirms this: 93.5% of malicious domains in the 2017 and 2018 iterations of the Avalanche takedown are registered in clusters of fewer than 10 domains at their given registrar in

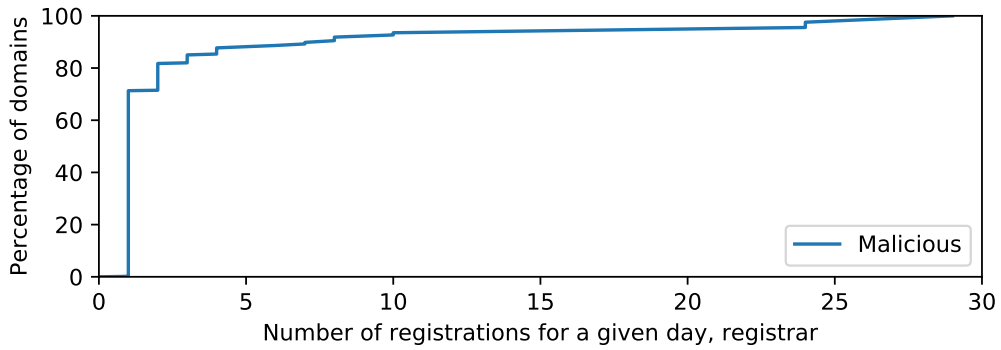


Figure 7.1: Cumulative distribution of registration counts for a given day and registrar, for malicious domains from the 2017 and 2018 iterations.

one day (as opposed to the five minute interval in PREDATOR [328]). Moreover, the accidentally colliding benign sites do not have any relationship and will therefore not share any properties either.

Systems such as DeepDGA [325] and FANCI [324] detect DGA domains from linguistic patterns in their label. However, we know that all domains are either generated by a DGA or hard coded in malware, so it would be incorrect to use such patterns to categorize them as malicious.

In summary, because of the characteristics of our domain set (singular malicious and unrelated benign domains, all output by a DGA), many of the assumptions that the above approaches make on patterns that determine maliciousness are no longer valid. We must therefore resort to capturing more generic, common registration and configuration patterns for individual domains. These patterns should not only capture 'obvious' maliciousness, but also properties that indicate benignness.

7.3.2.0.2 Proactive analysis Previous work relies on observing ongoing malicious behavior: e.g. Exposure [323] leverages irregular DNS configurations and access patterns to detect 'domain flux' [306]; Pleiades [303] captures patterns in `NXDOMAIN` responses to DNS queries by active malware. These systems rely on ongoing malware activity that generates the analyzed traffic. Similarly, systems that use only the label to detect DGA candidates based on their appearance [324–326] need ongoing malware activity, otherwise infected hosts are not contacting malicious domains that are then visible in traffic.

Crucially, because malicious domains have to be taken down before they can cause any harm, we have to classify them proactively, i.e. before infected machines would

actively query the malicious domain. This distinguishes our work from the above works, as we cannot analyze and rely on patterns within any (ongoing) malware activity. While we can and do use features similar to those from previous systems, we are restricted to detecting patterns in registration, configuration, and regular traffic. Moreover, we already know that a DGA generated the domains that we have to classify, meaning that we start with an assumption that the domains are malicious.

7.3.2.0.3 No active connections to domains Internet measurements can be classified into two groups: passive collection, where already ongoing traffic is observed, and active collection, where new traffic is injected into the network. Notos [322] and Exposure [323] are examples of systems that analyze patterns in passively collected DNS queries. In contrast, Mentor [330] relies in part on website content features to measure positive domain reputation, requiring active and targeted data collection through crawling the domains.

While we have a similar goal to Mentor of detecting benign domains within presumably malicious domains, we avoid including features that require us to actively connect to domains. Malicious actors are namely known to detect active scanning and respond differently to appear more benign ('cloaking') [331], and could thus mislead our classification. More broadly, such probes could alert them of efforts to investigate and disrupt malicious infrastructures, allowing attackers to shift their approach or hide any traces to avoid repercussions [309]. A stealthier analysis without targeted active data collection therefore avoids endangering the effectiveness of ongoing investigations [323, 332].

7.3.3 Ground truth data

The advantage of our collaboration with law enforcement is that we can use their manual classification of benign and malicious domains from the takedown as a trustworthy source of ground truth. Previous studies mostly rely on publicly available blacklists and whitelists as the labeled ground truth [333], but malware blacklists have been found to contain benign parked or sinkholed domains and are ineffective at fully covering domains of several malware families [334], while lists of popular domains commonly used as whitelists can easily be manipulated by malware providers [64].

However, the real-world context of the Avalanche takedown affects the composition of our ground truth data. Concretely, our data set is relatively small, as seen in

Table 7.2. Plohmann et al. [300] have seen a similarly small proportion of registered domains among DGA domains. We can expect this number to be small: malicious actors only need to register few domains, as the malware will try all DGA-generated domains; conversely, benign actors are less likely to be interested in using the often random-looking domains generated by the DGAs. Previous studies are able to evaluate their approach on much larger data sets, albeit self-constructed and arbitrarily selected. Nonetheless, training on a small data set is a challenge that prosecutors would also face, and our analysis is therefore valuable for informing them on the feasibility, constraints and benefits of an automated approach for such a practical use case.

7.3.4 Ethical considerations

We use the data set of the Avalanche takedown shared with us by our law enforcement partner. We augment this data with third-party data, avoiding unnecessary active probes of both benign and malicious domains. However, given the sensitivity of the former and commercial agreements for the latter, we cannot share this data with external parties. We release the data processing scripts and resulting models at `https://github.com/DistriNet/avalanche-ndss2020` to support reproducibility.

We assisted law enforcement agencies by applying our approach to the 2019 Avalanche iteration. While the use of machine learning for law enforcement purposes may be contested [335], human investigators may similarly make involuntary errors, e.g. due to 'decision fatigue' [320, 321].

7.4 Data set analysis and feature extraction

To determine a suitable takedown action for algorithmically generated domains (AGDs), we search for relevant features providing a full view of their properties over time. We then create a classifier that detects whether patterns in these properties are more likely to correspond to a benign or malicious domain without having to rely on ongoing malware activity.

In this section, we first analyze how different data sources can track different stages of the domain *life cycle* and we discuss the *insights* on how features capture contrasting properties of benign and malicious domains. Then, we select the final set of *features* and discuss the reasons for omitting certain features.

7.4.1 Life cycle of a domain

To correctly identify the intent of a domain registration, we need to observe patterns in the domain life cycle, as they indicate who obtained the domain, how they use it, and how they value it. For each identified step, we determine which relevant features capture the actions of the domain owner and list sources that track this information. Through our analysis, we can then ensure that our selection of features and data sets appropriately covers each step.

L1. Choice of the domain name The prospective owners of a domain (the registrants) must first choose the domain name that they want to purchase. Usually, the name is chosen to be easily memorized, sufficiently short, and representative of the service provided by the domain, but as malicious actors will need to produce domains in bulk, they will generate them automatically. The resulting names have a random or patterned appearance that we can capture in lexical features on the label itself in order to automatically detect DGAs [324, 325, 336].

L2. Registration of the domain A registrant registers a domain through a registrar, typically paying a registration fee for at least 1 year [337] (although free and shorter offers exist [338] that tend to attract abuse [339]). The registrant identity, the registrar used, and the timestamps of the registration start and end are then made publicly available in the WHOIS database. We can then extract the registration patterns to distinguish benign and malicious sites [340]. Due to privacy concerns and regulations (e.g., the European General Data Protection Regulation), the publicly available identity of the registrant may be obfuscated: the real identity is then only available to the registrar and the top-level domain (TLD) registry. This data may be leveraged in collaborations with registries, e.g. for detecting malicious domains at registration time [329, 341].

L3. DNS configuration Once a domain has been registered, its entry in the Domain Name System (DNS) must be configured to allow discovery of its services using the domain name. The nameserver is passed onto the TLD registry and will appear in its zone files. The domain resource records configured in the nameserver zone file then become available for querying. Active DNS data sets (collected by e.g., Open-

INTEL [342]) rely on scanning zone files or popular domains to obtain these records, while passive DNS data sets (collected by e.g., Farsight Security [91]) extract them from monitored DNS responses. Both types of data sets have been used to detect malicious domain registrations and activity [323, 343, 344].

L4. Setup of the service infrastructure The main purpose of a domain name is usually to provide a service for which an infrastructure needs to be set up. The records stored in DNS may reveal the hosting infrastructure or third-party service providers (e.g., cloud providers) from which actors that enable malicious activity can be derived [345, 346]. A scan of open ports accompanied by "banner grabs" may reveal provided services and the content available through the service may reveal its purpose. Such an operation requires active probing of the domain, which either can be executed ad hoc or is already performed regularly by e.g. Censys [347] and Project Sonar [348], whose scale enables analyses of botnet devices [349]. Furthermore, certificates obtained by the domain owner for their service may also be tracked in Certificate Transparency logs [350].

L5. Service activity Once the service is set up, end users can start interacting with it. Traffic to the service may be logged either at the server, the client, or in any network in-between. These logs can then be analyzed for multiple purposes. Malicious behavior can be detected and publicly shared in blacklists [334, 346, 351]. Commercial providers publish lists of the most popular websites that become base sets of seemingly benign domains [64]. The service may be crawled to populate search engine results or archive web content [352]: the latter enables longitudinal analyses of malicious activity [346, 353, 354]. These methods can be combined to calculate risk scores for the domain [355].

L6. Service unavailability and domain expiration The unavailability of the services offered by the domain, either intentionally or unintentionally due to misconfigurations, may be detected by any of the previously discussed data sets depending on the type of disruption. Once a domain is no longer needed, it may expire: domains that are set to expire are often monitored for drop-catching [356], i.e., registering domains as rapidly after expiry as possible. Malicious actors also reuse previously expired domains to capitalize on the reputation of those domains [357, 358]. Alternatively, a service may be interrupted or a domain may be made unavailable for legal reasons, e.g., in takedown

operations. As we study domains before they would be taken down, we do not consider this last step in our final feature set.

Table 7.4: Overview of the features used in our classifier. We indicate which outcome (benign or malicious) a higher or true value denotes and how the feature covers the domain life cycle and insights.

Set	#	Description	Type	Outcome	Life cycle step (Section 7.4.1)	Insight (Section 7.4.2)	Source
Lexical	1	Domain name length	Continuous	Malicious	L1. Domain choice	i1. Likelihood	[303]
	2	Digit ratio	Continuous	Malicious	L1. Domain choice	i1. Likelihood	[323]
Popularity	3	Number of pages found in Wayback Machine	Continuous	Benign	L5. Activity	i3. Popularity	*New*
	4	Time between first entry in Wayback Machine and takedown	Continuous	Benign	L5. Activity	i3. Popularity	*New*
	5	Time between first entry in Wayback Machine and start of malware validity period	Continuous	Benign	L5. Activity	i3. Popularity	*New*
	6-9	Presence in Alexa, Majestic, Quantcast, and Umbrella top websites rankings	Binary	Benign	L5. Activity	i3. Popularity	[359]
CT	10	TLS certificate found in Certificate Transparency logs	Binary	Benign	L4. Infrastructure	i2. Investment	*New*
WHOIS	11	Time between WHOIS creation date and start of AGD validity period	Continuous	Benign	L2. Registration	i2. Investment	*New*
	12	Time between WHOIS creation date and start of malware family activity	Continuous	Benign	L2. Registration	i2. Investment	*New*
	13	Time between WHOIS creation data and takedown date	Continuous	Benign	L2. Registration	i2. Investment	[360]
	14	Time between WHOIS creation date and WHOIS expiration date	Continuous	Benign	L2. Registration	i2. Investment	[330]
	15	Renewal of domain seen in WHOIS data	Binary	Benign	L2. Registration	i2. Investment	[328]
	16	Domain uses privacy/proxy service	Binary	Malicious	L2. Registration	i2. Investment	*New*
	17	WHOIS registrant email is a temporary/throwaway email service	Binary	Malicious	L2. Registration	i2. Investment	*New*
	18	WHOIS registrant phone number is valid	Binary	Benign	L2. Registration	i2. Investment	*New*
Passive DNS	19	Number of passive DNS queries	Continuous	Benign	L5. Activity	i3. Popularity	[359]
	20	Time between first and last seen passive DNS query	Continuous	Benign	L5. Activity	i3. Popularity	[359]
	21	Time between first seen passive DNS query and takedown	Continuous	Benign	L5. Activity	i3. Popularity	*New*
	22	Time between first seen passive DNS query and start of AGD validity period	Continuous	Benign	L5. Activity	i3. Popularity	*New*
	23-29	Presence of passive DNS query for resource record: `A`, `AAAA`, `CNAME`, `MX`, `NS`, `SOA`, `TXT`	Binary	Benign	L5. Activity	i3. Popularity	*New*
Active DNS	30	Time between first seen DNS record and takedown	Continuous	Benign	L3. DNS config.	i2. Investment	*New*
	31	Time between first seen DNS record and start of AGD validity period	Continuous	Benign	L3. DNS config.	i2. Investment	*New*
	32-36	Number of days DNS record was seen for resource records `A`, `AAAA`, `MX`, `NS`, `SOA`	Continuous	Benign	L3. DNS config.	i2. Investment	*New*

7.4.2 General insights

We want to design features that exhibit contrasting properties of benign and malicious domains and therefore provide a more accurate classification, while still acting within the constraints imposed by the Avalanche takedown use case (as outlined in Section 7.3.2). This requires insights into the generic differences in behavior of legitimate and malicious actors with respect to their domains. We choose our features to capture the following three characteristics:

i1. Likelihood of collisions Given that all domains are algorithmically generated, our target is to find "regular" (least random) looking domains as they are more likely to be a collision with a benign domain, which is opposite to other work that focuses on detecting DGAs solely based on how random their domain names appear [324, 325, 336, 361].

i2. Investment in the domain Obtaining and (validly) maintaining a domain requires an investment from its owner, both monetary for paying the registration fee and in effort for setting up DNS and WHOIS records correctly and installing services attached to the domain. While benign owners value their domains and are willing to

make such an investment, the opposite is true for malicious actors: they want to set up a campaign with minimal cost and effort to maximize their revenue. Certain indicators imply high investment, such as long-term registration (benign domains tend to be older, while malicious domains tend to be registered shortly before the start of the validity period [300, 323, 360, 362]) or valid DNS and WHOIS records (invalid, obfuscated or repeated values hint at malicious practices [341]).

i3. Website popularity Establishing a website that attracts sufficient traffic and is therefore perceived as popular, requires significant effort in creating content and building an audience. Website popularity is therefore an indication of benignness: malicious actors will not make the effort of setting up real websites on dormant domains, especially as it is not required for the correct operation of botnets. Regular users as well as web crawlers are also unlikely to end up on these domains. Moreover, if the domain has not yet been generated by a DGA, its traffic is low or non-existent, so we can assume that any traffic that the domain draws is legitimate.

7.4.3 Summary of feature sets

We aim to capture the broadest view possible of the life cycle of the domains to classify, and select the features and the data sources that provide their values accordingly, further inspired by our general insights. While potentially useful, certain features are not applicable to our use case or would have unwanted consequences for required data collection or wider applicability of our approach. We elaborate on the reasons for not retaining these features in Section 7.4.4.

Table 7.4 gives a summary of the 36 features that we compute. We distinguish between six feature sets: for each feature set, we describe what it represents, which features it includes, how it is obtained, and how complete its coverage is. We indicate for each feature 1) whether it is binary or continuous, 2) whether our intuition is that higher or true values indicate a benign or malicious domain,[1] 3) which life cycle step from Section 7.4.1 it covers, and 4) which insight from Section 7.4.2 is illustrated.

For each domain, we know the start and end dates of their validity period, i.e. when their respective DGA would generate the domain. We also retrieve the date when a malware family started being active from DGArchive [300], where available.

[1] Note that this is only an intuition—our classifier can detect edge cases that provide contrary evidence.

7.4.3.0.1 Two *lexical* features capture the linguistic structure of the domain name. We compute the domain name length, as shorter domains tend to be more popular and expensive, and the ratio of digits in the domain name, as domains with more digits tend to be less readable. Both features discard the TLD.

7.4.3.0.2 Seven *popularity-based* features capture whether a domain hosts a website that appears to attract regular visitors. Three features use data obtained through the Wayback Machine API[2]: the number of unique pages captured on the domain, the time between the first capture of any page and the takedown, and the time between this first capture and the start of the AGD validity period.

Four features capture whether the domain is present at any point in time in the Alexa[3], Majestic[4], Quantcast[5], and Umbrella[6] top websites rankings. These rankings serve as an approximation of popularity from different vantage points: web browser visits, incoming links, tracking script/ISP data, and DNS traffic, respectively. Although they can contain malicious domains and are susceptible to malicious manipulation [63, 64], we assume that presence in these lists still serves as a reasonable indication of benign intent. We retrieve historical data from an archive of historical top websites rankings [64].

7.4.3.0.3 One *Certificate Transparency* feature captures whether Certificate Transparency logs contain a certificate that was valid on the date of the takedown, i.e. whether the owner had obtained a TLS certificate for the domain. The feature in this set uses data obtained through an API from Entrust[7], which tracks Google Certificate Transparency logs [363]. Certificate Transparency logs have the most complete coverage of issued TLS certificates [364]. Recent browser policies that enforce logging further increase uptake [365].

7.4.3.0.4 Eight *WHOIS* features capture the registration cycle of a domain as well as registrant details. We base four features on the time between the WHOIS creation date and the start of the AGD validity period, the start of malware family activity,

[2] https://archive.org/help/wayback_api.php
[3] https://www.alexa.com/topsites
[4] https://majestic.com/reports/majestic-million
[5] https://www.quantcast.com/top-sites/
[6] https://umbrella-static.s3-us-west-1.amazonaws.com/index.html
[7] https://www.entrust.com/ct-search/

the takedown date, and the WHOIS expiration date respectively. For an additional feature, we compute whether the domain has been renewed at least once by the latest registrant, i.e. we find at least two records with different expiration dates.

We capture the validity of registrant data in three features. We determine if the domain uses a privacy/proxy service (replacing real registrant data with generic data) by checking for keywords (e.g. "privacy", "proxy") in the WHOIS registrant records. While legitimate users may prefer to use such a service to hide personal information [366], malicious domains also tend to use these services [367]. We also determine whether the WHOIS registrant email is a disposable address: as the email account can no longer be accessed after some time, this indicates that the owner does not consider the domain to be important. We test non-default/non-proxy email addresses against a manually curated list of disposable domains[8]. Finally, we check whether the WHOIS registrant phone number is valid: malicious actors would not want any trace leading to their real identity and therefore resort to fake (e.g., automatically generated) contact information. We test the validity of phone numbers using an API from numverify[9].

WHOIS-based features are based on historical data generously provided to us by DomainTools[10]. To observe long-term and renewed registrations, we obtain historical records spanning their full data collection period. The data reflects a state before the introduction of the European General Data Protection Regulation, so it contains more domains with publicly available contact details. We elaborate on the continued availability of such details in Section 7.6.2.

7.4.3.0.5 Eleven *passive DNS* features capture both the period and frequency of DNS resolutions for a particular domain, providing a viewpoint on both domain age and popularity. We retrieve the number of passive DNS queries: when more queries (for any resource record) have been made for the domain, the domain appears to be more popular. We base three features on the time between the first seen passive DNS query and the last seen query, the takedown date, and the start of the AGD validity period respectively. Finally, we record the presence of at least one passive DNS query for resource records `A`, `AAAA`, `CNAME`, `MX`, `NS`, `SOA`, and `TXT`: more (requested) record types with a value indicate proper domain setup and usage.

[8] https://github.com/ivolo/disposable-email-domains
[9] https://numverify.com/
[10] https://whois.domaintools.com/

The features in this set use passive DNS data generously provided to us by Farsight Security[11]. We retrieve aggregated data spanning the full data collection period (i.e., since 2010 [91]). For each resource record value seen, the aggregated data contains the number of queries and the timestamps when it was first and last seen.

7.4.3.0.6 Seven active DNS features capture the availability of DNS records for a particular domain. We base two features on the time between the first seen DNS record and the takedown date, and the start of the AGD validity period respectively. We also record the number of days any DNS record value was seen for resource records `A`, `AAAA`, `MX`, `NS`, and `SOA`.

The features in this set use active DNS data generously provided to us by the OpenINTEL[12] project [342]. We cap the data period at 333 days (i.e. starting from January 1 of the relevant year). While OpenINTEL collects data actively, it complies with our requirement that we do not contact domains ourselves. Moreover, data collection is not targeted at specific domains, yet sufficiently comprehensive to also capture most of the registered Avalanche domains as it covers full zone files.

7.4.4 Omitted features

Given our use case of proactive takedowns, we cannot consider features that try to detect ongoing malicious operations directly, as the maliciously registered domain does not yet necessarily exhibit such behavior at the time of the takedown: malicious actors can leave these domains dormant right until a DGA generates the domain and infected hosts start contacting the domain. This means for example that we do not verify whether a C&C server is running on the domain and do not check malware blacklists.

Approaches for detecting AGDs, especially per single domain, are often based on lexical features that seek to discover patterns unlikely to occur in "human-generated" domain names [324,336]. However, all of our candidate domains have been generated by a DGA, which leads us to use only a limited set of lexical features to find the domains that are more likely to be potential collisions (short and few digits).

Detecting patterns from DNS logs [362] that indicate fast flux services [306], often used by command and control servers, is not applicable as the malicious domains would only start operating in fast flux during the validity period of the AGD.

[11] https://www.farsightsecurity.com/solutions/dnsdb/
[12] https://www.openintel.nl/

Following our observation from Section 7.3.2 that bulk patterns do not apply for malware domains, we do not use approaches and features that rely on clustering domains [303] and batches of similar registrations [328], such as timing patterns or shared registrars.

The type of network could be an appropriate feature to take into account while the domain is active [362], with more trust in government or business networks hosting benign sites and domains in residential networks potentially being hosted by an infected machine. However, as a maliciously registered domain does not yet have to be actively malicious before the DGA generates the domain, its IP address can easily be set to a benign network (without the need for that network to actually host the domain) [368], thereby misleading our classifier.

Data collected through a crawl of candidate domains such as properties of the site content could indicate legitimately used domains [330]. However, following our stealth constraint from Section 7.3.2 and due to the need for historical data, we cannot do an active crawl of domains ourselves. We also cannot rely on existing third-party repositories of website crawls (e.g. the Internet Archive [369], Common Crawl [370] or Censys [347]): they do not provide historical data, do not crawl sufficiently regularly to capture recent data, do not have a consistent set of crawled domains and/or do not have sufficient domain coverage. Their data would therefore not be comprehensively representative of domain web content at the time of the takedown.

We do not include the malware family as a feature: as Avalanche provided domain registration as a service [309], we do not expect differences in behavior between the 21 supported malware families. Moreover, such a feature would go against our goal of capturing general differences in behavior between benign and malicious domains. We design the other features to represent distributions, for which the model can interpret the differences, whereas the malware family feature can only serve to refine the model for specific families. Finally, benign domains accidentally 'belong' to a certain malware family, so the feature is irrelevant in terms of registration behavior. We already capture relevant characteristics of the DGA in derived features such as the domain length that capture randomness in generated domains and therefore the likelihood of collisions.

We want to evaluate our approach as if it were deployed at the time of the takedown, so we do not use features for which we lack available historical data, as we would only be able to obtain the current state, which for malicious domains is post-takedown.

They include the features that require active probing or data collection such as the website properties discussed earlier or the existence of search engine results for the domain, which could serve as an additional indicator of popularity. However, if they meet the applicable requirements and constraints, we can add such features in an actual takedown as we can then collect accurate data.

7.5 Analysis of machine learning-based classification

To evaluate to what extent machine-learning based approaches can reduce the effort of law enforcement to execute a takedown, we develop and evaluate a classifier that decides whether future DGA domains are likely to be benign or malicious. The goals of our analysis are threefold: we want to evaluate the raw performance of the classifier, but also gain insights into its decision-making process to finally thoroughly assess the benefits and limitations of automated approaches for domain classification. Moreover, given that not all data sources are equally easy to collect, we assess their impact on the correctness of our classification.

7.5.1 Experimental protocol

We first design an experimental protocol to determine the most appropriate machine learning-based solution and evaluate it in a way that is accurate and representative of real-world takedowns. Given the investigative setting and our intention to thoroughly analyze the resulting model, we restrict our selection of machine learning algorithms to those that are sufficiently interpretable. Moreover, as we systematically develop high-level features that capture the full domain life cycle, we do not require automated feature engineering. Therefore, we would not benefit from a deep learning approach and only face drawbacks from its increased complexity, so we do not consider it further.

Before classifying benign and malicious domains, we discard domains that were already sinkholed by security organizations to study botnet behavior. These organizations can sinkhole the domains either because they detect that botnet hosts are already contacting the domain (whose validity period therefore starts before and extends beyond the takedown date), or because they generate the domains output by the DGA upfront. The sinkholed domains can be considered neither a benign collision, as they do not host real content and may even mimic the malware C&C server, nor a registra-

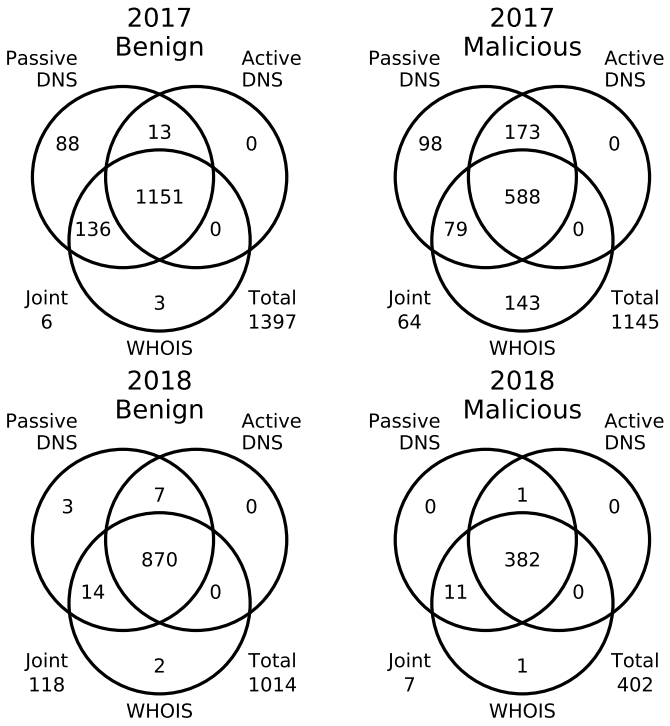


Figure 7.2: Number of domains where certain data sets are available, after removing sinkholed domains, for the 2017 and 2018 iterations. We separately mark the remainder of domains where only the joint data set (comprising lexical, popularity-based, and Certificate Transparency features) is available.

tion made with malicious intent, as they will not communicate with actual malware. This means that they would confuse our model, and should be removed upfront by preprocessing the data. We detect sinkholed domains by matching DNS and WHOIS records with those of the sinkhole providers collected in SinkDB [371], by Alowaisheq et al. [354], and by Stampar et al. [372, 373]. Table 7.2 summarizes the distribution of domains across classes.

We execute our protocol with four machine learning algorithms: decision tree, gradient boosted tree, random forest, and support vector machine. We split data sets in a training and test set according to the considered iterations. When training and testing on the same iteration, we split the ground truth according to a 10-fold cross validation procedure. Otherwise, we construct the training and test sets from the separate iteration ground truths as applicable. We perform all model training and analysis using `scikit-learn` [374]. We elaborate on the different steps of this protocol in Appendix 7.9.

We run our experimental protocol for all domains of the 2017, 2018 and 2019 takedown iterations. We only evaluate performance with the manually labeled ground truth

that we obtained from law enforcement for the 2017 and 2018 iterations (Section 7.3.3). In 2019, our model was used in the real-world classification effort, so a performance evaluation would be biased since we contributed to the ground truth.

As we want to measure the performance of our approach as if it were deployed at the time of the takedown operation, we use historical data that reflects the state of the domains as of each takedown, i.e. November 30 of each year. Data for the malicious domains collected after the takedown would refer to sinkholing and domain transfer infrastructure, making it a signal for maliciousness that would heavily bias our classifier.

As shown in Figure 7.2, we cannot obtain all data sets for all domains: this is because the third-party source could not collect relevant data (e.g. no WHOIS record is available or the domain was never seen at passive DNS sensors). In order to still generate a prediction for all domains, we develop an *ensemble model*. We train a model for each combination of available feature sets, where a domain is included in the training set if at least those data sets are available. To classify a domain, we use the output of the model of the domain's available data sets.

7.5.2 Results

Given that we are the first to analyze the specific issue of preemptively deciding whether DGA domains are actually malicious or accidentally benign for a real-world takedown (which brings about certain constraints), we are not able to compare our performance results with previous work. Instead, we go beyond reporting basic metrics and critically examine how its performance translates into a real-world reduction in effort, whether our solution correctly captures differences between benign and malicious domains, and how much it depends on the availability of different data sets.

7.5.2.0.1 Model performance Appendix 7.10 lists the relative performance of the four machine learning algorithms that we evaluate: we conclude that a gradient boosted tree classifier yields the best performance while still being sufficiently interpretable. We therefore analyze only its results.

We first train a *base* ensemble model, varying the training and test sets over the 2017 and 2018 iterations. From the performance metrics in Table 7.5, we can see that concept drift [375] occurs: performance drops when deploying our model across iterations instead of within. This suggests that over time, patterns that distinguish benign and malicious

Table 7.5: Performance metrics for the base ensemble model, varying the training and test set over the 2017 and 2018 iterations.

Training \ Test	Accuracy		F_1 score		Precision		Recall	
	2017	2018	2017	2018	2017	2018	2017	2018
2017	93.4%	84.3%	92.6%	73.4%	92.6%	70.8%	92.7%	76.1%
2018	76.1%	96.3%	70.9%	93.5%	78.6%	92.7%	64.6%	94.3%

Table 7.6: Performance metrics for models trained on the 2017 and (for the extended model) 15% of the 2018 iteration.

Ensemble model		Accuracy	F_1 score	Precision	Recall	FNR	FPR	Effort reduction
Base		84.3%	73.4%	70.8%	76.1%	23.9%	12.4%	100.0%
Extended	a priori	86.4%	78.6%	70.5%	88.6%	2.3%	2.0%	100.0%
Base	a posteriori	97.3%	95.3%	94.2%	96.5%	3.5%	2.4%	70.3%
Extended	a priori + a posteriori	97.6%	95.8%	94.3%	97.4%	2.6%	2.3%	66.2%

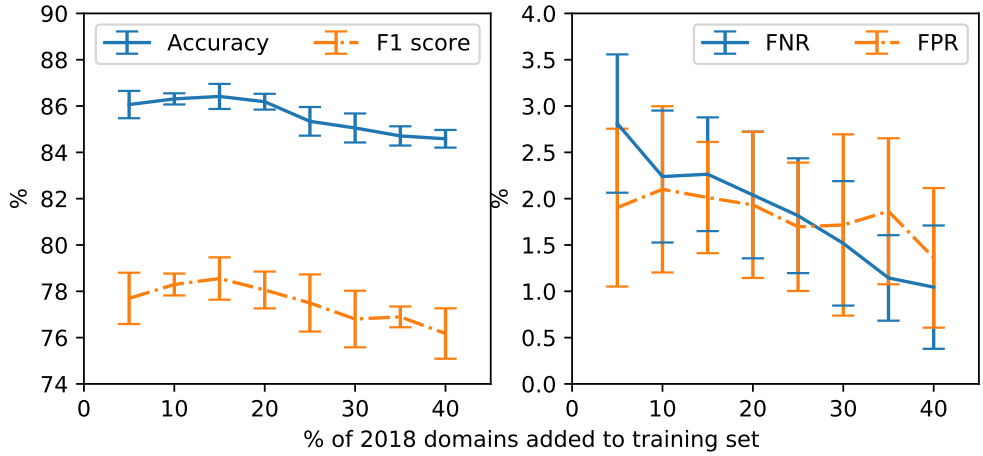


Figure 7.3: Performance metrics (mean and standard deviation) for the extended a priori ensemble model, trained on the 2017 and a varying part of the 2018 ground truth.

actors emerge or change, and these are therefore not captured by a model trained on only a single iteration.

We therefore develop an *extended* ensemble model, where we combine ground truth from a previous iteration with manual, *a priori* classifications of a subset of domains in the target iteration. This enables us to improve model performance by capturing the novel patterns in the new iteration, while still reducing manual effort overall.

We evaluate this extended model trained on all of the 2017 and part of the 2018 ground truth and tested on the remaining 2018 domains. Based on Figure 7.3, we empirically set the proportion of the 2018 ground truth that is (randomly) selected to be manually classified and added to the training set at 15%, as it represents the

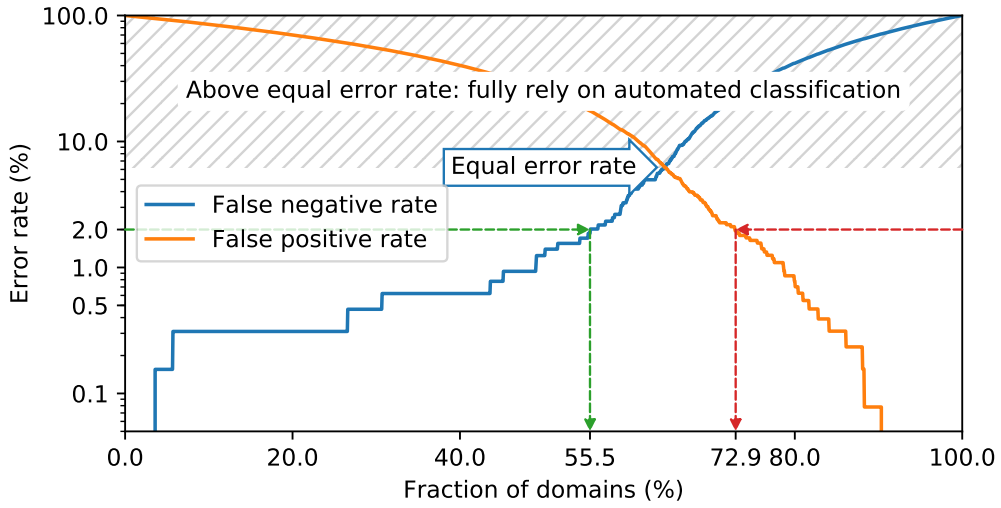


Figure 7.4: FNR and FPR as a function of the fraction of domains with a score below a certain value. By choosing the maximum error rate, we determine the fraction of domains that can be automatically classified.

best trade-off between improved performance and limited additional effort. We repeat this random selection ten times and report average results. Table 7.6 shows that this extended a priori ensemble model improves on the base model.

However, some misclassifications still occur in this extended a priori model. The gradient boosted tree model outputs a score that reflects its confidence in its prediction. We can leverage these scores to develop a directed semi-automated approach: uncertain domains are manually investigated in more detail *a posteriori*. We examine how effective this approach is in further improving performance while still reducing investigative effort.

We explain this approach using the extended model for domains where all data sets are available, which allows us to simplify and visually support our explanation, but then apply it to the extended ensemble model. Figure 7.4 shows the false negative and positive rates as a function of the fraction of domains with a score below a certain value. By choosing a target maximum FNR and FPR, we can determine the lower and upper bounds on the maliciousness score; these bounds are determined based on the training set, so they do not necessarily reflect the exact actual error rates on the test set. Domains with scores within these bounds have to be verified manually, while domains with a lower and higher score are automatically classified as benign and malicious, respectively.

For the extended model on domains with all data sets available as represented in Figure 7.4, when setting a 2% error tolerance, 55.5% of domains have a maliciousness score below the lower bound set by 2% FPR (i.e. are benign), while $(100\% - 72.9\%) = 27.1\%$ of domains exceed the upper bound set by 2% FNR (i.e. are malicious). $55.5\% + 27.1\% = 82.6\%$ of domains therefore no longer need to be manually inspected. Only $72.9\% - 55.5\% = 17.4\%$ of domains still require further manual investigation.

When we apply this a posteriori approach to the extended ensemble model evaluated on all domains from the 2017 and part of the 2018 iteration (by choosing appropriate bounds for each component model), we obtain an accuracy of 97.6%; overall, the performance metrics in Table 7.6 indicate a very high performance. The effective FNR and FPR are 2.6% and 2.3%, comparable to the target error rate of 2%.

Overall, this approach reduces manual effort by 66.2%, accounting for the 15% of domains manually classified a priori. When the error tolerance is 1% and 0.5%, the fraction of automatically classified domains is 52.5% and 35.7% respectively. The score thresholds become very strict when very low error tolerances must be maintained, reducing the fraction of domains that can be automatically classified. The comparable effort reduction for an ensemble model trained on the 2017 and 2018 and tested on the 2019 iteration and a 2% error tolerance amounts to 76.9%, again achieving a significant reduction in manual effort.

7.5.2.0.2 Feature analysis By using gradient boosted trees, we can measure how important individual features are to the overall performance. As we want to make an accurate assessment for the full feature set, we calculate importance scores for the extended model on domains where all data sets are available.

We show the ten most important features in Table 7.7 and find that they primarily capture the age and activity period of a domain. When malware creators want to evade our classifier, they would primarily want to influence these features. Figure 7.5 shows how the distributions of values for the most impactful feature (time between WHOIS creation and expiration date) are clearly distinct for benign and malicious domains. Misclassified benign domains (false positives) actually show a 'malicious' character, i.e. they are young; the malicious domains in our test set (from 2018) are never old, so other (but less expressive) features impact whether they are classified correctly.

Consistent with our second insight from Section 7.4.2, time-based features are costly

Table 7.7: Importance scores of the top 10 features in the full feature set for the extended a priori ensemble model.

#	Set	Feature	Score
14	WHOIS	Time between WHOIS creation and expiration date	0.230
13	WHOIS	Time between WHOIS creation and takedown date	0.219
21	Passive DNS	Time between first passive DNS query and takedown	0.057
20	Passive DNS	Time between first and last seen passive DNS query	0.049
11	WHOIS	Time between WHOIS creation date and AGD validity	0.041
15	WHOIS	Renewal of domain seen in WHOIS data (Unknown)	0.040
34	Active DNS	Days DNS record was seen for resource record `MX`	0.040
15	WHOIS	Renewal of domain seen in WHOIS data (False)	0.037
31	Active DNS	Time between first seen DNS record and AGD validity	0.029
3	Popularity	Number of pages found in Wayback Machine	0.028

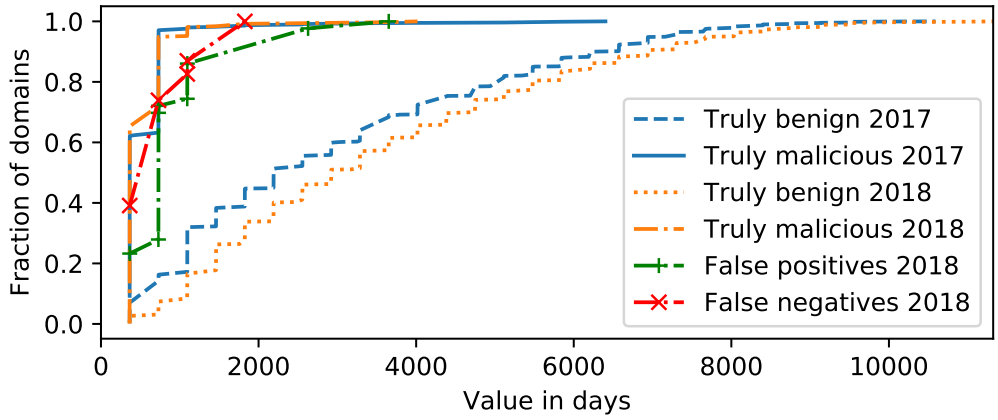
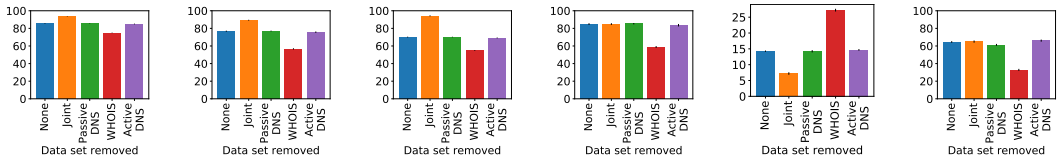


Figure 7.5: Cumulative distribution function of the values of benign, malicious, false positive, and false negative domains for the time between WHOIS creation and expiration date.

and difficult to evade: attackers have to register a domain name for a longer period of time, which translates into a higher monetary cost, and register it earlier, which is hard to achieve retroactively. In an extreme case, the domain name would have to be registered before the malware family becomes active.

7.5.2.0.3 Data set comparison We assess the impact of the availability of each data source on our performance starting from the extended a priori ensemble model, after which we retrain models with one feature set omitted each time. We join lexical, popularity-based, and Certificate Transparency features into a joint feature set, as they are the easiest to acquire and are always available, which leaves us with four feature sets: joint, WHOIS, passive DNS, and active DNS.

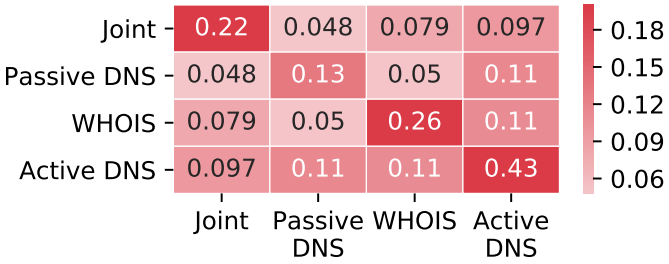
Figure 7.6 illustrates the performance of the models where one data set is discarded.



(a) Accuracy. (b) F_1 score. (c) Precision. (d) Recall. (e) Equal error rate. (f) Reduction in effort (for 2% error rate).

Figure 7.6: Performance metrics (mean and standard deviation, in percent) of extended a priori ensemble models where one data set is omitted.

Table 7.8: Average covariance between features of one set, for the domains from the 2017 and 2018 iterations.



We observe that missing WHOIS data has the most severe impact, significantly harming performance. Discarding the joint data set may actually improve performance, as its non-time-based features may lack sufficiently distinctive patterns, but it remains necessary for domains that lack any other data set (but these are likely candidates for manual verification).

Missing passive or active DNS data has a less pronounced effect. We find some degree of redundancy between passive and active DNS data, as their time-based features in particular represent similar concepts and are therefore intuitively dependent. We confirm this effect with the covariance between feature sets shown in Table 7.8: passive and active DNS data are relatively highly correlated with each other.

This effect means that passive and active DNS (as well as WHOIS) data all capture important and hard-to-evade time-based patterns, but that one missing data set can be substituted by the others without a significant loss in performance. This becomes important when considering that data sets such as WHOIS that lead to better performance may come with a significant cost to acquire. In Section 7.6.2, we elaborate on the implications of our findings on future takedown operations.

7.5.2.0.4 Conclusion We find that an approach combining primarily automated classification and targeted manual investigation across multiple iterations achieves the best compromise of high accuracy and low manual effort, with less than 3% mistakes. This reduces investigative effort by up to 76.9%, depending on the tolerated error rate, freeing up time to focus on those domains that are the hardest to classify.

Our analysis of features and data sets shows that time-based features are the most important ones, which at the same time increases the cost and difficulty of evading our classifier. However, our performance depends on data sources with a high cost of acquisition, in particular WHOIS data. We continue our discussion of these aspects in the next section.

7.6 Discussion

In this section, we elaborate on the factors that may influence the applicability of our approach to future takedowns. We first explain how a high cost and effort for attackers complicates the evasion of our classifier and may therefore discourage malicious actors. We then highlight how recent developments in the availability of data sets may have a negative impact on the performance of our approach.

7.6.1 Evasion

Previous work [328, 340] pointed out that attackers may develop bypasses to mislead a classifier like ours and therefore evade detection and subsequent takedown of their malicious domains, especially as we cannot rely on detecting the malicious activity that would be required for the correct functioning of the botnet. We discuss potential evasion strategies and how difficult they are for malicious actors to deploy. This proactive analysis allows for anticipating changes in attacker behavior, developing additional features that are even harder to circumvent and implementing infrastructural measures that complicate evasion.

Features that leverage the properties of the DGA itself, such as lexical features, can be evaded by redesigning DGAs. While it is feasible to carefully engineer DGAs to be more resilient against detection [361], such a DGA should generate domains that appear very similar to benign domains (e.g., only short domains). This yields a higher risk of collisions and fewer domains available for registration, endangering uninterrupted

control of the botnet.

Popularity-based features require setting up a website for discovery by web crawlers, and generating traffic, or at least the appearance thereof. Website popularity rankings can easily be manipulated at scale [64], allowing attackers to insert their domains and appear as benign. If malicious actors can have a presence within the networks where passive DNS data is collected, they could also insert DNS traffic that makes the domain appear regularly visited. Given that the attackers control their infected machines, the botnet itself could be leveraged for this purpose. However, as the traffic of infected machines can be monitored, these queries can be detected, revealing those domains that the malicious actors have registered upfront. Finally, the presence of certain DNS resource records can be forged by inserting fake records, but as some records require values of a specific format, their validity could be verified, as maintaining valid records requires more effort.

Given recent efforts to increase the ubiquity of TLS encryption by making free and automated TLS certificates available [376], malicious actors can relatively easily obtain them for malicious domains and therefore appear in Certificate Transparency logs. However, such a process still requires additional effort that is not strictly necessary for the correct operation of the C&C server. While the choice to obtain a paid certificate indicates a willingness to invest in the domain (and therefore suggests benignness), the use of a free certificate does not necessarily imply maliciousness.

Features that consider the age of a domain can be thwarted by registering malicious domains (long) before they become valid. However, it requires prolonged registrations and the corresponding payment of registration fees, which runs counter to minimizing the cost of the malicious campaign. Moreover, the longer a domain with malicious intent has been registered, whether active or dormant, the more susceptible it is to being blacklisted/taken down or to the attackers being identified.

Acquiring and managing domains may incur a significant (manual) effort. If the process is automated, certain registration patterns can emerge that make it easier to identify the maliciously registered domains [329, 341]. Malicious actors might attempt to compromise existing or reuse expired domains to exploit the (residual) trust in these domains [358] (for example their age). However, it would require even more effort, as they would need to find eligible domains, attempt to compromise them or monitor their expiration status to take them over at the right time, and finally deploy the

malicious operation. As domains are randomly generated by a DGA and often have a short validity, the likelihood of success is low.

To circumvent features that use WHOIS registrant records, malicious actors could insert forged yet realistically-looking data. However, if these records are automatically generated, detection becomes feasible and accurate [329,341]. Manual effort in creating fake records quickly becomes infeasible given the need to keep registering domains as they become (in)valid.

In summary, while the publication of features allows for an attacker to develop techniques to evade them, many of these would go against the goal of malware operators to set up these domains with low effort and at low cost. Moreover, if the attacker behavior would significantly shift, other evasion countermeasures and detection strategies remain available, although they might require increased effort and involvement by relevant stakeholders. Finally, we find time-based features to be the most important ones: they are particularly costly and hard to evade.

7.6.2 Availability of data sets

Our features come from different data sources that each present their own issues in terms of acquisition, affecting not only law enforcement but also adversaries seeking to evade the model. Moreover, our evaluation of the importance of different data sources for correctly classifying domains shows that the data sets that contribute the most to our model's performance have a significant cost in terms of money and effort.

WHOIS data in particular provides the highest accuracy, but obtaining it may be challenging. From a technical standpoint, WHOIS data is not machine-readable nor has a standard format [377], so it requires (sometimes manual) parsing. Moreover, access is rate limited [378].

Public availability of WHOIS data is also affected by privacy concerns [379] as well as strict limitations on the collection and dissemination of personal data due to privacy regulations. This triggered ICANN to adopt the "Temporary Specification for gTLD Registration Data", which allows generic TLD registries to redact personal data in WHOIS records, while having the intent to provide vetted partners such as law enforcement agencies with privileged access [178]. As a result of the European General Data Protection Regulation, European country-code TLD registries have also started to withhold personal data [380]. Security researchers have voiced concerns that the

unavailability of such data to them could significantly hamper efforts to identify and track malicious actors [381, 382].

Passive DNS data collection may also have privacy implications [343], and requires sufficient storage and processing resources. Active DNS data collection has similar storage and resource needs, especially to ensure that records are updated sufficiently frequently. The coverage of both data sets also depends on cooperation of third parties: passive DNS requires access to recursive resolvers ideally deployed all over the world, and active DNS collection often relies on zone files that must then be shared by registries. Although law enforcement may gain more extensive access, they may be more limited in terms of resources, and delays in procedures to obtain data may hamper swift action. Conversely, commercial providers that can deploy more extensive resources may not be able to access more sensitive information. Finally, from a cost perspective, these commercial providers may charge significant amounts, especially for historical data.

We see that our approach becomes less effective if certain data sets would be unavailable, and our discussion shows that comprehensive coverage of data sets comes at great cost. However, we can still achieve reasonable performance even with missing data, and we see that data sets are partially correlated. The continued availability of these data sets is therefore important to counter future malicious operations, but not to such an extent that their absence would be disrupting the effectiveness of takedowns.

7.7 Related work

Classifiers for detecting malicious domains: Numerous works have addressed the problem of designing classifiers to distinguish benign from malicious web pages and domains. Ma et al. [340] classified malicious URLs based on lexical and host-based features, comparing multiple feature sets and classifiers. Felegyhazi et al. [327] designed a classifier seeded with known malicious domains that uses DNS and WHOIS data. Antonakakis et al. [322] proposed Notos, which outputs a reputation score based on the determination of the reputation of domain clusters obtained from network properties, DNS data, and the ground truth on benign and malicious domains. Bilge et al. [323,362] proposed Exposure, which uses DNS-based and domain name features to detect domains contacted by infected machines within passive DNS traffic. Frosch et al. [360] proposed Predentifier, which combines passive DNS, WHOIS, and geolocation data to detect

botnet command and control servers. Hao et al. [328] proposed PREDATOR, a classifier for malicious domains based on features available at the time of registration and the identification of batch registrations. Spooren et al. [329] developed Premadoma, a model to detect malicious domains at the time of registration, leveraging features based on infrastructural reputation and registrant similarity, and discussed the challenges and tactics for deploying the model in an operational setting. Machlica et al. [383] created a model that uses two levels of classifiers to improve detecting malicious domains using lexical and traffic-based features. Kidmose et al. [384] and Zhauniarovich et al. [332] surveyed approaches to detecting malicious domains from (enriched) DNS data.

Classifiers for detecting algorithmically generated domains: Earlier work in detecting algorithmically generated domains (AGDs) identified clusters of likely candidates. Yadav et al. [302,307] evaluated several statistical measures for classifying groups of domains as algorithmically generated or not based on character distributions within the domain names and the IP addresses to which they resolve. Yadav and Reddy [385] applied similar statistical measures on successful and failed domain resolutions. Antonakakis et al. [303] proposed Pleiades, which clusters non-existent domains based on character distributions within the domain names and on the querying hosts, using the strategy on DNS traffic from large ISPs to discover six DGAs that were unknown at that time. Krishnan et al. [386] detected hosts in a botnet by analyzing patterns in DNS queries for non-existent AGDs through sequential hypothesis testing. Mowbray et al. [387] detected hosts that query domains with an unusual length distribution, deriving 19 DGAs of which nine were previously unknown.

Later work moved towards detecting AGDs per single domain name. Schiavone et al. [336] proposed Phoenix, which uses linguistic features to detect potential AGDs, afterwards using linguistic, IP-based and DNS-based features to cluster domains and extract properties of the DGAs that generated them. Abbink and Doerr [388] and Pereira et al. [389] highlighted how most classifiers focus on detecting the randomness in AGDs and are therefore not able to correctly classify dictionary-based DGAs, and proposed new methods for detecting such DGAs. Multiple deep learning-based approaches have since been proposed [304]. Spooren et al. [361] found one such deep learning model by Woodbridge et al. [325] to outperform the human-engineered features of the model by Schüppen et al. [324].

Takedowns of botnet infrastructures: Previous coordinated takedowns of botnet

infrastructures have been studied to evaluate their effectiveness over time in preventing further abuse. Nadji et al. [390] presented rza, a tool that uses a passive DNS database to analyze and improve the effectiveness of botnet takedowns. They evaluated the tool for three malware families and found mixed long-term impact of takedown operations. Asghari et al. [391] analyzed the institutional factors that influenced the cleanup effort of the Conficker worm, finding that cleanup was slow and that large-scale national initiatives did not have a visible impact. Shirazi [392] surveyed and taxonomized 19 botnet takedown initiatives from 2008 to 2014. Plohmann et al. [300] analyzed the structure of DGAs for 43 malware families and variants, and analyzed registrations of their AGDs, finding domains missed in takedowns, families for which few domains were sinkholed, and slowness in seizing AGDs registered by malicious actors. Alowaisheq et al. [354] studied the life cycle of takedown operations across sinkholes and registrars based on passive DNS and WHOIS data, finding several flaws that would allow malicious actors to regain control of some sinkholed domains. Hutchings et al. [315] provided insights into the effectiveness of takedown efforts by interviewing key actors, finding that law enforcement faces more challenges than commercial enterprises in effectively carrying out takedown operations.

7.8 Conclusion

Taking down the domains that compromised machines use to communicate with command and control servers is an effective measure to disrupt botnets such as Avalanche. However, law enforcement must take care not to affect any legitimate domains that happen to collide with algorithmically generated domains. For Avalanche, prosecutors manually conducted this classification process, requiring large amounts of time and effort as well as allowing for human error.

We therefore develop an automated approach for classifying benign and malicious registered DGA domains, within the constraints of the real-world takedown context that make previous approaches inapplicable: we cannot rely on bulk patterns, detecting ongoing malware activity or actively connecting to domains. We propose a hybrid model that balances automation with manual classification to achieve a higher performance as well as vastly reduce investigator effort. We develop and evaluate our approach to represent the Avalanche takedown most truthfully, such that our results and findings

reflect the utility of automated domain classifiers in a real-world takedown scenario, such as for our contribution to the 2019 iteration.

Given the increasing number and size of cybercrime operations, automated tools can assist law enforcement investigators in avoiding any harmful impact of their operation, especially on uninvolved legitimate parties. These tools will allow them to stay one step ahead of malicious actors and impair their activities with the goal of shielding end users from any harm.

Acknowledgment

We thank the reviewers for their valuable and constructive feedback, as well as the Security Analytics SIG at DistriNet, the Drakkar group at LIG, and Paul Vixie. We thank our partners for providing access to the Avalanche ground truth data: Benedict Addis of RoLR, the Shadowserver Foundation, Sascha Alexander Jopen and his team at Fraunhofer FKIE, and the law enforcement agencies involved. We thank Farsight Security for providing access to the DNSDB data as well as the DNSDB data contributors; DomainTools for providing historical WHOIS data; the OpenINTEL team, in particular Roland van Rijswijk-Deij, for their help in obtaining the OpenINTEL data; Roman Huessy at abuse.ch for the SinkDB data; and Daniel Plohmann for access to DGArchive.

This research is partially funded by the Research Fund KU Leuven. Victor Le Pochat holds a PhD Fellowship of the Research Foundation - Flanders (FWO). This work was partially supported by SIDN, the .NL Registry and AFNIC, the .FR Registry under the COMAR project. The research leading to these results was made possible by OpenINTEL (https://www.openintel.nl/), a joint project of SURFnet, the University of Twente, SIDN and NLnet Labs.

Appendix

7.9 Machine learning protocol

Machine learning algorithms are trained on a training set Tr and evaluated on a test set Te. As explained in Section 7.5, if we need to train and test on the same iteration, we split using a k-fold cross validation procedure: the data is split in k folds, with every

fold being used once as the test set, while we use the $k-1$ others for training, and finally, we average results over k experiments. We set k to 10. The advantage of using cross validation is that we can reduce bias in the composition of the selected training and test set, even with a relatively small data set.

Most ML algorithms have different hyperparameters to tune. Tuning on the test set would lead to highly biased results. Therefore, we have to split the training set Tr into a set for training Tr' and another one for validation V. We again use a 10-fold cross validation procedure. We treat and calculate the upper and lower bounds for the extended a posteriori model as hyperparameters.

We evaluate the following performance metrics over the test set:

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{7.1}$$

$$precision = \frac{tp}{tp + fp} \tag{7.2}$$

$$recall = \frac{tp}{tp + fn} \tag{7.3}$$

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{7.4}$$

where tp, tn, fp, fn stand for the number of true positives, true negatives, false positives and false negatives, respectively. Malicious domains are considered positive, benign domains are negative. Precision represents the fraction of samples identified as malicious that are actually malicious, while recall represents the fraction of malicious samples that were correctly identified. The F_1 score summarizes these two metrics, and is a superior metric compared to accuracy when dealing with unbalanced datasets, therefore we optimize for it.

Due to incompleteness of our data sets (e.g., WHOIS records not containing a parseable phone number), certain domains have missing feature values. We impute them (i.e., substituted them with plausible values to avoid bias) as follows (the feature numbers correspond to those defined in Section 7.4.3):

- No Wayback Machine data: feature values (3-5) are set to zero as no data means that the Wayback Machine has not found any page on the domain, suggesting un-

popularity.

- No WHOIS timestamps: feature values (11-14) are set to the mean, as no data implies that data could not be parsed or retrieved, not that the data does not exist (e.g., all domains have a registration date). By using the mean, we do not attach any statistical meaning to the absence of data and do not skew the distribution.
- Less than two WHOIS records: the renewal feature (15) gets a third value that indicates that only one historical WHOIS record was available (preventing a comparison of expiration dates).
- No WHOIS registrant records: features that rely on an address, an email address, or a phone number (16-18) get a third value that indicates that we do not have a value for the corresponding field.
- No passive or active DNS data: continuous feature values (19-22, 30-36) are set to zero and binary feature values (23-29) to false as no data means that DNS records for the domain were never queried, suggesting unpopularity.

7.10 Evaluation of machine learning algorithms

Table 7.9 presents the performance metrics of the machine learning algorithms that we evaluate in Section 7.5.2, for a base ensemble model trained and tested on the initial 2017 iteration. The results show that gradient boosted trees consistently outperform the other ML algorithms.

Table 7.9: Performance metrics of the evaluated machine learning algorithms.

Metric	Decision Tree	Gradient Boosted Tree	Random Forest	Support Vector Machine
Accuracy	88.6%	93.4%	92.8%	86.4%
Recall	86.6%	92.7%	92.6%	77.9%
Precision	87.8%	92.6%	91.5%	90.6%
F_1 score	87.2%	92.6%	92.0%	83.8%

Chapter 8

Conclusions

This dissertation focuses on improving global DNS security using traffic measurement and data analysis approaches. We have presented six selected contributions. The first three DNS measurement studies shed light on significant weaknesses inherent in the design of Internet protocols that can affect the correct operations of DNS infrastructures and domain names. In the subsequent three studies, we have presented statistical and machine learning approaches to support the community with inferential analysis and practical tools to combat domain name abuse more effectively. The presented contributions are highly empirical in nature. We actively engaged with the industry (practitioners and regulators) to raise awareness of the DNS-related cybersecurity issues we studied and, whenever possible, made proposed solutions available.

We have presented the first measurement study into the vulnerability of non-secure DNS dynamic updates, which enables an attack we referred to as *zone poisoning*. Initially, we have measured prevalence rates for a random sample of 2.9 million domains and for the Alexa top 1 million domains and found that the vulnerability poses a serious security flaw that deserves more attention from domain owners and DNS service operators. At the time of writing, we have extended our measurements to the global domain name population and have discovered approximately 400,000 domain names vulnerable to zone poisoning. Thanks to our repeated and sustained outreach campaigns to the affected parties (i.a., via notifications to national CERTs), more than 99% of misconfigured domain names were secured.

Next, we have presented a novel method to infer the deployment of inbound SAV for the IPv4 and IPv6 address spaces and performed the first global study of the problem. We measured the filtering policies of 52% of routable IPv4 autonomous systems

(26% for IPv6) and 28% of all the IPv4 BGP prefixes (almost 9% for IPv6). We have shown that most of the networks for which we obtained measurements are consistently or partially vulnerable to inbound spoofing. Note that the absence of inbound SAV makes closed DNS resolvers exposed to several types of DNS-based attacks, including zone poisoning and cache poisoning attacks, NXNSAttack, or potentially zero-day vulnerabilities in the DNS server software. To draw attention to the problem of inbound spoofing, we launched the Closed Resolver Project at `https://closedresolver.com` in collaboration with RIPE NCC and national CERTs. The ultimate goal is to run notification campaigns for network operators and provide them with an accessible platform to investigate and mitigate the vulnerability in their networks.

We also evaluated the adoption rates of the SPF and DMARC email security extensions for the global domain name population, and in particular high profile domains. We analyzed the potential for domain name spoofing posed by the absence of their rules or misconfiguration. The results show that a large part of the domains do not correctly configure the SPF and DMARC rules, which enables attackers to deliver forged emails to user inboxes successfully. For remediation, we have sent emails to notify the CSIRTs responsible for almost 6,5 million domains. As many as 23.2% of high-profile domains were re-configured at the end of our notification campaign. Our experience shows that disclosing vulnerabilities through CSIRTs can be effective, especially for valuable domain names.

We have presented the first comprehensive study comparing the rates of malicious and abusive behavior in the legacy gTLDs (e.g., `.com`, `.net`, `.biz`) the new gTLDs (e.g., `.top`, `.paris`, `.xyz`) introduced by ICANN as of 2012. While the number of abused domains in legacy gTLDs seems to stay relatively constant over time (or in some cases decreasing), new gTLDs that underwent rigorous application and evaluation process by ICANN are more frequently affected by phishing, malware, and especially spam activities. Investigating the relationship between structural and security-related properties of new gTLD operators and abuse counts revealed systematic and anecdotal evidence that low domain registration prices, unrestrictive registration practices, and the increased availability of domains decrease barriers to abuse and seem to make some new gTLDs very attractive for miscreants. Taken together, our findings indicate that the existing safeguards did not prevent domain name abuse. Therefore, we have further developed cases for modifying the existing safeguards and proposed new ones, which

we extensively discussed with the ICANN community and the more recently with the European Commission.

One limitation of our inferential analysis, explained above, was the use of a simple approach to distinguish maliciously registered domains from compromised sites. Therefore, we proposed to develop the COMAR system for this purpose. It uses publicly available datasets and makes classification decisions based on 38 extracted features. Registries, registrars, and hosting providers can use it to decide on appropriate remediation actions for each domain containing malicious content. It can also serve as an effective tool for blacklisting domains from existing URL lists. We plan to deploy COMAR at two European registry operators and create an early notification system to contact owners of compromised domains and domain registrars for maliciously registered domains. The results of this project were used in multiple discussions with the European Commission and the ICANN community on the definition of DNS abuse. As part of future work, we intend to systematically distill the set of registration characteristics preferred by attackers and analyze individual campaigns and long-term trends in domain name abuse.

Finally, we developed an automated approach for classifying maliciously registered DGA domains and legitimate domains that happen to collide with algorithmically generated domains. We took into account the constraints of the real-world takedown context that make previous approaches inapplicable: we cannot rely on bulk patterns, detecting ongoing malware activity or actively connecting to domains. We have proposed a hybrid model that balances automation with manual classification to achieve a higher performance as well as vastly reduce investigator effort. We have developed and evaluated our approach to represent the Avalanche takedown most truthfully, such that our results and findings reflect the utility of automated domain classifiers in a real-world takedown scenario, such as for our contribution to the 2019 iteration.

Bibliography

[1] "Internet Hall of Fame Innovator: Paul Vixie," https://www.internethalloffame.org/inductees/paul-vixie.

[2] P. Foremski, O. Gasser, and G. C. M. Moura, "DNS observatory: The big picture of the DNS," in *Proceedings of the Internet Measurement Conference*. ACM, 2019, pp. 87–100.

[3] "Internet Hall of Fame Pioneer: Jon Postel," https://www.internethalloffame.org/inductees/jon-postel.

[4] "Official Biography: Paul Mockapetris," https://www.internethalloffame.org/official-biography-paul-mockapetris.

[5] P. Mockapetris, "Domain names: Concepts and facilities," Internet Requests for Comments, RFC 882, 1983. [Online]. Available: http://www.rfc-editor.org/rfc/rfc882.txt

[6] P. Mockapetris, "Domain names: Implementation specification," Internet Requests for Comments, RFC 883, 1983.

[7] P. Mockapetris, "Domain Names - Concepts and Facilities," RFC 1034, Internet Engineering Task Force, Nov. 1987.

[8] P. Mockapetris, "Rfc 1035 domain names - implementation and specification," Internet Engineering Task Force, November 1987. [Online]. Available: http://tools.ietf.org/html/rfc1035

[9] "Danny Hillis: The Internet could crash. We need a Plan B," https://www.youtube.com/watch?v=AOEQ9GteWbg.

[10] "Internet Corporation for Assigned Names and Numbers (ICANN)," https://www.icann.org.

[11] M. Korczyński, M. Wullink, S. Tajalizadehkhoob, G. Moura, A. Noroozian, D. Bagley, and C. Hesselman, "Cybercrime after the sunrise: A statistical analysis of dns abuse in new gtlds," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 2018, pp. 609–623.

[12] M. Korczyński, M. Wullink, S. Tajalizadehkhoob, G. C. M. Moura, and C. Hesselman, "Statistical Analysis of DNS Abuse in gTLDs Final Report. Technical Report," 2017. [Online]. Available: https://www.icann.org/en/system/files/files/sadag-final-09aug17-en.pdf

[13] D. Kaminsky, "It's the End of the Cache as We Know It," 2008, https://www.slideshare.net/dakami/dmk-bo2-k8.

[14] Donald E. Eastlake 3rd, "Domain Name System Security Extensions," Internet Requests for Comments, RFC 2535, March 1999.

[15] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. R. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson, "A longitudinal, end-to-end view of the DNSSEC ecosystem," in *USENIX Security Symposium*, E. Kirda and T. Ristenpart, Eds. USENIX Association, 2017, pp. 1307–1322.

[16] "https://www.sidn.nl/en/news-and-blogs/dnssec-adoption-heavily-dependent-on-incentives-and-active-promotion," https://www.sidn.nl/en/news-and-blogs/dnssec-adoption-heavily-dependent-on-incentives-and-active-promotion.

[17] M. Korczyński and A. Noroozian, "Security reputation metrics," in *Encyclopedia of Cryptography, Security and Privacy*. Springer Berlin Heidelberg, 2021. [Online]. Available: https://doi.org/10.1007/978-3-642-27739-9_1625-1

[18] "Massive DDoS Attack Hit DNS Root Servers," https://www.cs.cornell.edu/people/egs/beehive/rootattack.html.

[19] G. C. M. Moura, R. de Oliveira Schmidt, J. S. Heidemann, W. B. de Vries, M. Müller, L. Wei, and C. Hesselman, "Anycast vs. ddos: Evaluating the november 2015 root DNS event," in *Proceedings of the 2016 ACM on Internet Measurement*

Conference, P. Gill, J. S. Heidemann, J. W. Byers, and R. Govindan, Eds. ACM, 2016, pp. 255–270.

[20] M. Jonker, A. Sperotto, R. van Rijswijk-Deij, R. Sadre, and A. Pras, "Measuring the adoption of ddos protection services," in *Proceedings of the 2016 ACM on Internet Measurement Conference*, P. Gill, J. S. Heidemann, J. W. Byers, and R. Govindan, Eds. ACM, 2016, pp. 279–285.

[21] J. Klensin, "RFC 5321: Simple Mail Transfer Protocol," Internet Requests for Comments, 2015. [Online]. Available: http://tools.ietf.org/html/rfc5321

[22] S. Kitterman, "RFC 7208: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email," Internet Requests for Comments, 2014. [Online]. Available: http://tools.ietf.org/html/rfc7208

[23] E. Kucherawy, M. Zwicky, and E. Zwicky, "RFC 7489: Domain-Based Message Authentication, Reporting, and Conformance (DMARC)," Internet Requests for Comments, 2015. [Online]. Available: http://tools.ietf.org/html/rfc7489

[24] S. Maroofi, M. Korczyński, and A. Duda, "From defensive registration to subdomain protection: Evaluation of email anti-spoofing schemes for high-profile domains," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2020.

[25] S. Maroofi, M. Korczyński, A. Hölzel, and A. Duda, "Adoption of email anti-spoofing schemes: A large scale analysis," *IEEE Transactions on Network and Service Management*, 2021.

[26] V. L. Pochat, T. van Hamme, S. Maroofi, T. V. Goethem, D. Preuveneers, A. Duda, W. Joosen, and M. Korczyński, "A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints," in *Proc. NDSS*, 2020.

[27] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *USENIX Conference on Security Symposium*, 2014.

[28] D. Senie and P. Ferguson, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," RFC 2827, May 2000. [Online]. Available: https://rfc-editor.org/rfc/rfc2827.txt

[29] P. Vixie, "Rate-limiting state," *Communications of the ACM*, vol. 57, no. 4, pp. 40–43, 2014.

[30] J. Mauch, "Open Resolver Project," https://archive.nanog.org/sites/default/files/tue.lightning3.open_resolver.mauch_.pdf.

[31] "Alert (TA14-017A) UDP-Based Amplification Attacks," https://us-cert.cisa.gov/ncas/alerts/TA14-017A/.

[32] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Network and Distributed System Security Symposium*, 2014.

[33] "Reports on openly accessible services," https://www.bsi.bund.de/EN/Topics/IT-Crisis-Management/CERT-Bund/CERT-Reports/Reports/openly-accessible-services/openly-accessible-services_node.html.

[34] "Open Resolver Scanning Project," https://scan.shadowserver.org/dns/.

[35] "Center for Applied Internet Data Analysis," https://www.caida.org/.

[36] "DNS-ITR Project," https://www.caida.org/funding/dns-itr.

[37] "Day In The Life of the Internet (DITL) ," https://www.caida.org/projects/ditl/.

[38] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a Dynamic Reputation System for DNS." in *Proc. USENIX Security Symposium*, 2010, pp. 273–290.

[39] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in *Proc. NDSS*, 2011, pp. 1–17.

[40] A. Noroozian, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Developing security reputation metrics for hosting providers," in *Proc. of the 8th USENIX CSET*, 2015, pp. 1–8.

[41] F. Weimer, "Passive DNS Replication," https://www.first.org/conference/2005/papers/florian-weimer-paper-1.pdf.

[42] D. Barr, "Common dns operational and configuration errors," Internet Requests for Comments, RFC 1912, 1996.

[43] S. Tajalizadehkhoob, R. Böhme, C. Gañán, M. Korczyński, and M. van Eeten, "Rotten Apples or Bad Harvest? What We Are Measuring When We Are Measuring Abuse," *Forthcoming ACM Transactions on Internet Technology (TOIT)*, 2017.

[44] S. Tajalizadehkhoob, T. van Goethem, M. Korczyński, A. Noroozian, R. Böhme, T. Moore, W. Joosen, and M. van Eeten, "Herding Vulnerable Cats: A Statistical Approach to Disentangle Joint Responsibility for Web Security in Shared Hosting," in *ACM CCS*, 2017.

[45] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for dns over transport layer security (tls)," Internet Requests for Comments, RFC 7858, 2016.

[46] P. Hoffman and P. McManus, "Dns queries over https (doh)," Internet Requests for Comments, RFC 8484, 2018.

[47] C. Huitema, A. Mankin, and S. Dickinson, "Specification of dns over dedicated quic connections," https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-00, Tech. Rep., 2020.

[48] P. Schmitt, A. Edmundson, A. Mankin, and N. Feamster, "Oblivious DNS: practical privacy for DNS queries," *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 2, pp. 228–244, 2019.

[49] J. Brodkin, "Comcast Fights Google's Encrypted-DNS Plan But Promises Not To Spy On Users," 2019, https://arstechnica.com/tech-policy/2019/10/comcast-fights-googles-encrypted-dns-plan-but-promises-not-to-spy-on-users.

[50] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A high-performance, scalable infrastructure for large-scale active DNS measurements," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 6, pp. 1877–1888, 2016.

[51] S. Maroofi, M. Korczyński, C. Hesselman, B. Ampeau, and A. Duda, "COMAR: Classification of Compromised versus Maliciously Registered Domains," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2020.

[52] O. Hohlfeld, "Operating a DNS-based Active Internet Observatory," in *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*. ACM, 2018, pp. 60–62.

[53] O. Cetin, C. Gañán, M. Korczyński, and M. van Eeten, "Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning," in *WEIS 2017*, June 2017.

[54] K. Harrenstien, M. Stahl, and E. Feinler, "Nicname/whois," Internet Requests for Comments, RFC 954, 1985.

[55] L. Daigle, "Whois protocol specification," Internet Requests for Comments, RFC 3912, 2004.

[56] A. Newton and S. Hollenbeck, "Registration data access protocol (rdap) query format," Internet Requests for Comments, RFC 7482, 2015.

[57] ICANN, "Registration Data Access Protocol (RDAP)," https://www.icann.org/rdap, 2019.

[58] J. Bayer, Y. Nosyk, O. Hureau, S. Fernandez, I. Paulovics, A. Duda, and M. Korczyński, "Study on Domain Name System Abuse VIGIE 2020/0653," 2021.

[59] C. Gañán, "Whois sunset? a primer in registration data access protocol (rdap) performance," in *Network Traffic Measurement and Analysis Conference*, 2021.

[60] "Temporary Specification for gTLD Registration Data," https://www.icann.org/resources/pages/gtld-registration-data-specs-en.

[61] "Advisory Statement: Temporary Specification for gTLD Registration Data," https://www.icann.org/en/system/files/files/advisory-statement-gtld-registration-data-specs-17may18-en.pdf.

[62] "SSAC Advisory on Registrar Impersonation Phishing Attacks ," https://www.icann.org/en/system/files/files/sac-028-en.pdf, 2008.

[63] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists," in *2018 Internet Measurement Conference*, ser. IMC '18, 2018, pp. 478–493.

[64] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: a research-oriented top sites ranking hardened against manipulation," in *Proc. of the 26th Annual Network and Distributed System Security Symposium*. Internet Society, 2019.

[65] D. Kaminsky, "It's The End Of The Cache As We Know It," In: Black Hat Conference, http://www.slideshare.net/dakami/dmk-bo2-k8, August 2008.

[66] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," in *Proc. of ACM IMC*, 2015, pp. 355–368.

[67] D. Dagon, N. Provos, C. P. Lee, and W. Lee, "Corrupted DNS Resolution Paths: The Rise of a Malicious Resolution Authority," in *Proc. of NDSS*, 2008.

[68] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," Internet RFC 2136, April 1997.

[69] N. Biasini and J. Esler, "Threat Spotlight: Angler Lurking in the Domain Shadows," http://blogs.cisco.com, March 2015.

[70] C. Arthur, "Twitter and New York Times Still Patchy as Registrar Admits SEA Hack," https://www.theguardian.com, 2013.

[71] R. Droms, "Dynamic Host Configuration Protocol," Internet RFC 2131, March 1997.

[72] D. Eastlake 3rd, "Secure Domain Name System Dynamic Update," Internet RFC 2137, April 1997.

[73] B. Wellington, "Secure Domain Name System (DNS) Dynamic Update," Internet RFC 3007, November 2000.

[74] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)," Internet RFC 2845, May 2000.

[75] Internet Systems Consortium, Inc., "BIND – The Most Widely Used Name Server Software," https://www.isc.org/downloads/bind, November 2015.

[76] Internet Systems Consortium, Inc., "History of BIND," https://www.isc.org/history-of-bind, January 2015.

[77] Univerität Tübingen, "BIND Version 8 Online Documentation," http://astro.uni-tuebingen.de/software/bind, March 1998.

[78] P. Albitz and C. Liu, *DNS and BIND, 4th Edition*. O'Reilly Media, 2001.

[79] Microsoft TechNet, "Dynamic Update and Secure Dynamic Update," https://technet.microsoft.com/en-us/library/cc959275.aspx, Retrieved March 2016 2016.

[80] Microsoft TechNet, "Active Directory-Integrated DNS Zones," https://technet.microsoft.com/en-us/library/cc731204(v=ws.10).aspx, April 2012.

[81] Microsoft TechNet, "Dynamic Update," https://technet.microsoft.com/en-us/library/cc784052(v=ws.10).aspx, January 2005.

[82] Microsoft TechNet, "Understanding Dynamic Update," https://technet.microsoft.com/en-us/library/cc771255.aspx, Retrieved March 2016.

[83] Microsoft TechNet, "What's New in DNS Server," https://technet.microsoft.com/en-us/library/dn305898.aspx, June 2015.

[84] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead, and R. Hall, "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)," Internet RFC 3645, October 2003.

[85] NLnet Labs, "NSD: Name Server Daemon," http://www.nlnetlabs.nl/projects/nsd/, Retrieved March 2016.

[86] D. J. Bernstein, "DJBDNS," https://cr.yp.to/djbdns.html, Retrieved March 2016.

[87] R. Olofsson, "Eagle DNS," http://www.unlogic.se/projects/eagledns, Retrieved March 2016.

[88] PowerDNS, "Dynamic DNS Update (RFC2136)," https://doc.powerdns.com/md/authoritative/dnsupdate, Retrieved March 2016.

[89] D. Dittrich and E. Kenneally, "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research," U.S. Department of Homeland Security, Tech. Rep., August 2012.

[90] "Alexa Top 1,000,000 Sites," http://s3.amazonaws.com/alexa-static/top-1m.csv.zip, Retrieved March 28, 2016.

[91] "Farsight Security: DNS Database (DNSDB)," https://www.dnsdb.info.

[92] "Internet-Wide Scan Data Repository: DNS Records (ANY)," https://scans.io/study/sonar.fdns.

[93] D. Wessels, "DNS Survey: Cache Poisoners," http://dns.measurement-factory.com/surveys/poisoners.html, 2007.

[94] H. Asghari, M. J. van Eeten, and J. M. Bauer, "Economics of Fighting Botnets: Lessons From a Decade of Mitigation," *IEEE Security & Privacy*, no. 5, pp. 16–23, 2015.

[95] S. Tajalizadehkhoob, M. Korczyński, A. Noroozian, C. Gañán, and M. van Eeten, "Apples, Oranges and Hosting Providers: Heterogeneity and Security in the Hosting Market," in *Proc. of IEEE NOMS*. IEEE Press, 2016.

[96] StopBadware, "StopBadware: A Nonprofit Anti-malware Organization." https://www.stopbadware.org, 2017.

[97] "Anti-Phishing Working Group (APWG): Cross-industry Global Group Supporting Tackling the Phishing Menace," http://www.antiphishing.org.

[98] G. Aaron and R. Rasmussen, "Anti-Phishing Working Group (APWG) Global Phishing Survey: Trends and Domain Name Use in 2H2014," http://internetidentity.com/wp-content/uploads/2015/05/APWG_Global_Phishing_Report_2H_2014.pdf, May 2015.

[99] "Over a Quarter of Phishing Attacks in 2014 Targeted Users' Financial Data," http://www.kaspersky.com, February 2015.

[100] "FPDNS-DNS Fingerprinting Tool," https://www.dns-oarc.net/tools/fpdns, 2014.

[101] E. L. Kaplan and P. Meier, "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958.

[102] R. Beverly, A. Berger, Y. Hyun, and k. claffy, "Understanding the Efficacy of Deployed Internet Source Address Validation Filtering," in *Internet Measurement Conference*. ACM, 2009.

[103] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Network and Distributed System Security Symposium*, 2014.

[104] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *USENIX Conference on Security Symposium*, 2014.

[105] J. Bushart and C. Rossow, "DNS Unchained: Amplified Application-Layer DoS Attacks Against DNS Authoritatives," in *Research in Attacks, Intrusions, and Defenses*, M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis, Eds. Cham: Springer International Publishing, 2018, pp. 139–160.

[106] D. Kopp, C. Dietzel, and O. Hohlfeld, "DDoS Never Dies? An IXP Perspective on DDoS Amplification Attacks," in *Proceedings of the Passive and Active Measurement International Conference*, ser. Lecture Notes in Computer Science. Springer, 2021, pp. 284–301.

[107] M. Korczyński and Y. Nosyk, "Source address validation," in *Encyclopedia of Cryptography, Security and Privacy*. Springer Berlin Heidelberg, 2021. [Online]. Available: https://doi.org/10.1007/978-3-642-27739-9_1626-1

[108] CAIDA, "The Spoofer Project," https://www.caida.org/projects/spoofer/.

[109] M. Korczyński, M. Król, and M. van Eeten, "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates," in *Internet Measurement Conference*. ACM, 2016.

[110] X. Luo, L. Wang, Z. Xu, K. Chen, J. Yang, and T. Tian, "A Large Scale Analysis of DNS Water Torture Attack," in *Conference on Computer Science and Artificial Intelligence*, 2018.

[111] L. Shafir, Y. Afek, and A. Bremler-Barr, "NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities," in *USENIX Security Symposium*, 2020.

[112] "The Closed Resolver Project," https://closedresolver.com.

[113] M. Korczyński, Y. Nosyk, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, "Don't Forget to Lock the Front Door! Inferring the Deployment of Source Address Validation of Inbound Traffic," in *Passive and Active Measurement*. Springer International Publishing, 2020.

[114] "University of Oregon Route Views Project," http://www.routeviews.org/routeviews/.

[115] "Google IPv6," https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoptionm.

[116] J. Czyz, M. Luckie, M. Allman, and M. Bailey, "Don't Forget to Lock the Back Door! A Characterization of IPv6 Network Security Policy," in *Network and Distributed Systems Security*, 2016.

[117] T. Chown, "IPv6 Implications for Network Scanning," RFC 5157, Mar. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5157.txt

[118] M. Skwarek, M. Korczyński, W. Mazurczyk, and A. Duda, "Characterizing Vulnerability of DNS AXFR Transfers with Global-Scale Scanning," in *IEEE Security and Privacy Workshops (SPW)*, 2019.

[119] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, "Clusters in the expanse: Understanding and unbiasing ipv6 hitlists," in *Proceedings of the 2018 Internet Measurement Conference*. New York, NY, USA: ACM, 2018.

[120] J. Mauch, "Spoofing ASNs," http://seclists.org/nanog/2013/Aug/132.

[121] M. Luckie, R. Beverly, R. Koga, K. Keys, J. Kroll, and k. claffy, "Network Hygiene, Incentives, and Regulation: Deployment of Source Address Validation in the Internet," in *Computer and Communications Security Conference*. ACM, 2019.

[122] F. Baker and P. Savola, "Ingress Filtering for Multihomed Networks," RFC 3704, Mar. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3704.txt

[123] S. Kottler, "February 28th DDoS Incident Report," https://github.blog/2018-03-01-ddos-incident-report/.

[124] S. Scheffler, S. Smith, Y. Gilad, and S. Goldberg, "The Unintended Consequences of Email Spam Prevention," in *Passive and Active Measurement*. Springer International Publishing, 2018.

[125] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Hell of a handshake: Abusing TCP for reflective amplification ddos attacks," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. USENIX Association, 2014.

[126] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, "Detection, Classification, and Analysis of Inter-domain Traffic with Spoofed Source IP Addresses," in *Internet Measurement Conference*. ACM, 2017.

[127] R. Beverly and S. Bauer, "The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet," in *USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop*, Jul. 2005.

[128] Q. Lone, M. Luckie, M. Korczyński, and M. van Eeten, "Using Loops Observed in Traceroute to Infer the Ability to Spoof," in *Passive and Active Measurement Conference*. Springer International Publishing, 2017.

[129] L. F. Müller, M. J. Luckie, B. Huffaker, kc claffy, and M. P. Barcellos, "Challenges in Inferring Spoofed Traffic at IXPs," in *Conference on Emerging Networking Experiments And Technologies*. ACM, 2019.

[130] Q. Lone, M. Luckie, M. Korczyński, H. Asghari, M. Javed, and M. van Eeten, "Using Crowdsourcing Marketplaces for Network Measurements: The Case of Spoofer," in *Traffic Monitoring and Analysis Conference*, 2018.

[131] Q. Lone, M. Korczyński, C. Gañán, and M. van Eeten, "SAVing the Internet: Explaining the Adoption of Source Address Validation by Internet Service Providers," in *Workshop on the Economics of Information Security*, 2020.

[132] A. Berger, N. Weaver, R. Beverly, and L. Campbell, "Internet Nameserver IPv4 and IPv6 Address Relationships," in *Internet Measurement Conference*. ACM, 2013.

[133] L. Hendriks, R. de Oliveira Schmidt, R. van Rijswijk-Deij, and A. Pras, "On the Potential of IPv6 Open Resolvers for DDoS Attacks," in *Passive and Active Measurement*. Springer International Publishing, 2017.

[134] R. Beverly and A. Berger, "Server Siblings: Identifying Shared IPv4/IPv6 Infrastructure Via Active Fingerprinting," in *Passive and Active Measurement*. Springer International Publishing, 2015.

[135] Q. Scheitle, O. Gasser, M. Rouhi, and G. Carle, "Large-scale Classification of IPv6-IPv4 Siblings with Variable Clock Skew," in *Network Traffic Measurement and Analysis Conference*. IEEE, 2017.

[136] M. Kührer, T. Hupperich, J. Bushart, C. Rossow, and T. Holz, "Going Wild: Large-Scale Classification of Open DNS Resolvers," in *Internet Measurement Conference*. ACM, 2015.

[137] Nmap, "Well Known Port List: nmap-services," https://nmap.org/book/nmap-services.html.

[138] D. Barr, "Common DNS Operational and Configuration Errors," RFC 1912, Feb. 1996. [Online]. Available: https://rfc-editor.org/rfc/rfc1912.txt

[139] T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, G. Vigna, and A. Feldmann, "In rDNS We Trust: Revisiting a Common Data-Source's Reliability," in *Passive and Active Measurement*. Springer International Publishing, 2018.

[140] S. Woolf and D. Conrad, "Requirements for a mechanism identifying a name server instance," Internet Requests for Comments, RFC 4892, June 2007.

[141] J. Martin, J. Burbank, W. Kasch, and P. D. L. Mills, "Network Time Protocol Version 4: Protocol and Algorithms Specification," RFC 5905, 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5905.txt

[142] Nmap, "File ntp-info," https://nmap.org/nsedoc/scripts/ntp-info.html.

[143] D. J. C. Klensin and R. Gellens, "Message Submission for Mail," RFC 4409, Apr. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4409.txt

[144] P. E. Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security," RFC 3207, Feb. 2002. [Online]. Available: https://rfc-editor.org/rfc/rfc3207.txt

[145] T. Z. Project, "ZGrab 2.0 - Go Application Layer Scanner," https://github.com/zmap/zgrab2.

[146] E. Rescorla and T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5246.txt

[147] M. Korczyński and A. Duda, "Markov Chain Fingerprinting to Classify Encrypted Traffic," in *2014 IEEE Conference on Computer Communications, (INFOCOM)*. IEEE, 2014, pp. 781–789.

[148] S. Jia, M. Luckie, B. Huffaker, A. Elmokashfi, E. Aben, K. Claffy, and A. Dhamdhere, "Tracking the Deployment of IPv6: Topology, Routing and Performance," *Computer Networks*, vol. 165, no. 106947, Dec 2019.

[149] T. Krenc and A. Feldmann, "BGP Prefix Delegations: A Deep Dive," in *Internet Measurement Conference*. ACM, 2016.

[150] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *USENIX Security Symposium*, 2013.

[151] C. Shue and A. Kalafut, "Resolvers Revealed: Characterizing DNS Resolvers and their Clients," *ACM Transactions on Internet Technology*, 2013.

[152] "Mutually Agreed Norms for Routing Security," https://www.manrs.org/.

[153] A. Feldmann and J. Rexford, "IP network configuration for intradomain traffic engineering," *IEEE Network*, vol. 15, no. 5, pp. 46–57, 2001.

[154] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, G. Riley *et al.*, "AS relationships: Inference and Validation," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 29–40, 2007.

[155] A. Marder, M. Luckie, A. Dhamdhere, B. Huffaker, K. Claffy, and J. M. Smith, "Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 56–69.

[156] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, "Measuring IPv6 Adoption," in *ACM SIGCOMM Conference*, 2014, pp. 87–98.

[157] M. Nikkhah and R. Guérin, "Migrating the Internet to IPv6: An Exploration of the When and Why," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2291–2304, 2016.

[158] I. Livadariu, A. Elmokashfi, and A. Dhamdhere, "Measuring IPv6 Adoption in Africa," in *AFRICOMM Conference*, 2017, pp. 345–351.

[159] (2019) Internet Crime Report. [Online]. Available: https://www.fbi.gov/news/stories/2019-internet-crime-report-released-021120

[160] (2019) A Phishing Campaign Reports Unusual Activity on Your Microsoft Account. [Online]. Available: https://www.logitheque.com

[161] D. Crocker, T. Hansen, and M. Kucherawy, "RFC 6376: DomainKeys Identified Mail (DKIM) Signatures," Internet Requests for Comments, 2011. [Online]. Available: http://tools.ietf.org/html/rfc6376

[162] Z. Durumeric, "Fast Internet-Wide Scanning: A New Security Perspective," Ph.D. dissertation, University of Michigan, 2017.

[163] H. Hu and G. Wang, "End-to-End Measurements of Email Spoofing Attacks," in *Proc. 27th USENIX Security Symposium*, 2018, pp. 1095–1112.

[164] S. Maroofi, M. Korczyński, and A. Duda, "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2020.

[165] (2018) Dmarc overview. [Online]. Available: https://dmarc.org/overview/

[166] (2019) The Top 500 Sites on the Web. [Online]. Available: https://www.alexa.com/topsites/countries

[167] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-Wide Scanning and its Security Applications," in *Proc. 23rd USENIX Security Symposium*, 2013, pp. 605–620.

[168] I. D. Foster *et al.*, "Security by Any Other Name: On the Effectiveness of Provider Based Email Security," in *Proc. 22nd ACM CCS Conference*. ACM, 2015, pp. 450–464.

[169] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. van Eeten, "Reputation Metrics Design to Improve Intermediary Incentives for Security of TLDs," in *Proc. Euro S&P*, 2017, pp. 579–594.

[170] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich, "The Long "Taile" of Typosquatting Domain Names," in *Proc. 23rd USENIX Security Symposium*, 2014, pp. 191–206.

[171] S. Schelter and J. Kunegis, "Tracking the Trackers: A Large-Scale Analysis of Embedded Web Trackers," in *Proc. 10th International AAAI Conference on Web and Social Media*, 2016.

[172] T. Vissers, W. Joosen, and N. Nikiforakis, "Parking sensors: Analyzing and detecting parked domains." in *Proc. of NDSS*, 2015.

[173] (2019) Shavar Tracking Protection Lists. [Online]. Available: https://github.com/mozilla-services/shavar-prod-lists

[174] (2019) Python SPF Package. [Online]. Available: https://pypi.org/project/pyspf/

[175] S. Scheffler, S. Smith, Y. Gilad, and S. Goldberg, "The Unintended Consequences of Email Spam Prevention," in *Proc. PAM Conference*. Springer, 2018, pp. 158–169.

[176] O. Cetin, C. Ganan, M. Korczyński, and M. van Eeten, "Make Notifications Great Again: Learning How to Notify in the Age of Large-Scale Vulnerability Scanning," in *Workshop on the Economy of Information Security*, 2017.

[177] S. M. Wissem Soussi, Maciej Korczyński and A. Duda, "Feasibility of Large-Scale Vulnerability Notifications after GDPR," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2020.

[178] Internet Corporation for Assigned Names and Numbers. (2018, May) Temporary specification for gTLD registration data. Internet Corporation for Assigned Names and Numbers. [Online]. Available: https://www.icann.org/resources/pages/gtld-registration-data-specs-en

[179] F. Li, Z. Durumeric, J. Czyz, M. Karami, M. Bailey, D. McCoy, S. Savage, and V. Paxson, "You've Got Vulnerability: Exploring Effective Vulnerability Notifications," in *USENIX Security*, 2016.

[180] Z. Durumeric *et al.*, "Neither Snow Nor Rain Nor MITM...: An Empirical Analysis of Email Delivery Security," in *Proc. ACM IMC Conference*. ACM, 2015, pp. 27–39.

[181] H. Hu, P. Peng, and G. Wang, "Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems," in *Proc. SecDev Conference*. IEEE Computer Society, 2018, pp. 94–101.

[182] T. Halvorson, M. F. Der, I. Foster, S. Savage, L. K. Saul, and G. M. Voelker, "From .Academy to .Zone: An Analysis of the New TLD Land Rush," in *Proc. of ACM IMC*, 2015, pp. 381–394.

[183] J. Postel and J. Reynolds, "Domain requirements," Internet Requests for Comments, RFC Editor, RFC 920, October 1984.

[184] ICANN, "New gTLD Program," https://icannwiki.com/New_gTLD_Program, 2017.

[185] ICANN, ".madrid," https://icannwiki.org/.madrid, March 2015.

[186] ICANN, "New gTLD Program Explanatory Memorandum: Mitigating Malicious Conduct," https://archive.icann.org/en/topics/new-gtlds/mitigating-malicious-conduct-04oct09-en.pdf, October 2009.

[187] ICANN, "New gTLD Program Safeguards Against DNS Abuse," https://newgtlds.icann.org/en/reviews/dns-abuse/safeguards-against-dns-abuse-18jul16-en.pdf, July 2016.

[188] S. Hao, N. Feamster, and R. Pandrangi, "Monitoring the initial dns behavior of malicious domains," in *Proc. of the IMC*. ACM, 2011, pp. 269–278.

[189] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *Proc. USENIX Security*, 2014.

[190] H. Shulman and M. Waidner, "Towards security of internet naming infrastructure," in *Proc. of the ESORICS*. Springer, 2015, pp. 3–22.

[191] I. Khalil, T. Yu, and B. Guan, "Discovering malicious domains through passive dns data graph analysis," in *Proc. of the ASIACCS*. ACM, 2016, pp. 663–674.

[192] D. Chiba, T. Yagi, M. Akiyama, T. Shibahara, T. Mori, and S. Goto, "Domainprofiler: toward accurate and early discovery of domain names abused in future," *International Journal of Information Security*, pp. 1–20, 2017.

[193] S. Liu, I. Foster, S. Savage, G. M. Voelker, and L. K. Saul, "Who Is .com?: Learning to Parse WHOIS Records," in *Proc. Internet Measurement Conference*. ACM, 2015, pp. 369–380.

[194] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the Domain Registration Behavior of Spammers," in *Proceedings of the 2013 Conference on Internet Measurement Conference (IMC'13)*. ACM, 2013, pp. 63–76.

[195] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis, "Domain-z: 28 registrations later measuring the exploitation of residual trust in domains," in *Proc. of the IEEE S&P*. IEEE, 2016, pp. 691–706.

[196] T. Lauinger, K. Onarlioglu, A. Chaabane, W. Robertson, and E. Kirda, "Whois lost in translation:(mis) understanding domain name expiration and re-registration," in *Proc. of the IMC*. ACM, 2016, pp. 247–253.

[197] G. Aaron and R. Rasmussen, "Global Phishing Survey: Trends and Domain Name Use in 2016," http : / / docs.apwg.org / reports / APWG_Global_Phishing_Report_2015-2016.pdf, 2016.

[198] T. Vissers, J. Spooren, P. Agten, D. Jumpertz, P. Janssen, M. Van Wesemael, F. Piessens, W. Joosen, and L. Desmet, "Exploring the ecosystem of malicious domain registrations in the .eu tld," in *Proc. of the RAID*. Springer, 2017, pp. 472–493.

[199] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda, "FIRE: FInding Rogue nEtworks," in *Proceedings of the 2009 Annual Computer Security Applications Conference*, ser. ACSAC '09. IEEE Computer Society, 2009, pp. 231–240.

[200] S. Tajalizadehkhoob, C. Gañán, A. Noroozian, and M. van Eeten, "The Role of Hosting Providers in Fighting Command and Control Infrastructure of Financial Malware," in *Proceedings of the 12th ACM Symposium (ASIACCS)*. ACM, 2017.

[201] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. van Eeten, "Reputation metrics design to improve intermediary incentives

for security of tlds," in *2017 IEEE European Symposium on Security and Privacy (Euro SP)*, April 2017.

[202] "The Spamhaus Project," www.spamhaus.org.

[203] "SURBL - URI reputation data," http://www.surbl.org.

[204] "The Secure Domain Foundation," https://securedomain.org/.

[205] "Spam-Filter Anti-Spam Virenschutz," http://clean-mx.de.

[206] "The Domain Block List," https://www.spamhaus.org/dbl.

[207] "StopBadware: DSP," https://www.stopbadware.org/data-sharing.

[208] "SURBL Lists," http://www.surbl.org/lists.

[209] G. Aaron and R. Rasmussen, "APWG Global Phishing Survey: Trends and Domain Name Use in 2H2014," http://docs.apwg.org/reports/APWG_Global_Phishing_Report_2H_2014.pdf, 2015.

[210] WhoisXML, "Whois XML API," https://www.whoisxmlapi.com/, 2017.

[211] "DomainTools: Domain Whois Lookup, Whois API & DNS Data Research," http://www.domaintools.com.

[212] ICANN, "TLD Startup Information," https://newgtlds.icann.org/en/program-status/sunrise-claims-periods, Retrieved on February 2017.

[213] ICANN, "ICANN: .zuerich TLD," https://icannwiki.org/.zuerich, 2017.

[214] ICANN, "Monthly Registry Reports," https://www.icann.org/resources/pages/registry-reports.

[215] IANA, "IANA: Registrar IDs," https://www.iana.org/assignments/registrar-ids/registrar-ids.xhtml, 2017.

[216] "National Physical Laboratory: A Study of Whois Privacy and Proxy Service Abuse," https://gnso.icann.org/en/issues/whois/pp-abuse-study-20sep13-en.pdf.

[217] S. Hansmann, "ICANN: Notice of Termination of Accreditation Agreement," https://www.icann.org/uploads/compliance_notice/attachment/895/serad-to-hansmann-4jan17.pdf, 2017.

[218] ICANN, "Registrar Accreditation Agreement," 2013. [Online]. Available: https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en#whois-accuracy

[219] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-of-Registration," in *Proc. ACM CCS*, 2016, pp. 1568–1579.

[220] (2019) EPP Status Codes – What Do They Mean, and Why Should I Know? [Online]. Available: https://www.icann.org/resources/pages/epp-status-codes-2014-06-16-en

[221] (2016) Avalanche Network Dismantled in International Cyber Operation. [Online]. Available: https://www.europol.europa.eu/newsroom/news/'avalanche'-network-dismantled-in-international-cyber-operation

[222] A. Noroozian, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Developing Security Reputation Metrics for Hosting Providers," in *Proc. Workshop on Cyber Security Experimentation and Test (CSET)*, 2015.

[223] M. Kührer, C. Rossow, and T. Holz, "Paint It Black: Evaluating the Effectiveness of Malware Blacklists," in *Proc. RAID*. Springer, 2014, pp. 1–21.

[224] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda, "Fire: Finding Rogue Networks," in *Proc. ACSAC*. IEEE, 2009, pp. 231–240.

[225] A. Noroozian, M. Ciere, M. Korczyński, S. Tajalizadehkhoob, and M. van Eeten, "Inferring Security Performance of Providers from Noisy and Heterogenous Abuse Datasets," in *Workshop on the Economics of Information Security (WEIS)*, 2017.

[226] D. Canali, D. Balzarotti, and A. Francillon, "The role of web hosting providers in detecting compromised websites," in *Proceedings of the 22nd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2013, pp. 177–188.

[227] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. v. Eeten, "Reputation Metrics Design to Improve Intermediary Incentives for Security of TLDs," in *IEEE EuroS&P*, 2017, pp. 579–594.

[228] P. Agten, W. Joosen, F. Piessens, and N. Nikiforakis, "Seven Months' Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse," in *Proc. NDSS*, 2015.

[229] H. Liu, K. Levchenko, M. Félegyházi, C. Kreibich, G. Maier, G. M. Voelker, and S. Savage, "On the Effects of Registrar-level Intervention," in *Proc. 4th USENIX Conference on Large-scale Exploits and Emergent Threats*, ser. LEET'11, 2011, pp. 5–5.

[230] OpenPhish. [Online]. Available: https://openphish.com/

[231] PishTank: Join the Fight Against Phishing. [Online]. Available: https://www.phishtank.com/

[232] APWG: Anti-Phishing Working Group. [Online]. Available: https://apwg.org

[233] URLhaus: Sharing Malicious URLs That Are Being Used for Malware Distribution. [Online]. Available: https://urlhaus.abuse.ch/

[234] "Freenom: Registry Operator of the .TK, .ML, .GA, .CF, and .GQ ccTLDs," http://www.freenom.com.

[235] "Farsight Security," https://www.farsightsecurity.com.

[236] "Public Suffix List," https://publicsuffix.org.

[237] Wappalyzer: Identify Technology on Websites. [Online]. Available: https://www.wappalyzer.com/

[238] T. Miu, A. Hui, W. Lee, D. Luo, A. Chung, and J. Wong, "Universal ddos mitigation bypass," *Black Hat USA*, 2013.

[239] V. L. Pochat, T. van Goethem, and W. Joosen, "A Smörgåsbord of Typos: Exploring International Keyboard Layout Typosquatting," in *Proc. IEEE WTMC Security and Privacy Workshop*, 2019.

[240] W. Wang and K. Shirley, "Breaking Bad: Detecting Malicious Domains Using Word Segmentation," in *Proc. 9th Workshop on Web 2.0 Security and Privacy*, 2015.

[241] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, and N. Asokan, "Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application," *IEEE Transactions on Computers*, vol. 66, no. 10, pp. 1717–1733, 2017.

[242] C. Whittaker, B. Ryner, and M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," in *Proc. NDSS*. The Internet Society, 2010.

[243] P. Lison and V. Mavroeidis, "Neural Reputation Models Learned from Passive DNS Data," in *IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3662–3671.

[244] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A Content-Based Approach to Detecting Phishing Web Sites," in *Proc. WWW Conference*. ACM, 2007, pp. 639–648.

[245] S. Le Page, G.-V. Jourdan, G. V. Bochmann, I.-V. Onut, and J. Flood, "Domain Classifier: Compromised Machines Versus Malicious Registrations," in *International Conference on Web Engineering*. Springer, 2019, pp. 265–279.

[246] B. Anderson, S. Paul, and D. McGrew, "Deciphering Malware's Use of TLS (Without Decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, 2018.

[247] MAXMIND: Detect Online Fraud and Locate Online Visitors. [Online]. Available: https://www.maxmind.com

[248] S. Kitterman, "Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1," Internet Requests for Comments, RFC 7208, April 2014. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7208.txt

[249] L.-H. Lee, K.-C. Lee, H.-H. Chen, and Y.-H. Tseng, "Poster: Proactive Blacklist Update for Anti-Phishing," in *Proc. ACM CCS*, 2014, pp. 1448–1450.

[250] DnsTwister: The Simple and Fast Domain Name Permutation Engine. [Online]. Available: https://dnstwister.report/

[251] D. Plohmann. (2018) DGArchive. [Online]. Available: https://dgarchive.caad.fkie.fraunhofer.de

[252] M. Korczyński, M. Król, and M. van Eeten, "Zone Poisoning: The How and Where of Non-Secure DNS Dynamic Updates," in *Proc. Internet Measurement Conference*. ACM, 2016, pp. 271–278.

[253] NormShield Blog. (Retrieved: August 2019) Domain shadowing. [Online]. Available: https://www.normshield.com/domain-shadowing/

[254] Wappalyzer Signature List. [Online]. Available: https://github.com/AliasIO/Wappalyzer

[255] (2019, May) WordPress Vulnerability Statistics. [Online]. Available: https://wpvulndb.com/statistics

[256] Exploit Database. [Online]. Available: https://www.exploit-db.com/

[257] VULDB: The Community-Driven Vulnerability Database. [Online]. Available: https://vuldb.com/

[258] J. Kornblum, "Identifying Almost Identical Files Using Context Triggered Piecewise Hashing," *Digital investigation*, vol. 3, pp. 91–97, 2006.

[259] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein, "Cloak of Visibility: Detecting When Machines Browse a Different Web," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 743–758.

[260] M. Kucherawy and E. Zwicky, "Domain-based Message Authentication, Reporting, and Conformance (DMARC)," Internet Requests for Comments, RFC 7489, March 2015. [Online]. Available: https://tools.ietf.org/html/rfc7489

[261] S. Maroofi, M. Korczyński, and A. Duda, "From Defensive Registration to Subdomain Protection: Evaluation of Email Anti-Spoofing Schemes for High-Profile Domains," in *Proc. Network Traffic Measurement and Analysis Conference (TMA)*, 2020.

[262] (2019) Wayback Machine General Information. [Online]. Available: https://help.archive.org/hc/en-us/articles/360004716091-Wayback-Machine-General-Information

[263] (Retrieved: March 2020) Cognitive services pricing - bing search api. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/cognitive-services/search-api/

[264] Alexa: SEO and Competitive Analysis Software. [Online]. Available: https://www.alexa.com/

[265] MAJESTIC: Find Out Who Links to Your Website. [Online]. Available: https://majestic.com/

[266] Quantcast Ranking. [Online]. Available: https://quantcast.com

[267] Umbrella: Top 1 Million Websites. [Online]. Available: http://umbrella-static.s3-us-west-1.amazonaws.com/

[268] The Spamhaus Project. [Online]. Available: https://www.spamhaus.org/

[269] M. Aertsen, M. Korczyński, G. C. M. Moura, S. Tajalizadehkhoob, and J. van den Berg, "No Domain Left Behind: Is Let's Encrypt Democratizing Encryption?" in *Proc. ANRW*, 2017, pp. 48–54.

[270] (2019) Phishing Trends and Intelligence Report: The Growing Social Engineering Threat. [Online]. Available: https://info.phishlabs.com/2019-pti-report-evolving-threat

[271] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. John Wiley & Sons, 2019, vol. 793.

[272] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A Gentle Introduction to Imputation of Missing Values," *Journal of Clinical Epidemiology*, vol. 59, no. 10, 2006.

[273] (2020) Internet Archive: Wayback Machine. [Online]. Available: https://archive.org/web/

[274] (2018) Google Transparency Report. [Online]. Available: https://transparencyreport.google.com/https/certificates

[275] O. Gasser, B. Hof, M. Helm, M. Korczyński, R. Holz, and G. Carle, "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements," in *Proc. PAM*, 2018, pp. 173–185.

[276] (2016) Location-Based Threats: How Cybercriminals Target You Based on Where You Live. [Online]. Available: https://news.sophos.com/en-us/2016/05/03/location-based-ransomware-threat-research/

[277] COMODO Certification Authority. [Online]. Available: https://ssl.comodo.com/

[278] S. J. Russell and P. Norvig, *Artificial Intelligence: a Modern Approach, 3rd Edition*. Prentice Hall, 2009.

[279] (2019) Phishing Activity Trends Report, 1stQuarter 2019. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf

[280] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a Phisher's Mind: Understanding the Anti-Phishing Ecosystem through Phishing Kit Analysis," in *IEEE eCrime*, 2018, pp. 1–12.

[281] N. Miramirkhani, T. Barron, M. Ferdman, and N. Nikiforakis, "Panning for gold.com: Understanding the Dynamics of Domain Dropcatching," in *Proc. WWW Conference*, 2018, pp. 257–266.

[282] A. K. Jain and B. B. Gupta, "Towards Detection of Phishing Websites on Client-Side Using Machine Learning Based Approach," *Telecommunication Systems*, vol. 68, no. 4, pp. 687–700, 2018.

[283] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild," in *Proc. ACM IMC*, 2018, pp. 429–442.

[284] C. L. Tan *et al.*, "PhishWHO: Phishing Webpage Detection via Identity Keywords Extraction and Target Domain Name Finder," *Decision Support Systems*, vol. 88, pp. 18–27, 2016.

[285] (2019) Google Custom Search API. [Online]. Available: https://developers.google.com/custom-search/v1/overview

[286] (2019, August) How We Made Our DNS Stack 3x Faster. [Online]. Available: https://blog.cloudflare.com/how-we-made-our-dns-stack-3x-faster/

[287] N. Kheir, F. Tran, P. Caron, and N. Deschamps, "Mentor: Positive DNS Reputation to Skim-Off Benign Domains in Botnet C&C Blacklists," in *IFIP SEC*. Springer, 2014, pp. 1–14.

[288] R. S. Rao and A. R. Pais, "Jail-Phish: An Improved Search Engine Based Phishing Detection System," *Computers & Security*, vol. 83, pp. 246–267, 2019.

[289] I. Corona, B. Biggio, M. Contini, L. Piras, R. Corda, M. Mereu, G. Mureddu, D. Ariu, and F. Roli, "Deltaphish: Detecting Phishing Webpages in Compromised Websites," in *Proc. ESORICS*, 2017.

[290] B. W. Matthews, "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme," *Biochimica et Biophysica Acta (BBA)-Protein Structure*, vol. 405, no. 2, 1975.

[291] D. Y. Wang, S. Savage, and G. M. Voelker, "Juice: A Longitudinal Study of an SEO Botnet," in *Proc. NDSS*, 2013.

[292] P. Thomas. (2010, July) Web Application Fingerprinting and Vulnerability Inferencing. [Online]. Available: https://media.blackhat.com/bh-us-10/presentations/Thomas/BlackHat-USA-2010-Thomas-BlindElephant-WebApp-Fingerprinting-slides.pdf

[293] (2018) Save Pages in the Wayback Machine. [Online]. Available: https://help.archive.org/hc/en-us/articles/360001513491-Save-Pages-in-the-Wayback-Machine

[294] T. Van Goethem, N. Miramirkhani, W. Joosen, and N. Nikiforakis, "Purchased Fame: Exploring the Ecosystem of Private Blog Networks," in *Proc. Asia CCS*, 2019, pp. 366–378.

[295] "'Avalanche' network dismantled in international cyber operation," Europol, Dec. 2016. [Online]. Available: https://www.europol.europa.eu/newsroom/news/%E2%80%98avalanche%E2%80%99-network-dismantled-in-international-cyber-operation

[296] R. Wainwright and F. J. Cilluffo, "Responding to cybercrime at scale: Operation Avalanche - a case study," Europol; Center for Cyber and Homeland Security, The George Washington University, Issue Brief 2017-03, Mar. 2017. [Online]. Available: https://cchs.gwu.edu/sites/g/files/zaxdzs2371/f/Responding%20to%20Cybercrime%20at%20Scale%20FINAL.pdf

[297] S. Alrwais, X. Liao, X. Mi, P. Wang, X. Wang, F. Qian, R. Beyah, and D. McCoy, "Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks," in *2017 IEEE Symposium on Security and Privacy*, ser. SP '17, 2017, pp. 805–823.

[298] G. Aaron and R. Rasmussen, "Global phishing survey: Trends and domain name use in 2H2009," Anti-Phishing Working Group, APWG Industry Advisory, May 2010. [Online]. Available: https://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_2H2009.pdf

[299] "Operation Avalanche: A closer look," Eurojust, EU publication QP-01-17-801-EN-N, Apr. 2017. [Online]. Available: http://www.eurojust.europa.eu/doclibrary/Eurojust-framework/Casework/Operation%20Avalanche%20-%20A%20closer%20look%20(April%202017)/2017-04_Avalanche-Case_EN.pdf

[300] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *25th USENIX Security Symposium*, ser. USENIX Security '16, 2016, pp. 263–278.

[301] (2018, Dec.) Avalanche 1,2,3.... The Shadowserver Foundation. [Online]. Available: http://blog.shadowserver.org/news/avalanche-123/

[302] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1663–1677, Oct. 2012.

[303] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in *21st USENIX Security Symposium*, ser. USENIX Security '12, 2012, pp. 491–506.

[304] R. Sivaguru, C. Choudhary, B. Yu, V. Tymchenko, A. Nascimento, and M. De Cock, "An evaluation of DGA classifiers," in *2018 IEEE International Conference on Big Data*, ser. Big Data '18, 2018, pp. 5058–5067.

[305] T. Barabosch, A. Wichmann, F. Leder, and E. Gerhards-Padilla, "Automatic extraction of domain name generation algorithms from current malware," in *IST-111/RSY-026 Symposium on Information Assurance and Cyber Defence*. NATO Science & Technology Organization, 2012.

[306] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling, "Measuring and detecting fast-flux service networks," in *15th Annual Network and Distributed System Security Symposium*, ser. NDSS '08, 2008.

[307] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '10, 2010, pp. 48–61.

[308] D. Schwarz. (2015, Apr.) Bedep's DGA: Trading foreign exchange for malware domains. Arbor Networks. [Online]. Available: https://web.archive.org/web/20160114122355/https://asert.arbornetworks.com/bedeps-dga-trading-foreign-exchange-for-malware-domains/

[309] "Declaration of special agent Aaron O. Francis in support of application for an emergency temporary restraining order and order to show cause re preliminary injunction," in *United States of America v. "flux" a/k/a "ffhost", and "flux2" a/k/a "ffhost2"*. District Court, Western District of Pennsylvania, Nov. 2016. [Online]. Available: https://www.justice.gov/opa/page/file/915231/download

[310] M. Heinemeyer. (2018, Mar.) How malware abused sixt.com and breitling.com for covert command & control communication. Darktrace. [Online]. Available: https://www.darktrace.com/en/blog/how-malware-abused-sixt-com-and-breitling-com-for-covert-command-control-communication/

[311] (2017, Dec.) Avalanche year two, this time with Andromeda. The Shadowserver Foundation. [Online]. Available: http://blog.shadowserver.org/news/avalanche-year-two-this-time-with-andromeda/

[312] Avalanche stats by subregion. The Shadowserver Foundation. [Online]. Available: https://avalanche.shadowserver.org/stats/

[313] "Preliminary injunction," in *United States of America v. "flux" a/k/a "ffhost", and "flux2" a/k/a "ffhost2"*. District Court, Western District of Pennsylvania, Dec. 2016. [Online]. Available: https://www.justice.gov/opa/page/file/917581/download

[314] O. Cetin, C. Gañán, L. Altena, T. Kasama, D. Inoue, K. Tamiya, Y. Tie, K. Yoshioka, and M. van Eeten, "Cleaning up the internet of evil things: Real-world evidence on ISP and consumer efforts to remove Mirai," in *26th Annual Network and Distributed System Security Symposium*, ser. NDSS '19, 2019.

[315] A. Hutchings, R. Clayton, and R. Anderson, "Taking down websites to prevent crime," in *2016 APWG Symposium on Electronic Crime Research*, ser. eCrime '16, 2016.

[316] Y. T. Chua, S. Parkin, M. Edwards, D. Oliveira, S. Schiffner, G. Tyson, and A. Hutchings, "Identifying unintended harms of cybersecurity countermeasures," in *2019 APWG Symposium on Electronic Crime Research*, ser. eCrime '19, 2019.

[317] K. Kopel, "Operation seizing our sites: How the federal government is taking domain names without prior notice," *Berkeley Technology Law Journal*, vol. 28, no. 4, pp. 859–900, 2013.

[318] boker *et al.* (2018, Dec.) Domain seized. [Online]. Available: https://www.namepros.com/threads/domain-seized.1116091/

[319] S. Pal. (2019, Dec.) Sinkholed. [Online]. Available: https://susam.in/blog/sinkholed/

[320] S. Danziger, J. Levav, and L. Avnaim-Pesso, "Extraneous factors in judicial decisions," *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6889–6892, 2011.

[321] J. Tierney, "Do you suffer from decision fatigue?" Aug. 2011. [Online]. Available: https://www.nytimes.com/2011/08/21/magazine/do-you-suffer-from-decision-fatigue.html

[322] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster, "Building a dynamic reputation system for DNS," in *19th USENIX Conference on Security*, ser. USENIX Security '10, 2010, pp. 273–289.

[323] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding malicious domains using passive DNS analysis," in *18th Annual Network and Distributed System Security Symposium*, ser. NDSS '11, 2011.

[324] S. Schüppen, D. Teubert, P. Herrmann, and U. Meyer, "FANCI : Feature-based automated NXDomain classification and intelligence," in *27th USENIX Security Symposium*, ser. USENIX Security '18, 2018, pp. 1165–1181.

[325] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting Domain Generation Algorithms with Long Short-Term Memory Networks," Nov. 2016, arXiv:1611.00791.

[326] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov, and A. Mosquera, "Detecting DGA domains with recurrent neural networks and side information," in *14th International Conference on Availability, Reliability and Security*, ser. ARES '19, 2019, pp. 20:1–20:10.

[327] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the potential of proactive domain blacklisting," in *3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, ser. LEET '10, 2010.

[328] S. Hao, A. Kantchelian, B. Miller, V. Paxson, and N. Feamster, "PREDATOR: Proactive recognition and elimination of domain abuse at time-of-registration," in *2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16, 2016, pp. 1568–1579.

[329] J. Spooren, T. Vissers, P. Janssen, W. Joosen, and L. Desmet, "Premadoma: An operational solution for DNS registries to prevent malicious domain registrations," in *35th Annual Computer Security Applications Conference*, ser. ACSAC '19, 2019, pp. 557–567.

[330] N. Kheir, F. Tran, P. Caron, and N. Deschamps, "Mentor: Positive DNS reputation to skim-off benign domains in botnet C&C blacklists," in *29th IFIP Inter-*

national Information Security and Privacy Conference, ser. SEC '14, 2014, pp. 1–14.

[331] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein, "Cloak of visibility: Detecting when machines browse a different web," in *2016 IEEE Symposium on Security and Privacy*, ser. SP '16, 2016, pp. 743–758.

[332] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Computing Surveys*, vol. 51, no. 4, pp. 67:1–67:36, Jul. 2018.

[333] M. Stevanovic, J. M. Pedersen, A. D'Alconzo, S. Ruehrup, and A. Berger, "On the ground truth problem of malicious DNS traffic analysis," *Computers & Security*, vol. 55, pp. 142–158, 2015.

[334] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *17th International Symposium on Research in Attacks, Intrusions and Defenses*, ser. RAID '14, 2014, pp. 1–21.

[335] N. Petit, "Artificial intelligence and automated law enforcement: A review paper," *SSRN Electronic Journal*, 2018. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3145133

[336] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in *11th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. DIMVA '14, 2014, pp. 192–211.

[337] Internet Corporation for Assigned Names and Numbers. (2012, Feb.) How long does a registration last? Can it be renewed? [Online]. Available: https://www.icann.org/resources/pages/faqs-84-2012-02-25-en#7

[338] Freenom. (2017) Free and paid domains. [Online]. Available: https://www.freenom.com/en/freeandpaiddomains.html

[339] M. Korczyński, S. Tajalizadehkhoob, A. Noroozian, M. Wullink, C. Hesselman, and M. van Eeten, "Reputation metrics design to improve intermediary incen-

tives for security of TLDs," in *2017 IEEE European Symposium on Security and Privacy*, ser. EuroS&P '17, 2017, pp. 579–594.

[340] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, 2009, pp. 1245–1254.

[341] T. Vissers, J. Spooren, P. Agten, D. Jumpertz, P. Janssen, M. V. Wesemael, F. Piessens, W. Joosen, and L. Desmet, "Exploring the ecosystem of malicious domain registrations in the .eu TLD," in *Proceedings of the 20th International Symposium on Research in Attacks, Intrusions, and Defenses*, ser. RAID '17, 2017, pp. 472–493.

[342] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A high-performance, scalable infrastructure for large-scale active DNS measurements," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 6, pp. 1877–1888, June 2016.

[343] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe, "Enabling network security through active DNS datasets," in *Research in Attacks, Intrusions, and Defenses*, ser. RAID '16, 2016, pp. 188–208.

[344] A. Sperotto, O. van der Toorn, and R. van Rijswijk-Deij, "TIDE: Threat identification using active DNS measurements," in *Proceedings of the SIGCOMM Posters and Demos*, ser. SIGCOMM Posters and Demos '17, 2017, pp. 65–67.

[345] M. Z. Rafique, T. Van Goethem, W. Joosen, C. Huygens, and N. Nikiforakis, "It's free for a reason: Exploring the ecosystem of free live streaming services," in *23rd Annual Network and Distributed System Security Symposium*, ser. NDSS '16, 2016.

[346] B. Z. H. Zhao, M. Ikram, H. J. Asghar, M. A. Kaafar, A. Chaabane, and K. Thilakarathna, "A decade of mal-activity reporting: A retrospective analysis of Internet malicious activity blacklists," in *14th ACM Asia Conference on Computer and Communications Security*, ser. ASIACCS '19, 2019, pp. 193–205.

[347] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by Internet-wide scanning," in *22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '15, 2015, pp. 542–553.

[348] Rapid7. Project sonar. [Online]. Available: https://www.rapid7.com/research/project-sonar/

[349] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet," in *26th USENIX Security Symposium*, ser. USENIX Security '17, 2017, pp. 1093–1110.

[350] B. Laurie, A. Langley, and E. Kasper, "Certificate Transparency," Internet Requests for Comments, RFC Editor, RFC 6962, June 2013.

[351] S. Sinha, M. Bailey, and F. Jahanian, "Shades of grey: On the effectiveness of reputation-based "blacklists"," in *3rd International Conference on Malicious and Unwanted Software*, ser. MALWARE '08, 2008, pp. 57–64.

[352] D. Gomes, J. Miranda, and M. Costa, "A survey on web archiving initiatives," in *International Conference on Theory and Practice of Digital Libraries*, ser. TPDL '11, 2011, pp. 408–420.

[353] K. Soska and N. Christin, "Automatically detecting vulnerable websites before they turn malicious," in *23rd USENIX Security Symposium*, ser. USENIX Security '14, 2014, pp. 625–640.

[354] E. Alowaisheq, P. Wang, S. Alrwais, X. Liao, X. Wang, T. Alowaisheq, X. Mi, S. Tang, and B. Liu, "Cracking the wall of confinement: Understanding and analyzing malicious domain take-downs," in *26th Annual Network and Distributed System Security Symposium*, ser. NDSS '19, 2019.

[355] IBM Security. IBM X-Force Exchange. frequently asked questions. [Online]. Available: https://exchange.xforce.ibmcloud.com/faq

[356] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck, "Understanding the domain registration behavior of spammers," in *2013 Internet Measurement Conference*, ser. IMC '13, 2013, pp. 63–76.

[357] W. Xu, K. Sanders, and Y. Zhang, "We know it before you do: Predicting malicious domains," in *Virus Bulletin Conference*, Sep. 2014, pp. 73–77.

[358] C. Lever, R. Walls, Y. Nadji, D. Dagon, P. McDaniel, and M. Antonakakis, "Domain-Z: 28 registrations later measuring the exploitation of residual trust in domains," in *2016 IEEE Symposium on Security and Privacy*, ser. SP '16, 2016, pp. 691–706.

[359] P. Lison and V. Mavroeidis, "Neural reputation models learned from passive DNS data," in *2017 IEEE International Conference on Big Data*, ser. Big Data '17, 2017, pp. 3662–3671.

[360] T. Frosch, M. Kührer, and T. Holz, "Predentifier: Detecting botnet C&C domains from passive DNS data," in *Advances in IT Early Warning*, M. Zeilinger, P. Schoo, and E. Hermann, Eds. Fraunhofer Verlag, Feb. 2013, pp. 78–90. [Online]. Available: http://publica.fraunhofer.de/documents/N-227985.html

[361] J. Spooren, D. Preuveneers, L. Desmet, P. Janssen, and W. Joosen, "Detection of algorithmically generated domain names used by botnets: A dual arms race," in *34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19, 2019, pp. 1916–1923.

[362] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel, "Exposure: A passive DNS analysis service to detect and report malicious domains," *ACM Transactions on Information and System Security*, vol. 16, no. 4, pp. 14:1–14:28, Apr. 2014.

[363] B. Morton. (2016, Oct.) Protect your domain with CT search. [Online]. Available: https://www.entrustdatacard.com/blog/2016/october/protect-your-domain-with-ct-search

[364] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey, and J. A. Halderman, "Towards a complete view of the certificate ecosystem," in *2016 Internet Measurement Conference*, ser. IMC '16, 2016, pp. 543–549.

[365] Q. Scheitle, O. Gasser, T. Nolte, J. Amann, L. Brent, G. Carle, R. Holz, T. C. Schmidt, and M. Wählisch, "The rise of certificate transparency and its implications on the Internet ecosystem," in *2018 Internet Measurement Conference*, ser. IMC '18, 2018, pp. 343–349.

[366] M. Korczyński, M. Wullink, S. Tajalizadehkhoob, G. C. M. Moura, A. Noroozian, D. Bagley, and C. Hesselman, "Cybercrime after the sunrise: A statistical analysis of DNS abuse in new gTLDs," in *13th ACM Asia Conference on Computer and Communications Security*, ser. ASIACCS '18, 2018, pp. 609–623.

[367] R. Clayton and T. Mansfield, "A study of Whois privacy and proxy service abuse," in *13th Annual Workshop on the Economics of Information Security*, ser. WEIS '14, 2014.

[368] L. B. Metcalf, D. Ruef, and J. M. Spring, "Open-source measurement of fast-flux networks while considering domain-name parking," in *2017 Learning from Authoritative Security Experiment Results Workshop*, ser. LASER '17, 2017, pp. 13–24.

[369] (2013, Sep.) Wayback Machine APIs. The Internet Archive. [Online]. Available: https://archive.org/help/wayback_api.php

[370] Common Crawl Foundation. Common Crawl. [Online]. Available: https://commoncrawl.org/

[371] abuse.ch. (2019) SinkDB. [Online]. Available: https://sinkdb.abuse.ch/

[372] M. Stampar *et al.* (2019) maltrail: Malicious traffic detection system. [Online]. Available: https://github.com/stamparm/maltrail

[373] M. Stampar. (2018, Oct.) Email addresses used in WHOIS registrations of sinkholed malicious/malware domains. [Online]. Available: https://gist.github.com/stamparm/9726d93fd0048aee6c54ec88a8e85bfc

[374] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[375] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, no. 1, pp. 69–101, Apr. 1996.

[376] M. Aertsen, M. Korczyński, G. C. M. Moura, S. Tajalizadehkhoob, and J. van den Berg, "No domain left behind: Is Let's Encrypt democratizing encryption?" in *2017 Applied Networking Research Workshop*, ser. ANRW '17, 2017, pp. 48–54.

[377] L. Daigle, "WHOIS protocol specification," Internet Requests for Comments, RFC Editor, RFC 3912, Sep. 2004.

[378] S. Liu, I. Foster, S. Savage, G. M. Voelker, and L. K. Saul, "Who is .com?: Learning to parse WHOIS records," in *2015 Internet Measurement Conference*, ser. IMC '15, 2015, pp. 369–380.

[379] S. Rodota, "Opinion 2/2003 on the application of the data protection principles to the Whois directories," Article 29 Data Protection Working Party, Jun. 2003. [Online]. Available: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2003/wp76_en.pdf

[380] DENIC. (2018, May) DENIC putting extensive changes into force for .DE Whois lookup service by 25 May 2018. [Online]. Available: https://www.denic.de/en/whats-new/press-releases/article/denic-putting-extensive-changes-into-force-for-de-whois-lookup-service-as-of-25-may-2018/

[381] D. Piscitello. (2018, Oct.) ICANN GDPR and WHOIS users survey. a joint survey by the anti-phishing working group (APWG) and the messaging, malware and mobile anti-abuse working group (M^3AAWG). [Online]. Available: https://www.m3aawg.org/sites/default/files/m3aawg-apwg-whois-user-survey-report-2018-10.pdf

[382] A. J. Ferrante, "The impact of GDPR on WHOIS: Implications for businesses facing cybercrime," *Cyber Security: A Peer-Reviewed Journal*, vol. 2, no. 2, pp. 143–148, 2018.

[383] L. Machlica, K. Bartos, and M. Sofka, "Learning detectors of malicious web requests for intrusion detection in network traffic," Feb. 2017, arXiv:1702.02530.

[384] E. Kidmose, E. Lansing, S. Brandbyge, and J. M. Pedersen, "Detection of malicious and abusive domain names," in *2018 1st International Conference on Data Intelligence and Security*, ser. ICDIS '18, Apr. 2018, pp. 49–56.

[385] S. Yadav and A. L. N. Reddy, "Winning with DNS failures: Strategies for faster botnet detection," in *7th International ICST Conference on Security and Privacy in Communication Networks*, ser. SecureComm '11, 2011, pp. 446–459.

[386] S. Krishnan, T. Taylor, F. Monrose, and J. McHugh, "Crossing the threshold: Detecting network malfeasance via sequential hypothesis testing," in *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, ser. DSN '13, 2013.

[387] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, 2014, pp. 395–400.

[388] J. Abbink and C. Doerr, "Popularity-based detection of domain generation algorithms," in *12th International Conference on Availability, Reliability and Security*, ser. ARES '17, 2017, pp. 79:1–79:8.

[389] M. Pereira, S. Coleman, B. Yu, M. De Cock, and A. C. A. Nascimento, "Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic," in *21st International Symposium on Research in Attacks, Intrusions, and Defenses*, ser. RAID '18, 2018, pp. 295–314.

[390] Y. Nadji, M. Antonakakis, R. Perdisci, D. Dagon, and W. Lee, "Beheading hydras: Performing effective botnet takedowns," in *2013 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '13, 2013, pp. 121–132.

[391] H. Asghari, M. Ciere, and M. J. van Eeten, "Post-mortem of a zombie: Conficker cleanup after six years," in *24th USENIX Security Symposium*, ser. USENIX Security '15, 2015, pp. 1–16.

[392] R. Shirazi, "Botnet takedown initiatives: A taxonomy and performance model," *Technology Innovation Management Review*, vol. 5, no. 1, pp. 15–20, Jan. 2015.