

Bridging the Gap: A Survey and Classification of Research-informed Ethical Hacking Tools (Supplementary Material)

Paolo Modesti, Lewis Golightly, Louis Holmes, Chidimma Opara and Marco Moscini

Teesside University, Middlesbrough, United Kingdom

Abstract

This document outlines the research-informed Ethical Hacking tools featured in the survey paper *Bridging the Gap: A Survey and Classification of Research-informed Ethical Hacking Tools* [1], published in the *Journal of Cybersecurity and Privacy*, July 2024. The tools are presented in alphabetical order according to their names.

1. Review of Tools

1.1. *Reducing bias in modelling real-world password strength via deep learning and dynamic dictionaries*

ADaMs (2021, Pasquini et al. [2]) is an advanced tool aimed at improving the precision of modelling real-world password strengths. This tool leverages deep neural networks to simulate the methodology used by adversaries when constructing attack configurations. The efficiency of ADaMs was demonstrated through case studies involving MyHeritage and Youku using the rules-set generated and RockYou as the external dictionary. For MyHeritage, ADaMs' attack matches the precision of the optimal dictionary approach and achieves a comparable number of successful guesses. Similarly, for Youku, ADaMs outperforms the optimal dictionary in terms of guessing speed, achieving faster results within the initial (10^{11}) guesses, demonstrating its effectiveness in real-world scenarios.

1.2. *AIBugHunter: A Practical tool for predicting, classifying and repairing software vulnerabilities*

AIBugHunter (2023, Fu et al. [3]) is a machine learning-based approach to detecting and mitigating software security bugs within C and C++. This package is integrated into Visual Studio Code to better bridge the gap between real-world developers and academic contributions to vulnerability detection frameworks. The tool was around 10% to 141% more effective than the compared baseline models at predicting CWE-ID within evaluated code samples.

1.3. *ARMORY: An automatic security testing tool for buffer overflow defect detection*

ARMORY (2013, Chen et al. [4]) is a kernel-based tool for detecting zero-day Buffer Overflow vulnerabilities to prevent the overflowing of sensitive data structures by testing the system and recording the results in a dump file. The components of the software include a coordinator, a checker, and two message storages.

1.4. *Autosploit: A Fully Automated Framework for Evaluating the Exploitability of Security Vulnerabilities*

Autosploit (2020, Moscovich et al. [5]) is a framework for virtualising and evaluating the components within a system that are required for exploitation via a given vector. Although in the paper the authors outline several factors that can affect exploitation, but the simulation agent only supports four actions: stopping services, deleting packages, changing system firewall rules, and modifying file permissions. The tool was tested against a Metasploitable 2 machine, which is designed to be intentionally vulnerable against a set number of attacks, and the attacker agent only supported Metasploit-based attacks, once again limiting what exploits can and cannot be assessed to those supported with a Metasploit module.

1.5. AVAIN – A Framework for Automated Vulnerability Indication for the IoT in IP-based Networks

AVAIN (2019, Egert et al. [6]) is a framework for Automated Vulnerability Indication in IP-based Internet of Things (IoT) networks. It aims at enhancing security practices of IoT system administrators by automatically deploying existing tools to generate results, presented according to the Common Vulnerability Scoring System (CVSS). The framework’s modular design enables the user to integrate arbitrary vulnerability scanners and analysis methods. AVAIN includes four components: the *Controller* processes user instructions to orchestrate the tasks. The *Module Updater* keeps modules up to date, the *Scanner* performs vulnerability scanning, and the *Analyser* processes the collected data.

1.6. Bbuzz: A bit-aware fuzzing framework for network protocol systematic reverse engineering and analysis

Bbuzz (2017, Blumbergs et al. [7]) is an open-source bit-aware network protocol fuzzing framework designed for systematic reverse engineering and analysis of network protocols. Operating at Layer-2, Bbuzz facilitates rapid protocol assessment, automatic test case creation, and user-friendly fuzzing. In a proprietary NATO Link-1 protocol case study, researchers used Bbuzz to efficiently reverse engineer the protocol, revealing critical components within a single day.

1.7. Black Ostrich: Web Application Scanning with String Solvers

Black Ostrich (2023, Eriksson et al. [8]) is a tool for crawling web applications with a content-aware approach. The tool is able to differentiate expected input formats such as email addresses, usernames, etc., using regex interpretation to discover more pages on web applications than other crawling approaches. This tool was shown to be effective when compared against existing web application crawlers such as ZAP, Enemy, and Arachni. This tool was shown to achieve 99% coverage, achieving high effectiveness in crawling capability. The tool also showed a 52% improvement over existing methods when discovering vulnerable patterns within those same web pages.

1.8. Black Widow: Blackbox Data-driven Web Scanning

Black Widow (2021, Eriksson et al. [9]) uses a black box data-driven approach for deep crawling and scanning of modern web apps. It focuses on three core pillars: navigation modelling, traversing, and tracking inter-state dependencies. Black Widow demonstrates significant code coverage improvements compared to other crawlers. Moreover, it excels in vulnerability scanning, detecting more cross-site scripting vulnerabilities without false positives, identifying missed vulnerabilities in older applications, and uncovering new vulnerabilities in production software like HotCRP, osCommerce, PrestaShop, and WordPress.

1.9. Bleem: Packet Sequence Oriented Fuzzing for Protocol Implementations

Bleem (2023, Luo et al. [10]) is a novel black-box fuzzer designed to enhance the vulnerability detection of protocol implementations through packet-sequence-oriented fuzzing. It features a noninvasive feedback mechanism that examines system outputs (packet sequences) to deduce the internal state transitions within the protocol implementation. This feedback guides the fuzzing process, including generating packet sequences that align with the protocol’s logic. Bleem significantly outperforms state-of-the-art protocol fuzzers in terms of branch coverage and vulnerability detection. It achieves up to a 175% increase in branch coverage within 24 hours compared to tools like Peach. Furthermore, Bleem discovered 15 security-critical vulnerabilities across prominent protocol implementations, resulting in 10 CVEs being attributed, showcasing its effectiveness in identifying previously undetected vulnerabilities.

1.10. Contextualisation of Data Flow Diagrams for Security Analysis

Cairis (2020, Faily et al. [11]) is a novel tool for identifying tainted data flows through the contextualisation of Data Flow Diagrams (DFDs) with other models related to usability and requirements. This adaptation of taint analysis, traditionally employed in code analysis to detect insecure data handling, is applied to design-level analysis. The Cairis tool identifies potential taint sources from human interactions or system processes, as represented in DFDs. The practical viability and the possibility of incorporating it into this tool’s current security analysis workflows are demonstrated through its implementation in an open-source software platform.

1.11. A Search Engine Backed by Internet-Wide Scanning

Censys (2015, Durumeric et al. [12]) is a vulnerability scanning tool that leverages data from continuous Internet-wide scans. Censys uses existing tools, specifically ZMap [13] and ZGrab, for hosting discovery scans across the IPv4 address spaces and application-layer handshakes. Case studies demonstrate Censys’s application in analyzing Industrial Control Systems, vulnerabilities like Heartbleed and SSLv3, and institutional attack surfaces. The application of the tool shows Censys’s ability to identify vulnerable devices and networks quickly, generate statistical reports on usage patterns, and provide insights into the security posture of devices across the internet.

1.12. Chainsaw: Chained Automated Workflow-based Exploit Generation

Chainsaw (2016, Alhuzali et al. [14]) offers a solution for automated exploit generation in web applications, surpassing existing methods in identifying and exploiting web injection vulnerabilities. It adeptly handles challenges posed by diverse web app structures, user input, and database back-ends by constructing precise models of application workflows, schemes, and native functions. Evaluated across 9 open-source applications, Chainsaw produced over 199 high-quality first- and second-order injection exploits, showcasing its superiority over comparable approaches in exploit generation for web vulnerabilities.

1.13. Chucky: exposing missing checks in source code for vulnerability discovery

Chucky (2013, Yamaguchi et al. [15]) is a tool aimed at expediting the auditing of software by exposing missing checks in source code, particularly focusing on input validation. By using static tainting techniques, Chucky identifies anomalies and omitted security-critical conditions, effectively pinpointing 12 previously undiscovered vulnerabilities in Pidgin and LibTIFF among five popular open-source projects. This method enhances the process of discovering security flaws by highlighting crucial areas of missing input validation checks.

1.14. Commix: automating evaluation and exploitation of command injection vulnerabilities in Web applications

Commix (2019, Stasinopoulos et al. [16]), short for COMMand Injection eXploiter, is an open-source tool aimed at automating the detection and exploitation of command injection flaws in web applications. Commix’s methodology encompasses attack vector generation, a vulnerability detection module, and an exploitation module. The vulnerability detection module utilises the generated attack vectors to probe for potential command injection vulnerabilities within target web applications by injecting and attempting command execution. Subsequently, the exploitation module leverages any confirmed vulnerabilities, exploiting them with the successful attack vector to facilitate the execution of arbitrary commands by the attacker. Commix underwent threefold experimentation—in a virtual lab, against other tools, and within real-world applications—demonstrating its efficacy in identifying and exploiting command injection vulnerabilities across these scenarios, thereby demonstrating its significant value in cybersecurity practices.

1.15. CryptoGuard: High Precision Detection of Cryptographic Vulnerabilities in Massive-sized Java Projects

CryptoGuard (2019, Rahaman et al. [17]) is a tool with efficient slicing algorithms in Java programs that reduce false alerts by 76% to 80% in cryptographic API misuse threats. Running on large-scale projects like Apache and Android apps, CRYPTO GUARD provided security insights and aided projects like Spark, Ranger, and Ofbiz in code hardening. The tool achieved 98.61% precision by manually confirming 1,277 true positives out of 1,295 Apache alerts. Additionally, it established a benchmark with basic and advanced cases, extensively comparing with CrySL, SpotBugs, and Coverity.

1.16. Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings

CuPerFuzzer (2021, Li et al. [18]) is an automatic fuzzing tool designed to detect vulnerabilities within the Android OS related to custom permissions. Through extensive testing, it uncovered 2,384 effective cases and identified 30 critical paths, exposing severe design shortcomings within the Android permission framework. These issues include dangling custom permissions, inconsistent permission group mapping, permission elevating, and inconsistent permission definition, potentially allowing malicious apps to gain unauthorised system permissions.

1.17. Deemon: Detecting CSRF with Dynamic Analysis and Property Graphs

Deemon (2017, Pellegrino et al. [19]) introduces an automated security testing framework for CSRF vulnerability detection. Utilising a novel modelling paradigm capturing web app aspects in a unified property graph, it autonomously constructs models from dynamic traces and identifies potentially vulnerable operations via graph traversals. Deemon’s validation approach conducts security tests, and 14 previously unknown CSRF vulnerabilities across 10 open-source web applications have been discovered, showing its efficacy in detecting exploitable issues threatening user accounts and entire websites.

1.18. Delta: A security assessment framework for software-defined networks.

Delta (2017, Lee et al. [20]) is a novel vulnerability analysis tool that focuses specifically on Software Defined Networking (SDN) and utilizes known published attacks. The tool uses a fuzzing module that automatically detects zero-day vulnerabilities and has been designed specifically for OpenFlow controller platforms.

1.19. DFBC Recon Tool: Digital Footprint and Breach Check Reconnaissance Tool

DFBC (2021, Yusof et al. [21]) is a reconnaissance tool that provides a Digital Footprint and Breach Check. This software can extract user information that is publicly available and check the breach activity status for accounts with high speed. The tool has both a CLI and GUI and focuses on gathering data on social networks such as Facebook and X/Twitter, as well as email breaches.

1.20. Diane: Identifying fuzzing triggers in apps to generate under-constrained inputs for iot devices

Diane (2021, Redini et al. [22]) is a tool for vulnerability analysis in IoT targeting Android applications used to control IoT devices, at the network level and UI level by detecting fuzzing triggers. The software has been tested and has successfully identified nine zero-day vulnerabilities. This tool can be applied to discover vulnerabilities in Smart Home appliances such as door Smart Locks. Given its reliance on dynamic analysis, Diane lacks the capability to detect fuzzing triggers that are not executed by the application. Additionally, it is unable to perform fuzzing on nested Java objects.

1.21. Finding Security Vulnerabilities in IoT Cryptographic Protocol and Concurrent Implementations

EBF (2021, Aljaafari et al. [23]) is a tool used for the discovery of vulnerabilities within IoT devices. It uses static and dynamic analysis to discover issues surrounding memory safety, race conditions, thread leak and arithmetic overflow. The tool simulates a server and a client in order to detect such vulnerabilities within the implemented protocol. The tool demonstrated its effectiveness by detecting a previously unknown race condition bug within WOLFMQTT in only 15 minutes with 22 MB memory consumption. The tool was also verified against pre-existing vulnerabilities within OpenSSL and successfully identified the issues as expected.

1.22. ELAID: detecting integer-Overflow-to-Buffer-Overflow vulnerabilities by light-weight and accurate static analysis

ELAID (2020, Xu et al. [24]) (Enhanced Lightweight and Accurate method of static IO2BO vulnerability Detection) is a tool designed for detecting Integer-Overflow-to-Buffer-Overflow (IO2BO) vulnerabilities. Built on LLVM [25], ELAID’s indirect call analysis aims to eliminate false positives. It has been tested on NIST’s SAMATE Juliet 1.2 suite and other real-world applications, demonstrating the ability to detect 152 vulnerabilities without false positives. However, the authors acknowledge the need for a combined symbolic execution and fuzzing approach to enhance practicality.

1.23. ESASCF: Expertise Extraction, Generalization and Reply Framework for Optimized Automation of Network Security Compliance

ESASCF (2023, Ghanem et al. [26]) is a tool designed to address the resource-intensity and repetitiveness of the network security auditing process by automating the extraction, processing, storage, and reuse of expertise in similar scenarios or during periodic re-testing. ESASCF leverages industrial and open-source vulnerability assessment and penetration testing tools to automate the security compliance (SC) process. It is designed to autonomously handle SC re-testing, offloading human experts from repetitive SC segments, allowing them to focus on more critical tasks in ad-hoc compliance tests. The framework was tested on networks of different sizes, demonstrating time efficiency and testing effectiveness improvements. Specifically, ESASCF significantly reduces the time required for an expert to complete the first security compliance of typical corporate networks by 50% and 20% in re-testing scenarios.

1.24. ESRFuzzer: an enhanced fuzzing framework for physical SOHO router devices to discover multi-Type vulnerabilities

ESRFuzzer (2021, Zhang et al. [27]) (Enhanced SOHO Router Fuzzing Framework) is as a vulnerability discovery tool tailored for small office and home office (SOHO) routers. It operates through an automated FWSR fuzzing framework, incorporating KEY-VALUE and CONF-READ semantic models with power management for testing environment recovery. Equipped with diverse mutation rules and monitoring mechanisms, the tool efficiently identifies various vulnerability types. ESRFuzzer excels in discovering CONF and READ operation issues, particularly in general and D-CONF modes. Testing on 10 routers revealed 136 issues, with 120 of them confirmed as 0-day vulnerabilities.

ESRFuzzer (2021, Zhang et al. [27]) (Enhanced SOHO Router Fuzzing Framework) is a vulnerability discovery tool for small office and home office (SOHO) routers, employing an automated FWSR fuzzing framework. It uses KEY-VALUE and CONF-READ semantic models with power management for testing environment recovery. With diverse mutation rules and monitoring mechanisms, it identifies various vulnerability types. In general and D-CONF modes, ESRFuzzer excels in discovering CONF and READ operation issues. Testing on 10 routers unveiled 136 issues, 120 confirmed as 0-day vulnerabilities.

1.25. ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems

ESSecA (2022, Rak et al. [28]) is an autonomous system for Threat Modelling. It employs an algorithm that correlates attacks to threats in the penetration testing of IoT systems using a threat catalogue as a query table. This approach involves mapping threats to attacks and filtering out those that are not relevant. The tool is designed with a modular architecture and produces two main outputs: a threat model and a testing plan table. It utilises databases, such as Cyber Threat Intelligence (CTI), to enhance its functionality.

1.26. Firmaster: Analysis Tool for Home Router Firmware

Firmaster (2018, Visoottiviset et al. [29]) is a tool designed to analyse home router firmware source code vulnerabilities. By emulating router firmware, Firmaster identifies, evaluates, and simulates potential vulnerabilities. Its functions include Password Cracking, SSL Scanning, and Web Static Analysis, addressing the OWASP Top 10 2014 IoT vulnerabilities. The tool validates the uploaded firmware, attempts to crack root passwords, and verifies secure connections. Furthermore, it conducts static and dynamic web analysis to identify PHP source code vulnerabilities.

1.27. FUGIO: Automatic Exploit Generation for PHP Object Injection Vulnerabilities

FUGIO (2022, Park et al. [30]) is a tool addressing PHP Object Injection (POI) vulnerabilities, enabling automatic exploit generation. It uses static and dynamic analyses to create gadget chains, serving as exploit blueprints. By running fuzzing campaigns, FUGIO successfully generated exploit objects, producing 68 exploits from 30 vulnerable applications without false positives. Additionally, it uncovered two previously unreported POI vulnerabilities and created five functional exploits, showing its effectiveness in automatic exploit creation for POI vulnerabilities.

1.28. FUSE: Finding File Upload Bugs via Penetration Testing

FUSE (2020, Lee et al. [31]) is a penetration testing tool specifically created to uncover Unrestricted File Upload (UFU) and Unrestricted Executable File Upload (UEFU) vulnerabilities in PHP-based server-side web applications. It effectively generates exploit payloads via upload requests, overcoming content-filtering checks while preserving file execution semantics. FUSE identified 30 previously unreported UEFU vulnerabilities, including 15 CVEs, across 33 real-world web applications.

1.29. GAIL-PT: An intelligent penetration testing framework with generative adversarial imitation learning

Gail-PT (2023, Chen et al. [32]) as a framework to automate penetration testing. The tool provides advice to the penetration testers for enhanced decision making reducing the reliance on manual testing whilst utilising Generative Adversarial Imitation Learning (GAIL) based on Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) that innovates the testing process. The tool has been observed to achieve state-of-the-art results when applied to the Metasploitable2 penetration testing target VM.

1.30. GNPassGAN: Improved Generative Adversarial Networks For Trawling Offline Password Guessing

GNPassGAN (2022, Yu et al. [33]) is designed to enhance offline password guessing by leveraging generative adversarial networks (GANs) to generate password guesses by training on datasets obtained from real-world breaches. The tool is benchmarked against PassGAN and other password-guessing methods, using datasets like Rockyou and phpbb for training and evaluation. GNPassGAN demonstrates a significant improvement over PassGAN, achieving 88.03% more accurate password guesses and producing 31.69% fewer duplicates when generating a large number of guesses (10^8).

1.31. HARMer: Cyber-Attacks Automation and Evaluation

HARMer (2020, Enoch et al. [34]) is an automation framework for cyber-attack generation, that addresses limitations in manual attack execution by red teams. Using the Hierarchical Attack Representation Model (HARM), it outlines requirements, key phases, and security metrics-based attack planning strategies. Through experiments conducted in both enterprise networks and Amazon Web Services, HARMer demonstrates effective modelling of attackers' operations. This framework offers automated assessment capabilities, enabling security administrators to evaluate diverse threats and attacks systematically, facilitating a more efficient defence against cyber threats in networked systems.

1.32. HILTI: an Abstract Execution Environment for Deep, Stateful Network Traffic Analysis

HILTI (2014, Sommer et al. [35]) is a tool that employs an abstract machine model specifically designed for deep stateful network traffic analysis. It addresses the significant gap between the high-level conceptualisation of network analysis tasks (such as pattern searching in HTTP requests) and the intricate low-level implementation details required to execute these tasks efficiently and securely. The system facilitates the development of network traffic analysis applications by providing built-in support for common data types, state management, concurrency models, and a secure memory model. This foundation enables developers to create robust applications capable of handling the vast and varied data flows in network traffic, all while operating within real-time performance constraints.

1.33. IoTfuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing

IoTfuzzer (2018, Chen et al. [36]) is a vulnerability scanning tool designed to detect memory corruption vulnerabilities in IoT devices. It utilises an app-based fuzzing framework that operates without the need for access to the devices' firmware images. The process involves UI analysis to identify network event triggers and data-flow analysis to monitor the movement of fields related to the application protocol. By altering the original fields, it generates fuzzing messages. The effectiveness of IoTfuzzer is demonstrated through experiments, where it successfully uncovered 15 serious vulnerabilities in 9 out of 17 evaluated IoT devices. These vulnerabilities range from stack-based and heap-based buffer overflows to null pointer de-references and unidentified crashes, highlighting the framework's capability to reveal critical security flaws.

1.34. JCOMIX: a search-based tool to detect XML injection vulnerabilities in web applications

JCOMIX (2019, Stallenberg et al. [37]) is a tool developed in Java that is designed to generate attacks (test objectives) to identify XML injection threats in web applications. It assesses information sanitation and validation in micro-service frameworks, aiding in the identification of vulnerabilities.

1.35. A Light-Weight and Accurate Method of Static Integer-Overflow-to-Buffer-Overflow Vulnerability Detection

LAID (2018, Hu et al. [38]) is framework designed to accurately identify potential Integer-Overflow-to-Buffer-Overflow (IO2BO) vulnerabilities in software. The framework combines inter-procedural dataflow analysis and taint analysis to identify potential vulnerabilities and employs a lightweight method for further filtering out false positives. The framework’s effectiveness was evaluated using NIST’s SAMATE Juliet test suite and six known IO2BO vulnerabilities in real-world applications. The framework effectively detected all known IO2BO vulnerabilities in the test suite without any false positives.

1.36. Link: Black-box detection of cross-site scripting vulnerabilities using reinforcement learning

Link (2022, Lee et al. [39]) is an autonomous black-box web scanner that operates without any input from humans, using Reinforcement Learning (RL). This software has been observed to decrease the quantity of attack attempts and finds more true positives with fewer showing as false. Particular success has been observed in XSS vulnerability detection, which can be leveraged in penetration testing.

1.37. Lore a Red Team Emulation Tool

Lore (2023, Holm et al. [40]) is a red team emulation tool utilising boolean logic and trained models for automated red team actions, avoiding manual approaches in cyber defence exercises. Empirical tests demonstrate its model accuracy, achieving twice the compromised machines compared to expert-defined models and five times more than random action selection. Lore’s unique approach enhances red team automation, offering a more engaging and educational experience in cyber defence simulations.

1.38. LTESniffer: An Open-Source LTE Downlink/Uplink Eavesdropper

LTESniffer (2023, Hoang et al. [41]) is designed to capture both uplink and downlink LTE traffic passively. Its architecture encompasses key components such as the conversion of analogue signals to digital samples, identification of modulation schemes and other radio configurations, and data channel decoding, which processes uplink and downlink signals according to their configurations. The results demonstrate that LTESniffer significantly surpasses existing tools and commercial sniffers, achieving a success rate more than twice that of AirScope, particularly in decoding LTE packets within high-throughput scenarios and LTE-A (Advanced) environments.

1.39. Mace: Detecting privilege escalation vulnerabilities in web applications

Mace (2014, Monshizadeh et al. [42]) is an automatic privilege escalation vulnerability analysis tool for web applications. It analyses large code bases to discover zero-day vulnerabilities by observing inconsistencies in the authorisation context and understanding flaws within the application using fundamental abstractions. Implementing the tool has been observed to save weeks of labour-intensive work for security professionals in penetration testing.

1.40. MAIT: Malware Analysis and Intelligence Tool

MAIT (2021, Yucel et al.[43]) (Malware Analysis and Intelligence Tool) utilises state-of-the-art static and dynamic malware analysers alongside open-source malware databases to generate malware signatures and intelligence reports. The tool offers chronological data for malicious files, revealing related vulnerabilities and providing insights into attribution, techniques, tactics, and procedures employed by Advanced Persistent Threat groups in attacks.

1.41. A meta-language for threat modelling and attack simulations

MAL (2018, Johnson et al. [44]) is a tool designed to facilitate the creation of domain-specific attack languages for cybersecurity threat modelling and attack simulations. MAL enables the semi-automated generation and efficient computation of large attack graphs, distinguishing it from traditional, manual attack graph constructions.

1.42. Malicescript: A novel browser-based intranet threat

MaliceScript (2018, Liu et al. [45]) is a tool that introduces a novel browser-based Web attack model allowing browsers to collect intranet topology and infiltrate websites from within. The tool is designed to exploit vulnerabilities, insert foreground JavaScript code into malicious web pages, and monitor intranet topology to ensure successful infiltration. Its significance lies in its potential ease of deployment and the challenges associated with detection, emphasising the need for proactive security measures.

1.43. Masat: Model-based automated security assessment tool for cloud computing

MASAT (2015, Mjihil et al. [46]) (Model-based Automated Security Assessment Tool) is a tool designed to address security issues of cloud computing. Focused on adaptive security assessments, MASAT utilises distributed agents to evaluate the security of virtual machines at different virtualisation levels. These agents employ vulnerability scanners, representation tools for attack models like graphs, and a communication mechanism to share analysis results. MASAT's contributions include comprehensive security assessments for both the cloud infrastructure and virtual machines, efficient task distribution among agents, and parallelised subsystem assessments to reduce analysis time.

1.44. Mirage: towards a Metasploit-like framework for IoT

Mirage (2019, Cayre et al. [47]) is an open-source attack framework applied to IoT by exploiting wireless communication protocols such as Bluetooth Classic/BLE, Zigbee, Enhanced ShockBurst, Mosart, and Wi-Fi. The software is modular and has the potential for future expansion with additional functionality, which is useful for the development of new wireless protocols. The tool was tested using a Smart Bulb, demonstrating success in assessing the attack surface and reverse-engineering the wireless protocol.

1.45. Mitch: A Machine Learning Approach to the Black-Box Detection of CSRF Vulnerabilities

The study by **Mitch** (2019, Calzavara et al. [48]) introduced a browser extension that leverages supervised machine learning to identify Cross-Site Request Forgery (CSRF) vulnerabilities. This extension features an automated system that spots sensitive HTTP requests which need CSRF protection for enhanced security. It was trained using nearly 6,000 HTTP requests from popular websites, enabling it to surpass the effectiveness of existing detection methods. This advancement was evidenced by finding three CSRF vulnerabilities in production software that had previously gone unnoticed by the most advanced tools available.

1.46. MoScan: a model-based vulnerability scanner for web single sign-on services

MoScan (2021, Wei et al. [49]) is a vulnerability scanning tool for identifying security vulnerabilities in Single Sign-On (SSO) implementations through model-based scanning. By capturing network traces during the execution of SSO services, MoScan incrementally constructs and refines the state machine. This refined state machine enables MoScan to generate specific payloads for testing protocol participants, aiming to identify security vulnerabilities. MoScan's adaptability is demonstrated by testing it against other SSO services, such as Twitter, LinkedIn, and GitHub's authentication plugin in Jenkins. Despite minor adjustments needed for parameter names, MoScan's primary state machine required minor modifications to accommodate different implementations of the OAuth 2.0 standard.

1.47. NAUTILUS: Automated RESTful API Vulnerability Detection

NAUTILUS (2023, Deng et al. [50]) is proposed to improve RESTful API vulnerability scanning by addressing limitations in existing black box scanners. It utilises a novel annotation strategy to identify proper operation relations and generate meaningful sequences for vulnerability detection. Compared to four state-of-the-art tools, NAUTILUS demonstrates superior performance, detecting 141% more vulnerabilities on average and covering 104% more API operations across six tested services. In real-world scenarios, NAUTILUS detected 23 unique 0-day vulnerabilities, including a remote code execution flaw in Atlassian Confluence and high-risk issues in Microsoft Azure.

1.48. NAVEX: Precise and Scalable Exploit Generation for Dynamic Web Applications

NAVEX (2018, Alhuzali et al. [51]) is a tool addressing complex vulnerabilities in multi-tier web applications by combining dynamic and static analysis. It integrates both techniques to automatically detect vulnerabilities and create functional exploits. Evaluated over 3.2 million lines of PHP code, NAVEX successfully identified and created 204 exploits, showcasing its scalability and effectiveness in vulnerability analysis and exploit generation for large-scale applications.

1.49. NetCAT: Practical Cache Attacks from the Network

NetCAT (2020, Kurth et al. [52]) is a tool designed to exploit Data-Direct I/O (DDIO) to observe and manipulate Last-Level Cache (LLC) states, thereby leaking sensitive information from a remote target server without requiring local access or execution privileges. NetCAT reverse engineers the behaviour of DDIO on Intel processors and develops a network-based PRIME+PROBE cache attack technique by crafting specific network packets to manipulate and observe changes in the LLC's state. The efficiency of NetCAT is demonstrated in various attack scenarios, including creating covert channels between network clients and executing keystroke timing attacks to infer sensitive information from encrypted SSH sessions.

1.50. Fast, lean, and accurate: modelling password guessability using neural networks

NeuralNetworkCracking (2016, Melicher et al. [53]) is a tool designed to predict the guessability of passwords accurately and efficiently in real-time. By harnessing the power of neural networks for generating sequential data, the model utilizes an Artificial Neural Network (ANN) to anticipate the next character in a password sequence, thereby estimating the password's guessability. The effectiveness of the ANN model was benchmarked against traditional password guessability models, where it demonstrated superior performance across various scenarios. Notably, the neural networks were able to guess 70% of 4class8 passwords within 10151015 guesses, significantly outperforming the next best guessing method, which guessed 57%.

1.51. Identification and Mitigation Tool For Cross-Site Request Forgery (CSRF)

No Name (CSRF) (2020, Rankothge and Randeniya [54]) focuses on the automated detection and mitigation of Cross-Site Request Forgery (CSRF) vulnerabilities in PHP-based web applications [54]. It identifies and mitigates CSRF vulnerabilities by scanning form tags and automatically adding security solutions. However, the evaluation is limited to a few test websites, and the tool is applicable exclusively to PHP.

1.52. Development of a suite of IPv6 vulnerability scanning tests using the TTCN-3 language

The No Name (TTCN-3) tool, (2018, Leal and Teixeira [55]), is a collection of tests designed to find security weaknesses in IPv6 setups. It uses TTCN-3, a language for defining and executing tests that is endorsed by the European Telecommunications Standards Institute. The tool was evaluated using the SAMATE Juliet 1.2 suite from NIST [56] and proved its effectiveness by identifying and exploiting flaws, particularly focusing on preventing DOS attacks on the IPv6 ICMPv6 protocol over Ethernet networks.

1.53. *Cracking-Resistant Password Vaults Using Natural Language Encoders*

NoCrack (2015, Chatterjee et al. [57]) is a password-cracking tool that overcomes the limitations of a previous design, Kamouflage [58], which was shown to degrade security. NoCrack utilizes Natural Language Encoder (NLE) as a scheme for secure encoding. This approach utilises natural language processing (n-gram models) with probabilistic context-free grammars, to construct NLEs. These encoders can create realistic decoy passwords, thereby significantly enhancing the resistance of password vaults against cracking efforts.

1.54. *NodeXP: NOde.js server-side JavaScript injection vulnerability DETection and eXPloitation*

NodeXP (2021, Ntantogian et al. [59]) is a tool addressing security vulnerabilities in web applications, particularly Server-Side JavaScript Injection (SSJI) threats in Node.js. It detects and automatically exploits SSJI vulnerabilities, using obfuscation for enhanced stealth. NodeXP employs dynamic analysis with result and blind-based injection techniques for detection and automated exploitation. Its advanced functionalities, including attack vector obfuscation, distinguish it from other tools, allowing evasion of filtering mechanisms and security measures. Thorough assessments show NodeXP outperforming existing scanners. Released as open-source, it aims to drive research in SSJI vulnerabilities. Contributions include SSJI analysis, a novel detection method, and the discovery of a 0-day SSJI.

1.55. *ObjectMap: detecting insecure object deserialization*

ObjectMap (2019, Koutroumpouchos et al. [60]) is a tool designed to address serialization-based vulnerabilities prevalent in web applications, particularly in Java and PHP. It aims to fill the existing gap by detecting implementation-agnostic deserialization and object injection vulnerabilities. Additionally, it introduces the first deserialization test environment, serving as a platform for vulnerability detection tool evaluation and educational purposes. Both tools are highly extendable and represent a unique combination of features in this domain, potentially fostering further research and aiding the development of more comprehensive solutions.

1.56. *OMEN: Faster password guessing using an ordered Markov enumerator*

OMEN (2015, Durmuth et al. [61]) employs an advanced Markov model algorithm that generates passwords in order of decreasing probability, a departure from earlier approaches [62]. This method involves discretising probabilities into bins and iterating over them to identify all passwords corresponding to each bin's probability. OMEN has shown a significant improvement in guessing speed over existing methods, accurately guessing over 40% of passwords within the first 90 million attempts, marking a notable efficiency gain against other methods such as John the Ripper (JtR) and probabilistic context-free grammar (PCFG)-based strategies [63].

1.57. *OSV: OSPF vulnerability checking tool*

OSV (2017, Kasemsuwan et al. [64]) is a tool that targets OSPF (Open Shortest Path First) network vulnerabilities prevalent in enterprise networks. Typically, OSPF vulnerabilities in router implementations can be mitigated through firmware updates. The tool conducts penetration tests and generates reports, aiding network operators in identifying and rectifying security issues. The tool has been validated on Quagga and Cisco routers.

1.58. *Owfuzz: Discovering Wi-Fi Flaws in Modern Devices through Over-The-Air Fuzzing*

Owfuzz (2023, Cao et al. [65]) is a fuzzing tool designed for discovering security flaws in Wi-Fi protocols through over-the-air fuzzing methods. Owfuzz sets itself apart from other Wi-Fi fuzzers by offering the capability to conduct fuzzing tests on any Wi-Fi device, enabling the fuzzing of all three Wi-Fi frame types (management, control, and data) across every version of the 802.11 standards, and facilitating interactive testing for a variety of protocol models. In experiments conducted on over 40 contemporary Wi-Fi devices from 7 chipset manufacturers, Owfuzz identified 23 security issues, resulting in the assignment of 8 CVE IDs.

1.59. PassGAN: A Deep Learning Approach for Password Guessing

PassGAN (2019, Hitaj et al. [66]) utilises a Generative Adversarial Network (GAN) trained on real-world datasets to make accurate password guesses. By autonomously learning the distribution of real passwords from actual leaks, PassGAN eliminates the need for manual rule creation. When combined with HashCat, PassGAN’s output successfully matches 51%-73% more passwords than HashCat alone. This demonstrates the tool’s capacity to automatically extract nuanced password properties not encompassed by current state-of-the-art rules.

1.60. PassGPT: Password modelling and (Guided) Generation with Large Language Models

PassGPT (2023, Rando et al. [67]) is a tool utilizing GPT-2 architecture and trained on leaked passwords, aiming to improve password guessing and strength estimation. PassGPT incorporates vector quantization to enhance the complexity of password generation (PassVQT). Comprehensive tests were conducted to evaluate PassGPT’s efficacy against current password-guessing tools and to assess its generalization capabilities across various datasets. PassGPT recovers 41.9% of the test set among 109109 guesses, whereas state-of-the-art GAN models matched 23.33%.

1.61. The Revenge of Password Crackers: Automated Training of Password Cracking Tools

PasswordCrackingTraining (2022, Di Campi et al. [68]) is a password-cracking trainer combining various password-cracking techniques, trained and tested on a dataset of over 700 million real passwords. Their methodology includes evaluating existing hashcat rules and dictionaries and developing efficient algorithms to simulate mask attacks without actual password cracking. This approach nearly doubles the success rate of password-cracking tools compared to off-the-shelf configurations, achieving over 70% success in certain instances.

1.62. PenQuest: a gamified attacker/defender meta model for cyber security assessment and education

PenQuest (2020, Luh et al. [69]) is a tool designed as a dynamic, multiplayer game that captures the behaviour of attackers and defenders over time, using a rule set based on established information security sources such as STIX CAPEC, CVE/CWE, and NIST SP 800-53. The tool is designed to improve cybersecurity risk assessments and serve as a platform for simulating specific attack scenarios within an abstracted IT infrastructure. PenQuest is structured around three main layers: the service layer, the information layer, and the event layer, each contributing to a comprehensive representation of a cybersecurity scenario.

1.63. PentestGPT: An LLM-empowered Automatic Penetration Testing Tool

PentestGPT (2023, Deng et al. [70]) utilises Large Language Models (LLMs) for penetration testing, revealing their proficiency in specific tasks but struggles in holistic understanding. Based on LLM-powered automatic penetration testing tool, PENTESTGPT has been released with three modules addressing distinct sub-tasks. Evaluation demonstrates PentestGPT’s 228.6% improvement over GPT-3.5 and effectiveness in real-world challenges. The tool employs Reasoning, Generation, and Parsing Modules, simulating human-like behaviour and adopting a divide-and-conquer problem-solving approach.

1.64. phpSAFE: A Security Analysis Tool for OOP Web Application Plugins

PhpSAFE (2015, Nunes et al. [71]) is introduced as a solution to identify security vulnerabilities in PHP plugins developed with Object-Oriented Programming (OOP) for web applications. In contrast to existing free tools that lack OOP support, PhpSAFE excels in static code analysis, outperforming two well-known tools when evaluated with 35 plugins in a popular Content Management System. The results highlight the prevalence of vulnerabilities in these plugins, indicating an increasing trend over time, emphasizing the necessity of robust security measures in plugin development.

1.65. PJCT: Penetration testing based JAVA code testing tool

PJCT (2015, Jain et al. [72]) is a tool designed to secure attributes in Java code. This addresses a crucial aspect of secure software development while emphasizing early detection of vulnerabilities during the development process. PJCT highlights seven essential security attributes for identifying vulnerabilities in Java applications. These attributes include network and security packages, techniques for handling exceptions, secure data packet transmission and multithreading.

1.66. Project Achilles: A Prototype Tool for Static Method-Level Vulnerability Detection of Java Source Code Using a Recurrent Neural Network

Project Achilles (2019, Saccente et al. [73]) is a prototype tool for static method-level vulnerability detection in Java source code, leveraging LSTM Recurrent Neural Networks. The tool utilizes NIST’s Juliet Java Suite, which includes several examples of defective Java methods for various vulnerabilities. Employing an array of LSTM networks, the tool achieves over 90% accuracy for 24 out of 29 Common Weakness Enumeration (CWE) vulnerabilities in an evaluation with around 45,000 test cases.

1.67. PURITY: a Planning-based secURITY testing tool

PURITY (2015, Bozic and Wotawa [74]) is a security testing tool that employs a planning-centric methodology to preemptively identify and rectify vulnerabilities during the software development cycle. By executing automated test cases, PURITY focuses on detecting prevalent security flaws like SQL injections and cross-site scripting, simulating malicious activities through predetermined sequences of actions. PURITY generates concrete test cases from plans based on specific initial values and predefined actions, mimicking the behaviours of potential attackers, providing a versatile platform for both automated and manual evaluation of web applications throughout the software development life cycle.

1.68. Pyciuti: A Python Based Customizable and Flexible Cybersecurity Utility Tool for Penetration Testing

Pyciuti (2023, Muralidharan et al. [75]) is a Python-based general-purpose tool that integrates custom scanners, crawlers, malware functionalities, and more, offering flexibility and customisation for both beginners and experienced professionals. Users can access various subdomains such as OSINT, network-based tools, web-based tools, malware tools, and documentation, streamlining the execution of tasks. The tool records findings, scans, and exploits, providing a consolidated report while prioritising accuracy and performance.

1.69. RAT: Reinforcement-Learning-Driven and Adaptive Testing for Vulnerability Discovery in Web Application Firewalls

RAT (2022, Amouei et al. [76]) (Reinforcement-Learning-Driven and Adaptive Testing) is a tool for discovering vulnerabilities, specifically SQL injection (SQLi) and Cross-site Scripting (XSS), in Web Application Firewalls (WAFs). The tool’s methodology begins with clustering similar attack samples, followed by applying a reinforcement learning technique that efficiently identifies bypassing attack patterns. The approach is further refined by integrating an adaptive search algorithm, which aids in discovering almost all possible bypassing payloads with higher efficiency. RAT outperforms existing methods by significant margins (33.53% and 63.16% on average) when tested against well-configured WAFs.

1.70. Revealer: Detecting and Exploiting Regular Expression Denial-of-Service Vulnerabilities

Revealer (2021, Liu et al. [77]) is a tool designed to detect and exploit Regular expression Denial-of-Service (ReDoS) vulnerabilities present in extended-featured regular expressions (regexes). Revealer employs a combined method using static and dynamic analyses to identify vulnerable regex structures and generate attack strings that induce recursive backtracking. In evaluation against 29,088 regexes and comparison with three state-of-the-art tools, Revealer outperformed existing solutions, detecting all known vulnerabilities, finding 213 new vulnerabilities whilst surpassing the highest performing tool by 140.64%. Additionally, it detected 45 vulnerable regexes in real-world applications, demonstrating its effectiveness and efficiency in detecting and exploiting ReDoS vulnerabilities.

1.71. *RiscyROP: Automated Return-Oriented Programming Attacks on RISC-V and ARM64*

RiscyROP (2022, Cloosters et al. [78]) is an automated Return-Oriented Programming (ROP) tool specifically designed for RISC-V and ARM64 architectures. It employs symbolic execution to analyze available gadgets and autonomously generate complex multi-stage chains for arbitrary function calls. The tool's effectiveness is evidenced by its analysis of real-world software from public repositories, demonstrating its capability to identify usable gadgets for executing attacker-controlled function calls.

1.72. *Robin: A Web Security Tool*

Robin (2020, Giroto and Zorzo [79]) is a web security tool that includes a Proxy module for listing, intercepting, and editing HTTP and HTTPS requests. Robin's capabilities extend to active scanning, brute-force attack simulations, encoding/decoding contents using various hashing patterns, and providing a detailed Wiki module for understanding and safeguarding against common vulnerabilities. The tool's potential is demonstrated through a real case study involving a news company's website.

1.73. *ROSploit: Cybersecurity Tool for ROS*

ROSploit (2019, Rivera et al. [80]) is a tool for the assessment of the security of the Robot Operating System (ROS). The tool covers reconnaissance and exploitation. Its reconnaissance component integrates with Nmap, offering two scripts for master node scans and wide port scans, identifying ROS nodes and services. The exploitation component, built in Python, mirrors Metasploit's modular design. Moreover, ROSploit enables simulated attacks without a complete ROS installation.

1.74. *RT-RCT: an online tool for real-time retrieval of connected things*

RT-RCT (2021, Fagroud et al. [81]) utilises network port scanning techniques [82] to extract real-time data from connected devices within the IoT domain. The main goal is to introduce a retrieval tool that swiftly provides users with current information about each requested entity. Real-time device data is collected through network port scanning, with a specific emphasis on the Python-nmap library. The data collection process involves a series of scans, each targeting specific information that may take considerable time to retrieve. To reduce time, parallel scans are implemented, ensuring simultaneous and accelerated execution of all scans to improve overall retrieval speed and efficiency.

1.75. *Scanner++: Enhanced Vulnerability Detection of Web Applications with Attack Intent Synchronization*

Scanner++ (2023, Yin et al. [83]) demonstrates an enhancement of vulnerability assessment scanning technology in web applications. By utilizing purification mechanisms to refine attack intents from the request packets in the base scanners. The scanner has a synchronisation mechanism for runtime intent for the scanner's detection spots. This has been evaluated against four practitioner tools including BurpSuite, AWVS, Arachni, and ZAP testing on real-world finance web apps where the tool demonstrated to have a higher coverage when being compared to the practitioner tools by approximately 15%-70%.

1.76. *On the Semantic Patterns of Passwords and their Security Impact*

SemanticGuesser (2014, Veras et al. [84]) is a framework that uses Natural Language Processing (NLP) techniques for the segmentation, semantic classification, and generalisation of passwords. The methodology begins with password segmentation to isolate distinct words or elements found within the passwords. Following this, it employs part-of-speech tagging and semantic classification through NLP methodologies, primarily utilising resources such as WordNet to decipher the meanings of the segments. Experimental results have shown that this method can guess about 67% more passwords from the LinkedIn data breach and 32% more from the MySpace leak, demonstrating its effectiveness in understanding and exploiting the semantic structures of passwords.

1.77. SerialDetector: Principled and Practical Exploration of Object Injection Vulnerabilities for the Web

SerialDetector (2021, Shcherbakov et al. [85]) employs taint-based dataflow analysis to automatically detect OIV (Object Injection Vulnerability) patterns in .NET assemblies. By identifying untrusted information flow from public entry points to sensitive methods, it uncovers vulnerabilities and matches them with available gadgets, validating the feasibility of OIV attacks. The evaluation conducted on Azure DevOps Server showcased SerialDetector’s effectiveness by discovering three CVEs and demonstrating its capability in identifying remote code execution vulnerabilities. Additionally, it performed a broad security analysis of recent CVEs, affirming its efficiency and effectiveness in OIV detection.

1.78. ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services

ShoVAT (2016, Genge et al. [86]) is a vulnerability analysis tool based on Shodan. This software acquires the output of traditional queries in Shodan to analyse service-specific data by leveraging the search engine. By adding to existing modules this tool provides the opportunity to analyse other targets in the penetration test.

1.79. Snout: An Extensible IoT Pen-Testing Tool

Snout (2019, Mikulskis et al. [87]) is a Software-Defined Radio (SDR)-based utility toolkit for passive sniffing and interaction with various IoT protocols, including Zigbee, Bluetooth, and Wi-Fi. Available as a Python 3 package or as a Docker container, Snout enables enumeration and analysis of multiple wireless protocols, including non-IP IoT protocols.

1.80. SOA-Scanner: an integrated tool to detect vulnerabilities in service-based infrastructures

SOA-Scanner (2013, Antunes et al. [88]) is a vulnerability analysis tool designed for service-oriented architectures (SOA) after the application has been deployed. The tool tests services based on their level of access using an iterative process to discover services, resources, and interactions in real-time. Additionally, it provides anomaly detection, categorising services based on whether they are within reach, under full control, or under partial control.

1.81. Spicy: a unified deep packet inspection framework for safely dissecting all your data

Spicy (2016, Sommer et al. [89]) includes a format specification language, a compiler toolchain, and an API to address the challenges of deep packet inspection (DPI) across diverse network protocols and file formats. The Spicy framework automates the dissection process, enhancing DPI efficiency and safety. Enabling developers to create specific dissectors for varied protocols, Spicy is a reliable DPI tool. Its flexibility makes it valuable for processing network data from untrusted sources in diverse formats.

1.82. SuperEye: A Distributed Port Scanning System

SuperEye (2019, Li et al. [90]) is an advanced distributed port scanning system adopting a distributed structure with task redundancy and real-time state display. SuperEye’s core control subsystem efficiently manages distributed nodes, tasks, and result processing, leveraging a custom protocol stack to optimise resource utilisation. The distributed architecture significantly boosts scanning speed, mitigating risks associated with Intrusion Detection Systems (IDS). Innovative features include a script for port scanning and visualisation tools offering real-time updates.

1.83. SVED: Scanning, Vulnerabilities, Exploits and Detection

SVED (2016, Holm et al. [91]) is a tool designed for secure and replicable experiments, enabling controlled execution and logging of malicious activities, including software exploits and intrusion detection alerts. Its distributed architecture supports extensive experiments involving numerous attackers, sensors, and targets. SVED automatically integrates threat intelligence from diverse services, ensuring up-to-date information for enhanced experimentation and analysis in cybersecurity.

1.84. *A Hybrid Threat Model for Smart Systems*

TAMELESS (2023, Valenza et al. [92]) is an automated threat modelling tool that includes a threat modelling approach integrating cyber, physical, and human elements, and a threat analysis method designed to evaluate the security posture of system components. TAMELESS can analyse threats, verify security properties, and produce graphical outputs of its analyses, thereby assisting security architects in identifying optimal prevention and mitigation solutions. The efficiency and applicability of TAMELESS have been demonstrated through case study evaluations involving unauthorised access to safe boxes, web servers, and wind farms, showcasing its effectiveness in real-world scenarios.

1.85. *TChecker: Precise Static Inter-Procedural Analysis for Detecting Taint-Style Vulnerabilities in PHP Applications*

TChecker (2022, Luo et al. [93]) introduces a context-sensitive inter-procedural static taint analysis tool specifically tailored for PHP applications, addressing the challenge of taint-style vulnerabilities like SQL injection and cross-site scripting. By modelling PHP objects and dynamic language features, TChecker conducts iterative data-flow analysis to refine object types and accurately identify call targets. Comprehensive evaluations across diverse modern PHP applications showcased TChecker’s effectiveness, discovering 18 previously unknown vulnerabilities while outperforming existing static analysis tools in vulnerability detection. It not only detected more vulnerabilities but also maintained a relatively good precision, surpassing competitors while releasing its source code to foster further research in this domain.

1.86. *Detecting and exploiting second order denial-of-service vulnerabilities in web applications*

TORPEDO (2015, Olivo et al. [94]) is a second-order vulnerability scanning tool that detects Denial of Service (DoS), Cross-Site Scripting (XSS), and SQL Injection. The program searches for two-phased DoS attacks that work by polluting a database with junk entries and resource exhaustion. When applied to six highly used web apps, it detected thirty-seven vulnerabilities and eighteen false positives.

1.87. *UE Security Reloaded: Developing a 5G Standalone User-Side Security Testing Framework*

UE Security Reloaded (2023, Hoang et al. [95]) is an open-source security testing framework specifically developed for 5G Standalone (SA) User Equipment (UE). This tool enhances existing open-source suites (Open5GS and srsRAN) by creating an extensive range of test cases for both the 5G Non-Access Stratum (NAS) and Radio Resource Control (RRC) layers. Such an approach offers in-depth insights through experiments on 5G SA mobile phones. The framework allows for the transmission of 5G control-plane messages (NAS and RRC) to a UE and facilitates the modification of these messages to examine the UE’s reactions under a variety of conditions.

1.88. *Untangle: Aiding Global Function Pointer Hijacking for Post-CET Binary Exploitation*

Untangle (2023, Bertani et al. [96]) is a tool that exploits global function pointer hijacking in order to defeat Intel’s Control-Flow Enforcement Technology (CET). The method combines symbolic execution and static code analysis to identify global function pointers within C libraries, which, when compromised, facilitate control-flow hijacking attacks. Experimental results demonstrated the effectiveness of Untangle in identifying global function pointers across eight widely used open-source C libraries.

1.89. *VAPE-BRIDGE: Bridging OpenVAS Results for Automating Metasploit Framework*

VAPE-BRIDGE (2022, Vimala et al. [97]) is a tool designed to streamline the transition between vulnerability assessment (VA) and penetration testing (PenTest) processes by automating the conversion of scan results from the Open Vulnerability Assessment Scanner (OpenVAS) into executable scripts for the Metasploit Framework. The VAPE-BRIDGE system comprises three main components: Scan result extraction, responsible for parsing the VA scan results from OpenVAS; Target list repository, accountable for maintaining a database of identified vulnerabilities to be used in the PenTest process; and the Automated shell scripts exploitation, which generates shell scripts based on the extracted vulnerabilities, which are then executed within Metasploit to simulate attacks and test the system’s resilience.

1.90. Vera: A flexible model-based vulnerability testing tool

VERA (2013, Blome et al. [98]) is an automated tool that supports Penetration Testers to define attacker models (separating payloads and behaviour) using state machines for vulnerability analysis in web applications. The models acquired are then converted into libraries for specific vulnerability targeting. The tool is highly flexible with the availability to expand and integrate custom libraries to enhance functionality.

1.91. VUDDY: A Scalable Approach for Vulnerable Code Clone Discovery

VUDDY (2017, Kim et al. [99]) aims to detect defective code in large open-source programs. In particular, it is capable to process a billion lines of code in under 15 hours and quickly identify code clones using function-level granularity and a length-filtering technique. The evaluation included comparison with four other code clone detection methods and VUDDY showed better scalability and accuracy. For example, it was possible to find zero-day vulnerabilities popular software like Apache Web Server and Ubuntu Linux OS.

1.92. Vulcan: Vulnerability assessment framework for cloud computing

Vulcan (2013, Kamongi et al. [100]) is a tool for vulnerability analysis and remediation in cloud and mobile computing. This tool provides software and zero-day vulnerability modelling and assessments. The tool is very flexible, presenting the opportunity to add original modules by developers and the integration of Vulcan into other vulnerability analysis tools that, for example, focus on web application vulnerabilities to expand their assessment to cloud and mobile technology.

1.93. VulCNN: An Image-Inspired Scalable Vulnerability Detection System

VulCNN (2022, Wu et al. [101]) is designed to address the limitations of existing text-based and graph-based vulnerability detection methods. The tool converts the source code of functions into images that preserve program details and then uses these images to detect vulnerabilities through a Convolutional Neural Network (CNN) model. VulCNN was evaluated on a dataset of 13,687 vulnerable and 26,970 non-vulnerable functions. With an accuracy of 82% and a True Positive Rate (TPR) of 94%, VulCNN outperformed eight other state-of-the-art vulnerability detectors, including both commercial tools and deep learning-based approaches.

1.94. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection

VulDeePecker (2018, Zhen et al. [102]) employs deep learning for software vulnerability detection, aiming to reduce reliance on human-defined features and mitigate false negatives. It uses code gadgets to represent and transform programs into vectors suitable for deep learning. The system's evaluation, using the first vulnerability dataset for deep learning, demonstrates significantly fewer false negatives compared to other methods, with reasonable false positives. VulDeePecker successfully detects four previously unreported vulnerabilities in Xen, Seamonkey, and Libav, unnoticed by other detection systems, highlighting its efficacy in uncovering vulnerabilities missed by existing approaches.

1.95. An Intelligent and Automated WCMS Vulnerability-Discovery Tool: The Current State of the Web

Vulnet (2019, Cigoj and Blazic [103]) is characterized by its capability to conduct automated, rapid, and dynamic vulnerability scans across a wide array of internet websites. Specifically, VulNet focuses on those utilising the WordPress Web Content Management Systems (WCMS) and its associated plugins. A crucial aspect of the tool involves the application of a scoring mechanism tailored to evaluate known vulnerabilities. It's important to note that VulNet's vulnerability detection is limited to WordPress web applications and their associated plugins.

1.96. Vulnsloit: A Module for Semi-automatic Exploitation of Vulnerabilities

Vulnsloit (2020, Castiglione et al. [104]) is a semi-automatic penetration testing tool that collects vulnerability data using existing tools like the Nmap Scripting Engine (NSE) and the Vulscan scanner [82]. This data is then processed to identify relevant exploits from various repositories, including local and remote sources. In preliminary testing on Metasploitable2, Vulnsloit identified 23 open ports and approximately 220,000 vulnerabilities.

1.97. VulPecker: an automated vulnerability detection system based on code similarity analysis

VulPecker (2016, Li et al. [105]) automatically detects specific vulnerabilities within software source code. Leveraging a set of defined features characterizing patches and utilizing code-similarity algorithms tailored for different vulnerability types, VulPecker successfully identifies 40 vulnerabilities not listed in the National Vulnerability Database (NVD). Among these, 18 previously unknown vulnerabilities (anonymized for ethical considerations) are confirmed, while 22 vulnerabilities have been patched silently by vendors in later product releases.

1.98. WAPTT-Web application penetration testing tool

WAPTT (2014, Duric et al. [106]) is designed for web application penetration testing using page similarity detection. The structure of the tool is modular and when compared to tools such as Nikto, Vega, and ZAP, the tool detected similar or greater quantity of vulnerabilities in the area of XSS, but at the price of increasing detection time.

1.99. webFuzz: Grey-Box Fuzzing for Web Applications

WebFuzz (2021, van Rooij et al. [107]) is a gray-box fuzzing tool designed for web applications, with a focus on discovering vulnerabilities like cross-site scripting (XSS). It effectively utilises instrumentation, outperforming black-box fuzzers by identifying XSS vulnerabilities swiftly and covering more code. WebFuzz has demonstrated its capability by discovering one zero-day vulnerability in WordPress and five in CE-Phoenix.

1.100. Identification and Mitigation Tool for Sql Injection Attacks (SQLIA)

WebVIM (2020, Rankothge et al. [108]) is a tool designed for identifying SQL injection vulnerabilities in PHP-based web applications during the development phase. When vulnerabilities are detected, WebVIM automatically adds security solutions to the source code. However, the evaluation is very limited, and the tool focuses exclusively on PHP applications.

References

- [1] P. Modesti, L. Golightly, L. Holmes, C. Opara, M. Moscini, Bridging the Gap: A Survey and Classification of Research-Informed Ethical Hacking Tools, *Journal of Cybersecurity and Privacy* (2024).
- [2] D. Pasquini, M. Cianfriglia, G. Ateniese, M. Bernaschi, Reducing bias in modeling real-world password strength via deep learning and dynamic dictionaries, in: M. D. Bailey, R. Greenstadt (Eds.), 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021, USENIX Association, 2021, pp. 821–838.
URL <https://www.usenix.org/conference/usenixsecurity21/presentation/pasquini>
- [3] M. Fu, C. Tantithamthavorn, T. Le, Y. Kume, V. Nguyen, D. Phung, J. Grundy, Aibug Hunter: A practical tool for predicting, classifying and repairing software vulnerabilities, *Empirical Software Engineering* 29 (1) (Nov. 2023). doi:10.1007/s10664-023-10346-3.
- [4] L.-H. Chen, F.-H. Hsu, Y. Hwang, M.-C. Su, W.-S. Ku, C.-H. Chang, Armory: An automatic security testing tool for buffer overflow defect detection, *Computers & Electrical Engineering* 39 (7) (2013) 2233–2242. doi:10.1016/j.compeleceng.2012.07.005.
- [5] N. Moscovich, R. Bitton, Y. Mallah, M. Inokuchi, T. Yagyu, M. Kalech, Y. Elovici, A. Shabtai, Autosplit: A fully automated framework for evaluating the exploitability of security vulnerabilities (2020). doi:10.48550/arXiv.2007.00059.

- [6] R. Egert, T. Grube, D. Born, M. Mühlhäuser, AVAIN – a framework for automated vulnerability indication for the iot in ip-based networks, in: 2019 International Conference on Networked Systems, NetSys 2019, Munich, Germany, March 18-21, 2019, IEEE, 2019, pp. 1–3. doi:10.1109/NetSys.2019.8854493.
- [7] B. Blumbergs, R. Vaarandi, Bbuzz: A bit-aware fuzzing framework for network protocol systematic reverse engineering and analysis, in: 2017 IEEE Military Communications Conference, MILCOM 2017, Baltimore, MD, USA, October 23-25, 2017, IEEE, 2017, pp. 707–712. doi:10.1109/MILCOM.2017.8170785.
- [8] B. Eriksson, A. Stjerna, R. De Masellis, P. Rüemmer, A. Sabelfeld, Black ostrich: Web application scanning with string solvers, in: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS '23, ACM, 2023. doi:10.1145/3576915.3616582.
- [9] B. Eriksson, G. Pellegrino, A. Sabelfeld, Black widow: Blackbox data-driven web scanning, in: 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, IEEE, 2021, pp. 1125–1142. doi:10.1109/SP40001.2021.00022.
- [10] Z. Luo, J. Yu, F. Zuo, J. Liu, Y. Jiang, T. Chen, A. Roychoudhury, J. Sun, Bleem: Packet sequence oriented fuzzing for protocol implementations, in: J. A. Calandrino, C. Troncoso (Eds.), 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023, USENIX Association, 2023, pp. 4481–4498.
URL <https://www.usenix.org/conference/usenixsecurity23/presentation/luo-zhengxiong>
- [11] S. Faily, R. Scandariato, A. Shostack, L. Sion, D. Ki-Aries, Contextualisation of data flow diagrams for security analysis, in: H. E. III, O. Gadyatskaya (Eds.), Graphical Models for Security - 7th International Workshop, GraMSec 2020, Boston, MA, USA, June 22, 2020 Revised Selected Papers, Vol. 12419 of Lecture Notes in Computer Science, Springer, 2020, pp. 186–197. doi:10.1007/978-3-030-62230-5_10.
- [12] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, J. A. Halderman, A search engine backed by Internet-wide scanning, in: 22nd ACM Conference on Computer and Communications Security, 2015, pp. 542–553. doi:10.1145/2810103.2813703.
- [13] Z. Durumeric, E. Wustrow, J. A. Halderman, {ZMap}: fast internet-wide scanning and its security applications, in: 22nd USENIX Security Symposium (USENIX Security 13), 2013, pp. 605–620.
- [14] A. Alhuzali, B. Eshete, R. Gjomemo, V. N. Venkatakrishnan, Chainsaw: Chained automated workflow-based exploit generation, in: E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, S. Halevi (Eds.), Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, ACM, 2016, pp. 641–652. doi:10.1145/2976749.2978380.
- [15] F. Yamaguchi, C. Wressnegger, H. Gascon, K. Rieck, Chucky: exposing missing checks in source code for vulnerability discovery, in: A. Sadeghi, V. D. Gligor, M. Yung (Eds.), 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013, ACM, 2013, pp. 499–510. doi:10.1145/2508859.2516665.
- [16] A. Stasinopoulos, C. Ntantogian, C. Xenakis, Commix: automating evaluation and exploitation of command injection vulnerabilities in web applications, Int. J. Inf. Sec. 18 (1) (2019) 49–72. doi:10.1007/s10207-018-0399-z.
- [17] S. Rahaman, Y. Xiao, S. Afrose, F. Shaon, K. Tian, M. Frantz, M. Kantarcioglu, D. D. Yao, Cryptoguard: High precision detection of cryptographic vulnerabilities in massive-sized java projects, in: L. Cavallaro, J. Kinder, X. Wang, J. Katz (Eds.), Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019, ACM, 2019, pp. 2455–2472. doi:10.1145/3319535.3345659.

- [18] R. Li, W. Diao, Z. Li, J. Du, S. Guo, Android custom permissions demystified: From privilege escalation to design shortcomings, in: 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, IEEE, 2021, pp. 70–86. doi:[10.1109/SP40001.2021.00070](https://doi.org/10.1109/SP40001.2021.00070).
- [19] G. Pellegrino, M. Johns, S. Koch, M. Backes, C. Rossow, Deemon: Detecting CSRF with dynamic analysis and property graphs, in: B. Thuraisingham, D. Evans, T. Malkin, D. Xu (Eds.), Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017, ACM, 2017, pp. 1757–1771. doi:[10.1145/3133956.3133959](https://doi.org/10.1145/3133956.3133959).
- [20] S. Lee, C. Yoon, C. Lee, S. Shin, V. Yegneswaran, P. A. Porras, Delta: A security assessment framework for software-defined networks., in: NDSS, 2017. doi:[10.14722/ndss.2017.23457](https://doi.org/10.14722/ndss.2017.23457).
- [21] C. K. Ng, Y. Yusof, N. S. N. Ab Aziz, Dfbc recon tool: Digital footprint and breach check reconnaissance tool, in: 2021 14th International Conference on Developments in eSystems Engineering (DeSE), IEEE, 2021, pp. 526–530. doi:[10.1109/dese54285.2021.9719440](https://doi.org/10.1109/dese54285.2021.9719440).
- [22] N. Redini, A. Continella, D. Das, G. De Pasquale, N. Spahn, A. Machiry, A. Bianchi, C. Kruegel, G. Vigna, Diane: Identifying fuzzing triggers in apps to generate under-constrained inputs for iot devices, in: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, 2021, pp. 484–500. doi:[10.1109/sp40001.2021.00066](https://doi.org/10.1109/sp40001.2021.00066).
- [23] F. Aljaafari, R. Menezes, M. A. Mustafa, L. C. Cordeiro, Finding security vulnerabilities in iot cryptographic protocol and concurrent implementations (2021). doi:[10.48550/arXiv.2103.11363](https://doi.org/10.48550/arXiv.2103.11363).
- [24] L. Xu, M. Xu, F. Li, W. Huo, ELAID: detecting integer-overflow-to-buffer-overflow vulnerabilities by light-weight and accurate static analysis, Cybersecur. 3 (1) (2020) 1–19. doi:[10.1186/s42400-020-00058-2](https://doi.org/10.1186/s42400-020-00058-2).
- [25] C. Lattner, V. S. Adve, LLVM: A compilation framework for lifelong program analysis & transformation, in: 2nd IEEE / ACM International Symposium on Code Generation and Optimization (CGO 2004), 20-24 March 2004, San Jose, CA, USA, IEEE Computer Society, 2004, pp. 75–88. doi:[10.1109/CGO.2004.1281665](https://doi.org/10.1109/CGO.2004.1281665).
- [26] M. C. Ghanem, T. M. Chen, M. A. Ferrag, M. E. Kettouche, Esascf: Expertise extraction, generalization and reply framework for optimized automation of network security compliance, IEEE Access 11 (2023) 129840–129853. doi:[10.1109/access.2023.3332834](https://doi.org/10.1109/access.2023.3332834).
- [27] Y. Zhang, W. Huo, K. Jian, J. Shi, L. Liu, Y. Zou, C. Zhang, B. Liu, Esrfuzzer: an enhanced fuzzing framework for physical SOHO router devices to discover multi-type vulnerabilities, Cybersecur. 4 (1) (2021) 24. doi:[10.1186/s42400-021-00091-9](https://doi.org/10.1186/s42400-021-00091-9).
- [28] M. Rak, G. Salzillo, D. Granata, Esseca: An automated expert system for threat modelling and penetration testing for iot ecosystems, Computers and Electrical Engineering 99 (2022) 107721. doi:[10.1016/j.compeleceng.2022.107721](https://doi.org/10.1016/j.compeleceng.2022.107721).
- [29] V. Visoottiviseth, P. Jutadhammakorn, N. Pongchanchai, P. Kosolyudhthasarn, Firmaster: Analysis tool for home router firmware, in: 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2018, pp. 1–6. doi:[10.1109/JCSSE.2018.8457340](https://doi.org/10.1109/JCSSE.2018.8457340).
- [30] S. Park, D. Kim, S. Jana, S. Son, FUGIO: automatic exploit generation for PHP object injection vulnerabilities, in: K. R. B. Butler, K. Thomas (Eds.), 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022, USENIX Association, 2022, pp. 197–214.
- [31] T. Lee, S. Wi, S. Lee, S. Son, FUSE: finding file upload bugs via penetration testing, in: 27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020, The Internet Society, 2020. doi:[10.14722/ndss.2020.23126](https://doi.org/10.14722/ndss.2020.23126).

- [32] J. Chen, S. Hu, H. Zheng, C. Xing, G. Zhang, Gail-pt: An intelligent penetration testing framework with generative adversarial imitation learning, *Computers & Security* 126 (2023) 103055. doi:10.1016/j.cose.2022.103055.
- [33] F. Yu, M. V. Martin, Gnpassgan: Improved generative adversarial networks for trawling offline password guessing, in: *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2022. doi:10.1109/eurospw55150.2022.00009.
- [34] S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, D. S. Kim, Harmer: Cyber-attacks automation and evaluation, *IEEE Access* 8 (2020) 129397–129414. arXiv:2006.14352, doi:10.1109/ACCESS.2020.3009748.
- [35] R. Sommer, M. Vallentin, L. De Carli, V. Paxson, Hilti: an abstract execution environment for deep, stateful network traffic analysis, in: *Proceedings of the 2014 Conference on Internet Measurement Conference, IMC '14*, ACM, 2014. doi:10.1145/2663716.2663735.
- [36] J. Chen, W. Diao, Q. Zhao, C. Zuo, Z. Lin, X. Wang, W. C. Lau, M. Sun, R. Yang, K. Zhang, Iotfuzzer: Discovering memory corruptions in iot through app-based fuzzing, in: *25th Annual Network and Distributed System Security Symposium, NDSS 2018*, San Diego, California, USA, February 18-21, 2018, The Internet Society, 2018. doi:10.14722/ndss.2018.23159.
- [37] D. M. Stallenberg, A. Panichella, JCOMIX: a search-based tool to detect XML injection vulnerabilities in web applications, in: *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2019*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1090–1094. doi:10.1145/3338906.3341178.
- [38] M. Xu, S. Li, L. Xu, F. Li, W. Huo, J. Ma, X. Li, Q. Huang, A light-weight and accurate method of static integer-overflow-to-buffer-overflow vulnerability detection, in: F. Guo, X. Huang, M. Yung (Eds.), *Information Security and Cryptology - 14th International Conference, Inscrypt 2018*, Fuzhou, China, December 14-17, 2018, Revised Selected Papers, Vol. 11449 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 404–423. doi:10.1007/978-3-030-14234-6_22.
- [39] S. Lee, S. Wi, S. Son, Link: Black-box detection of cross-site scripting vulnerabilities using reinforcement learning, in: *Proceedings of the ACM Web Conference 2022*, 2022, pp. 743–754. doi:10.1145/3485447.3512234.
- [40] H. Holm, Lore a red team emulation tool, *IEEE Trans. Dependable Secur. Comput.* 20 (2) (2023) 1596–1608. doi:10.1109/TDSC.2022.3160792.
- [41] T. D. Hoang, C. Park, M. Son, T. Oh, S. Bae, J. Ahn, B. Oh, Y. Kim, Ltesniffer: An open-source lte downlink/uplink eavesdropper, in: *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '23*, Association for Computing Machinery, New York, NY, USA, 2023, pp. 43–48. doi:10.1145/3558482.3590196.
- [42] M. Monshizadeh, P. Naldurg, V. N. Venkatakrishnan, Mace: Detecting privilege escalation vulnerabilities in web applications, in: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 690–701. doi:10.1145/2660267.2660337.
- [43] C. Yucel, A. Lockett, I. Chalkias, D. Mallis, V. Katos, Mait: Malware analysis and intelligence tool, *Information & Security* 50 (1) (2021) 49–65. doi:10.11610/isij.5024.
- [44] P. Johnson, R. Lagerström, M. Ekstedt, A meta-language for threat modelling and attack simulations, in: *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, pp. 1–8. doi:10.1145/3230833.3232799.

- [45] C. Liu, X. Cui, Z. Wang, X. Wang, Y. Feng, X. Li, Malicescript: A novel browser-based intranet threat, in: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), IEEE, 2018, pp. 219–226. doi:10.1109/dsc.2018.00039.
- [46] O. Mjihil, D. S. Kim, A. Haqiq, Masat: Model-based automated security assessment tool for cloud computing, in: 2015 11th International Conference on Information Assurance and Security (IAS), IEEE, 2015, pp. 97–103. doi:10.1109/isias.2015.7492752.
- [47] R. Cayre, V. Nicomette, G. Auriol, E. Alata, M. Kaaniche, G. Marconato, Mirage: towards a metasploit-like framework for iot, in: 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), IEEE, 2019, pp. 261–270. doi:10.1109/issre.2019.00034.
- [48] S. Calzavara, M. Conti, R. Focardi, A. Rabitti, G. Tolomei, Mitch: A machine learning approach to the black-box detection of CSRF vulnerabilities, in: IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019, IEEE, 2019, pp. 528–543. doi:10.1109/EuroSP.2019.00045.
- [49] H. Wei, B. Hassanshahi, G. Bai, P. Krishnan, K. Vorobyov, Moscan: a model-based vulnerability scanner for web single sign-on services, in: C. Cadar, X. Zhang (Eds.), ISSTA '21: 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, Virtual Event, Denmark, July 11-17, 2021, ACM, 2021, pp. 678–681. doi:10.1145/3460319.3469081.
- [50] G. Deng, Z. Zhang, Y. Li, Y. Liu, T. Zhang, Y. Liu, G. Yu, D. Wang, NAUTILUS: automated restful API vulnerability detection, in: J. A. Calandrino, C. Troncoso (Eds.), 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023, USENIX Association, 2023.
- [51] A. Alhuzali, R. Gjomemo, B. Eshete, V. N. Venkatakrishnan, NAVEX: precise and scalable exploit generation for dynamic web applications, in: W. Enck, A. P. Felt (Eds.), 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018, USENIX Association, 2018, pp. 377–392.
- [52] M. Kurth, B. Gras, D. Andriesse, C. Giuffrida, H. Bos, K. Razavi, Netcat: Practical cache attacks from the network, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020. doi:10.1109/sp40000.2020.00082.
- [53] W. Melicher, B. Ur, S. Komanduri, L. Bauer, N. Christin, L. F. Cranor, Fast, lean, and accurate: Modeling password guessability using neural networks, in: D. D. Silva, B. Ford (Eds.), 2017 USENIX Annual Technical Conference, USENIX ATC 2017, Santa Clara, CA, USA, July 12-14, 2017, USENIX Association, 2017.
URL <https://www.usenix.org/conference/atc17/technical-sessions/presentation/melicher>
- [54] W. H. Rankothge, S. M. N. Randeniya, Identification and mitigation tool for cross-site request forgery (csrf), in: 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC), 2020, pp. 1–5. doi:10.1109/R10-HTC49770.2020.9357029.
- [55] A. G. Leal, I. C. Teixeira, Development of a suite of ipv6 vulnerability scanning tests using the TTCN-3 language, in: 2018 International Symposium on Networks, Computers and Communications, ISNCC 2018, Rome, Italy, June 19-21, 2018, IEEE, 2018, pp. 1–6. doi:10.1109/ISNCC.2018.8530888.
- [56] Juliet test suites.
URL <https://samate.nist.gov/SRD/testsuite.php>
- [57] R. Chatterjee, J. Bonneau, A. Juels, T. Ristenpart, Cracking-resistant password vaults using natural language encoders, in: 2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015, IEEE Computer Society, 2015, pp. 481–498. doi:10.1109/SP.2015.36.

- [58] H. Bojinov, E. Bursztein, X. Boyen, D. Boneh, Kamouflage: Loss-resistant password management, in: *Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security*, Athens, Greece, September 20-22, 2010. Proceedings 15, Springer, 2010, pp. 286–302. doi:10.1007/978-3-642-15497-3_18.
- [59] C. Ntantogian, P. Bountakas, D. Antonaropoulos, C. Patsakis, C. Xenakis, Nodexp: Node.js server-side javascript injection vulnerability detection and exploitation, *Journal of Information Security and Applications* 58 (2021) 102752. doi:10.1016/j.jisa.2021.102752.
- [60] N. Koutroumpouchos, G. Lavdanis, E. Veroni, C. Ntantogian, C. Xenakis, Objectmap: detecting insecure object deserialization, in: Y. Manolopoulos, G. A. Papadopoulos, A. Stassopoulou, I. Dionysiou, I. Kyriakides, N. Tsapatsoulis (Eds.), *Proceedings of the 23rd Pan-Hellenic Conference on Informatics, PCI 2019, Nicosia, Cyprus, November 28-30, 2019*, ACM, 2019, pp. 67–72. doi:10.1145/3368640.3368680.
- [61] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, A. Chaabane, Omen: Faster password guessing using an ordered markov enumerator, in: *Engineering Secure Software and Systems: 7th International Symposium, ESSoS 2015, Milan, Italy, March 4-6, 2015*. Proceedings 7, Springer, 2015, pp. 119–132. doi:10.1007/978-3-319-15618-7_10.
- [62] A. Narayanan, V. Shmatikov, Fast dictionary attacks on passwords using time-space tradeoff, in: *Proceedings of the 12th ACM conference on Computer and communications security*, 2005, pp. 364–372. doi:10.1145/1102120.1102168.
- [63] M. Weir, S. Aggarwal, B. De Medeiros, B. Glodek, Password cracking using probabilistic context-free grammars, in: *2009 30th IEEE symposium on security and privacy*, IEEE, 2009, pp. 391–405. doi:10.1109/sp.2009.8.
- [64] P. Kasemsuwan, V. Visoottiviset, Osv: Ospf vulnerability checking tool, in: *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2017, pp. 1–6. doi:10.1109/jcsse.2017.8025919.
- [65] H. Cao, L. Huang, S. Hu, S. Shi, Y. Liu, Owfuzz: Discovering wi-fi flaws in modern devices through over-the-air fuzzing, in: *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '23*, Association for Computing Machinery, New York, NY, USA, 2023, pp. 263–273. doi:10.1145/3558482.3590174.
- [66] B. Hitaj, P. Gasti, G. Ateniese, F. Pérez-Cruz, Passgan: A deep learning approach for password guessing, in: R. H. Deng, V. Gauthier-Umaña, M. Ochoa, M. Yung (Eds.), *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019, Bogota, Colombia, June 5-7, 2019*, Proceedings, Vol. 11464 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 217–237. doi:10.1007/978-3-030-21568-2_11.
- [67] J. Rando, F. Pérez-Cruz, B. Hitaj, Passgpt: Password modeling and (guided) generation with large language models, in: G. Tsudik, M. Conti, K. Liang, G. Smaragdakis (Eds.), *Computer Security - ESORICS 2023 - 28th European Symposium on Research in Computer Security*, The Hague, The Netherlands, September 25-29, 2023, Proceedings, Part IV, Vol. 14347 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 164–183. doi:10.1007/978-3-031-51482-1_9.
- [68] A. M. D. Campi, R. Focardi, F. L. Luccio, The revenge of password crackers: Automated training of password cracking tools, in: V. Atluri, R. D. Pietro, C. D. Jensen, W. Meng (Eds.), *Computer Security - ESORICS 2022 - 27th European Symposium on Research in Computer Security*, Copenhagen, Denmark, September 26-30, 2022, Proceedings, Part II, Vol. 13555 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 317–336. doi:10.1007/978-3-031-17146-8_16.

- [69] R. Luh, M. Temper, S. Tjoa, S. Schrittwieser, H. Janicke, Penquest: a gamified attacker/defender meta model for cyber security assessment and education, *J. Comput. Virol. Hacking Tech.* 16 (1) (2020) 19–61. doi:[10.1007/S11416-019-00342-X](https://doi.org/10.1007/S11416-019-00342-X).
- [70] G. Deng, Y. Liu, V. M. Vilches, P. Liu, Y. Li, Y. Xu, T. Zhang, Y. Liu, M. Pinzger, S. Rass, Pentestgpt: An llm-empowered automatic penetration testing tool, *CoRR abs/2308.06782* (2023). arXiv:[2308.06782](https://arxiv.org/abs/2308.06782), doi:[10.48550/arXiv.2308.06782](https://doi.org/10.48550/arXiv.2308.06782).
- [71] P. J. C. Nunes, J. Fonseca, M. Vieira, phpsafe: A security analysis tool for OOP web application plugins, in: 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2015, Rio de Janeiro, Brazil, June 22-25, 2015, IEEE Computer Society, 2015, pp. 299–306. doi:[10.1109/DSN.2015.16](https://doi.org/10.1109/DSN.2015.16).
- [72] S. Jain, R. Johari, A. Kaur, Pjct: Penetration testing based java code testing tool, in: International Conference on Computing, Communication & Automation, IEEE, 2015, pp. 800–805. doi:[10.1109/ccaa.2015.7148483](https://doi.org/10.1109/ccaa.2015.7148483).
- [73] N. Saccente, J. Dehlinger, L. Deng, S. Chakraborty, Y. Xiong, Project achilles: A prototype tool for static method-level vulnerability detection of java source code using a recurrent neural network, in: 34th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASE Workshops 2019, San Diego, CA, USA, November 11-15, 2019, IEEE, 2019, pp. 114–121. doi:[10.1109/ASEW.2019.00040](https://doi.org/10.1109/ASEW.2019.00040).
- [74] J. Bozic, F. Wotawa, Purity: a planning-based security testing tool, in: 2015 IEEE International Conference on Software Quality, Reliability and Security-Companion, IEEE, 2015, pp. 46–55. doi:[10.1109/qrs-c.2015.19](https://doi.org/10.1109/qrs-c.2015.19).
- [75] M. Muralidharan, K. B. Babu, G. Sujatha, Pyciuti: A python based customizable and flexible cybersecurity utility tool for penetration testing, in: 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), IEEE, 2023, pp. 679–683. doi:[10.1109/icidca56705.2023.10099938](https://doi.org/10.1109/icidca56705.2023.10099938).
- [76] M. Amouei, M. Rezvani, M. Fateh, RAT: reinforcement-learning-driven and adaptive testing for vulnerability discovery in web application firewalls, *IEEE Trans. Dependable Secur. Comput.* 19 (5) (2022) 3371–3386. doi:[10.1109/TDSC.2021.3095417](https://doi.org/10.1109/TDSC.2021.3095417).
- [77] Y. Liu, M. Zhang, W. Meng, Revealer: Detecting and exploiting regular expression denial-of-service vulnerabilities, in: 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, IEEE, 2021, pp. 1468–1484. doi:[10.1109/SP40001.2021.00062](https://doi.org/10.1109/SP40001.2021.00062).
- [78] T. Cloosters, D. Paaßen, J. Wang, O. Draissi, P. Jauernig, E. Stapf, L. Davi, A.-R. Sadeghi, Riscyrop: Automated return-oriented programming attacks on risc-v and arm64, in: Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, RAID '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 30–42. doi:[10.1145/3545948.3545997](https://doi.org/10.1145/3545948.3545997).
- [79] G. Giroto, A. F. Zorzo, Robin: A web security tool, *CoRR abs/2007.06629* (2020). arXiv:[2007.06629](https://arxiv.org/abs/2007.06629), doi:[10.48550/arxiv.2007.06629](https://doi.org/10.48550/arxiv.2007.06629).
- [80] S. Rivera, S. Lagraa, R. State, Rosploit: Cybersecurity tool for ROS, in: 3rd IEEE International Conference on Robotic Computing, IRC 2019, Naples, Italy, February 25-27, 2019, IEEE, 2019, pp. 415–416. doi:[10.1109/IRC.2019.00077](https://doi.org/10.1109/IRC.2019.00077).
- [81] F. Z. Fagroud, H. Toumi, Y. Baddi, S. El Filali, et al., Rt-rct: an online tool for real-time retrieval of connected things, *Bulletin of Electrical Engineering and Informatics* 10 (5) (2021) 2804–2810. doi:[10.11591/eei.v10i5.2901](https://doi.org/10.11591/eei.v10i5.2901).

- [82] P. C. Pale, *Mastering the Nmap Scripting Engine*, Packt Publishing Ltd, 2015.
- [83] Z. Yin, Y. Xu, F. Ma, H. Gao, L. Qiao, Y. Jiang, Scanner++: Enhanced vulnerability detection of web applications with attack intent synchronization, *ACM Trans. Softw. Eng. Methodol.* 32 (1) (Feb. 2023). doi:10.1145/3517036.
- [84] R. Veras, C. Collins, J. Thorpe, On semantic patterns of passwords and their security impact, in: 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014, The Internet Society, 2014. doi:10.14722/ndss.2014.23103.
- [85] M. Shcherbakov, M. Balliu, Serialdetector: Principled and practical exploration of object injection vulnerabilities for the web, in: 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021, The Internet Society, 2021. doi:10.14722/ndss.2021.24550.
- [86] B. Genge, C. Enachescu, Shovat: Shodan-based vulnerability assessment tool for internet-facing services, *Secur. Commun. Networks* 9 (15) (2016) 2696–2714. doi:10.1002/SEC.1262.
- [87] J. Mikulskis, J. K. Becker, S. Gvozdenovic, D. Starobinski, Snout: An extensible iot pen-testing tool, in: L. Cavallaro, J. Kinder, X. Wang, J. Katz (Eds.), *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, ACM, 2019, pp. 2529–2531. doi:10.1145/3319535.3363248.
- [88] N. Antunes, M. Vieira, Soa-scanner: an integrated tool to detect vulnerabilities in service-based infrastructures, in: 2013 IEEE International Conference on Services Computing, IEEE, 2013, pp. 280–287. doi:10.1109/scc.2013.28.
- [89] R. Sommer, J. Amann, S. Hall, Spicy: a unified deep packet inspection framework for safely dissecting all your data, in: S. Schwab, W. K. Robertson, D. Balzarotti (Eds.), *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, December 5-9, 2016*, ACM, 2016, pp. 558–569. doi:10.1145/2991079.2991100.
- [90] Z. Li, X. Yu, D. Wang, Y. Liu, H. Yin, S. He, Supereye: A distributed port scanning system, in: X. Sun, Z. Pan, E. Bertino (Eds.), *Artificial Intelligence and Security - 5th International Conference, ICAIS 2019, New York, NY, USA, July 26-28, 2019, Proceedings, Part IV, Vol. 11635 of Lecture Notes in Computer Science*, Springer, 2019, pp. 46–56. doi:10.1007/978-3-030-24268-8_5.
- [91] H. Holm, T. Sommestad, SVED: scanning, vulnerabilities, exploits and detection, in: J. Brand, M. C. Valenti, A. Akinpelu, B. T. Doshi, B. L. Gorsic (Eds.), *2016 IEEE Military Communications Conference, MILCOM 2016, Baltimore, MD, USA, November 1-3, 2016*, IEEE, 2016, pp. 976–981. doi:10.1109/MILCOM.2016.7795457.
- [92] F. Valenza, E. Karafili, R. V. Steiner, E. C. Lupu, A hybrid threat model for smart systems, *IEEE Trans. Dependable Secur. Comput.* 20 (5) (2023) 4403–4417. doi:10.1109/TDSC.2022.3213577.
- [93] C. Luo, P. Li, W. Meng, Tchecker: Precise static inter-procedural analysis for detecting taint-style vulnerabilities in PHP applications, in: H. Yin, A. Stavrou, C. Cremers, E. Shi (Eds.), *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, ACM, 2022, pp. 2175–2188. doi:10.1145/3548606.3559391.
- [94] O. Olivo, I. Dillig, C. Lin, Detecting and exploiting second order denial-of-service vulnerabilities in web applications, in: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 616–628. doi:10.1145/2810103.2813680.

- [95] E. Bitsikas, S. Khandker, A. Salous, A. Ranganathan, R. Piqueras Jover, C. Pöpper, Ue security reloaded: Developing a 5g standalone user-side security testing framework, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 121–132. doi:10.1145/3558482.3590194.
- [96] A. Bertani, M. Bonelli, L. Binosi, M. Carminati, S. Zanero, M. Polino, Untangle: Aiding global function pointer hijacking for post-cet binary exploitation, in: D. Gruss, F. Maggi, M. Fischer, M. Carminati (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment - 20th International Conference, DIMVA 2023, Hamburg, Germany, July 12-14, 2023, Proceedings, Vol. 13959 of Lecture Notes in Computer Science, Springer, 2023, pp. 256–275. doi:10.1007/978-3-031-35504-2_13.
- [97] K. Vimala, S. Fugkeaw, Vape-bridge: Bridging openvas results for automating metasploit framework, in: 2022 14th International Conference on Knowledge and Smart Technology (KST), IEEE, 2022. doi:10.1109/kst53302.2022.9729085.
- [98] A. Blome, M. Ochoa, K. Li, M. Peroli, M. T. Dashti, Vera: A flexible model-based vulnerability testing tool, in: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, IEEE, 2013, pp. 471–478. doi:10.1109/icst.2013.65.
- [99] S. Kim, S. Woo, H. Lee, H. Oh, VUDDY: A scalable approach for vulnerable code clone discovery, in: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, IEEE Computer Society, 2017, pp. 595–614. doi:10.1109/SP.2017.62.
- [100] P. Kamongi, S. Kotikela, K. Kavi, M. Gomathisankaran, A. Singhal, Vulcan: Vulnerability assessment framework for cloud computing, in: 2013 IEEE 7th international conference on software security and reliability, IEEE, 2013, pp. 218–226. doi:10.1109/sere.2013.31.
- [101] Y. Wu, D. Zou, S. Dou, W. Yang, D. Xu, H. Jin, Vulcnn: An image-inspired scalable vulnerability detection system, in: Proceedings of the 44th International Conference on Software Engineering, ICSE '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 2365–2376. doi:10.1145/3510003.3510229.
- [102] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, Y. Zhong, Vuldeepecker: A deep learning-based system for vulnerability detection, in: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018, The Internet Society, 2018. doi:10.14722/ndss.2018.23158.
- [103] P. Cigoj, B. J. Blazic, An intelligent and automated wcms vulnerability-discovery tool: The current state of the web, IEEE Access 7 (2019) 175466–175473. doi:10.1109/ACCESS.2019.2957573.
- [104] A. Castiglione, F. Palmieri, M. Petraglia, R. Pizzolante, Vulsploit: A module for semi-automatic exploitation of vulnerabilities, in: V. Casola, A. De Benedictis, M. Rak (Eds.), Testing Software and Systems, Springer, Springer International Publishing, Cham, 2020, pp. 89–103. doi:10.1007/978-3-030-64881-7_6.
- [105] Z. Li, D. Zou, S. Xu, H. Jin, H. Qi, J. Hu, Vulpecker: an automated vulnerability detection system based on code similarity analysis, in: S. Schwab, W. K. Robertson, D. Balzarotti (Eds.), Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, December 5-9, 2016, ACM, 2016, pp. 201–213. doi:10.1145/2991079.2991102.
- [106] Z. Đurić, Waptt-web application penetration testing tool, Advances in Electrical and Computer Engineering 14 (1) (2014) 93–102. doi:10.4316/AECE.2014.01015.
- [107] O. van Rooij, M. A. Charalambous, D. Kaizer, M. Papaevripides, E. Athanasopoulos, webfuzz: Grey-box fuzzing for web applications, in: E. Bertino, H. Schulmann, M. Waidner (Eds.), Computer Security - ESORICS 2021 - 26th European Symposium on Research in Computer Security, Darmstadt, Germany,

October 4-8, 2021, Proceedings, Part I, Vol. 12972 of Lecture Notes in Computer Science, Springer, 2021, pp. 152–172. doi:10.1007/978-3-030-88418-5_8.

- [108] W. H. Rankothge, M. Randeniya, V. Samaranayaka, Identification and mitigation tool for sql injection attacks (SQLIA), in: 15th IEEE International Conference on Industrial and Information Systems, ICIIS 2020, Rupnagar, India, November 26-28, 2020, IEEE, 2020, pp. 591–595. doi:10.1109/ICIIS51140.2020.9342703.