A Formal Approach towards Assessing the Effectiveness of Anti-spam Procedures

Guido Schryen
Institute of Business Information Systems, RWTH Aachen University
schryen@winfor.rwth-aachen.de

Abstract

Spam e-mails have become a serious technological and economic problem. So far we have been reasonably able to resist spam e-mails and use the Internet for regular communication by deploying complementary anti-spam approaches. However, if we are to avert the danger of losing the Internet e-mail service as a valuable, free, and worldwide medium of open communication, anti-spam activities should be performed more systematically than is done in current, mainly heuristic, anti-spam approaches. A formal framework within which the modes of spam delivery, anti-spam approaches, and their effectiveness can be investigated, may encourage a shift in methodology and pave the way for new, holistic anti-spam approaches.

This paper presents a model of the Internet e-mail infrastructure as a directed graph and a deterministic finite automaton, and draws on automata theory to formally derive the modes of spam delivery possible. Finally the effectiveness of anti-spam approaches in terms of coverage of spamming modes is assessed.

1. Introduction

Spam e-mails have become a serious technological and economic problem. So far we have been reasonably able to resist spam e-mails, although statistics show a proportion of spam higher than 60% [11, 12]. The availability of the Internet e-mail system for regular e-mail communication is currently ensured by complementary anti-spam approaches, mainly by blocking and filtering procedures. However, if we are to avert the danger of losing the Internet e-mail service as a valuable, free, and worldwide medium of open communication, anti-spam activities should be performed more systematically than is done in current, mainly heuristic, anti-spam approaches. A formal framework within which the modes of spam delivery, anti-spam approaches, and their effectiveness can be investigated, may encourage a shift in methodology and pave the way for new, holistic anti-spam approaches. In Section 2 the Internet e-mail infrastructure is modeled as a directed graph and a deterministic finite automaton and the appropriateness of the model is proved. In section 3 all possible modes of sending (spam) e-mails are derived and presented as regular expressions. These (technically-oriented) expressions are then grouped into categories according to types of organization participating in e-mail delivery. The effectiveness of today's most important anti-spam approaches is assessed in section 4 by matching them with the modes of spamming. Section 5 summarizes the results presented in this manuscript and outlines future work.

2. Model of the Internet e-mail infrastructure

The Internet e-mail infrastructure can be modeled as a directed graph G which is defined in the first subsection. In the second subsection it is shown that all types of e-mail deliveries can be described with a set of (directed) paths in G. As each option to send e-mails at the same time represents an option to send spam e-mails the set of spamming options and the set of mailing options are regarded to be equal.

2.1 Definition

Since the Internet e-mail network infrastructure which the graph is intended to represent is dynamic, it is not meaningful to model each concrete e-mail node. The different *types* of Internet e-mail nodes on the other hand are static, and it is these which can serve the purpose. An e-mail node is here defined as a software unit which is involved in the Internet e-mail delivery process and works on the TCP/IP application layer. Consideration of software working exclusively on lower levels such as routers and bridges is beyond the scope of this work, as are options to send an e-mail without any SMTP communication with an e-mail node of the recipient's organization. However, this does not seem to be an important restriction, as almost all e-mail users receive their e-mails from a server that is (directly or indirectly) SMTP connected to the Internet.

Let G={V,E,c} be a directed graph with vertex set V and edge set E, and let c: E→L be a total function on E where L denotes a set of (protocol) labels. First the structure of the graph is presented graphically (see figure 1) and formally. Its semantics is then explained in more detail.

The set of vertices can be depicted as the disjoint union of five vertex sets $V_1,\ldots,V_5$. Each of these sets is attached to one of the organizational units participating in e-mail delivery: *sender*, *sending organization* or *e-mail (service) provider*, *Internet*, *receiving organization*, and *recipient*. In cases where the recipient does not use a receiving organization as e-mail service provider (ESP) but runs his own e-mail receiving and processing environment the organizational units *receiving organization* and *recipient* merge. This, however, does not affect the structure of the graph, which retains its general validity. The set of vertices be $V=V_1\cup\ldots\cup V_5$ with

$V_1 = \{MTA_{send}, MUA_{send}, OtherAgent_{send}\}$ set of vertices attached to *sender*,

$V_2 = \{MTA^{inc}_{sendOrg}, MTA_{sendOrg}, WebServ_{sendOrg}\}$ set of vertices attached to *sending organization*,

$V_3 = \{SMTP - \mathrm{Re}\,lay, GW_{SMTP,B}, GW_{A,SMTP}, GW_{A,B}\}$

set of vertices attached to *Internet*,

$V_4 = \{MTA^{inc}_{recOrg}, MTA_{recOrg}, MDA_{recOrg},$
$MailServ_{recOrg}, WebServ_{recOrg}\}$

set of vertices attached to *receiving organization*, and

$V_5 = \{MUA_{recOrg}\}$ set of vertices attached to *recipient.*

The set of (protocol) labels be

$L = \{SMTP, SMTP^*, \overline{SMTP^*}, HTTP(S), INT, MAP\}$.

E and c are not defined formally here, but shown graphically in figure 1.

Each vertex corresponds to a type of e-mail node. An edge $e=(v_1,v_2)$ with a label $c(e) \in L$ attached exists if and only if the Internet e-mail infrastructure allows e-mail flow between the corresponding node types; c(e) denotes the set of feasible protocols.

The set SMTP contains SMTP (as a protocol) extended by all IANA-registered SMTP service extensions, also referred to as ESMTP, such as SMTP Service Extension for Authentication (RFC 2554), Deliver By SMTP Service Extension (RFC 2852), SMTP Service Extension for Returning Enhanced Error (RFC 2034), and SMTP Service Extension for Secure SMTP over Transport Layer Security (RFC 3207); see www.iana.org/assignments/mail-parameters for a list of SMTP service extensions.

The set $SMTP^*$ contains the set SMTP and all SMTP extensions specified for e-mail submission from a Mail User Agent (MUA) to an e-mail node with an SMTP incoming interface. This e-mail node can be a Mail Transfer Agent (MTA) as specified in RFC 2821 or a Message Submission Agent (MSA) as specified in RFC 2476. With reference to the latter $MTA^{inc}_{sendOrg}$ can alternatively be denoted as $MSA_{sendOrg}$ and MTA^{inc}_{recOrg} as MSA_{recOrg},respectively. Port 587 is reserved for e-mail message submission. However, while most e-mail clients and servers can be configured to use port 587 instead of 25, there are cases where this is not possible or convenient. A site may use port 25 even for message submission. Using an MSA numerous methods can be applied to ensure that only authorized users can submit messages. These methods include authenticated SMTP, IP address restrictions, secure IP, and prior Post Office Protocol (POP) authentication, where clients are required to authenticate their identity before an SMTP submission session ("SMTP after POP"). $\overline{SMTP^*}$ is the union of three sets of protocols. The first contains all Internet application protocols excepting SMTP*, the second, all proprietary application protocols used on the Internet: this inclusion takes tunneling procedures into account. The third set, since use of application protocols is not mandatory for the exchange of data in a network, are all Internet protocols on the transport and network layer of the Department of Defense (DoD) model, such as TCP and IP. MAP is the set of all e-mail access protocols used to transfer e-mails from the recipient's e-mail server to his MUA. Internet Message Access Protocol (IMAP) version 4 (RFC 1730) and POP version 3 (RFC 1939) are here among the protocols most deployed. The set HTTP(S) contains the protocols HTTP (RFC 2616) as well as its secure versions "HTTP over SSL" [3] and "HTTP over TLS" (RFC2818). Finally, the set INT denotes protocols and procedures for internal e-mail delivery, i. e. inside the receiving organization, such as getting e-mails from an internal MTA and storing them into the users' e-mail boxes.

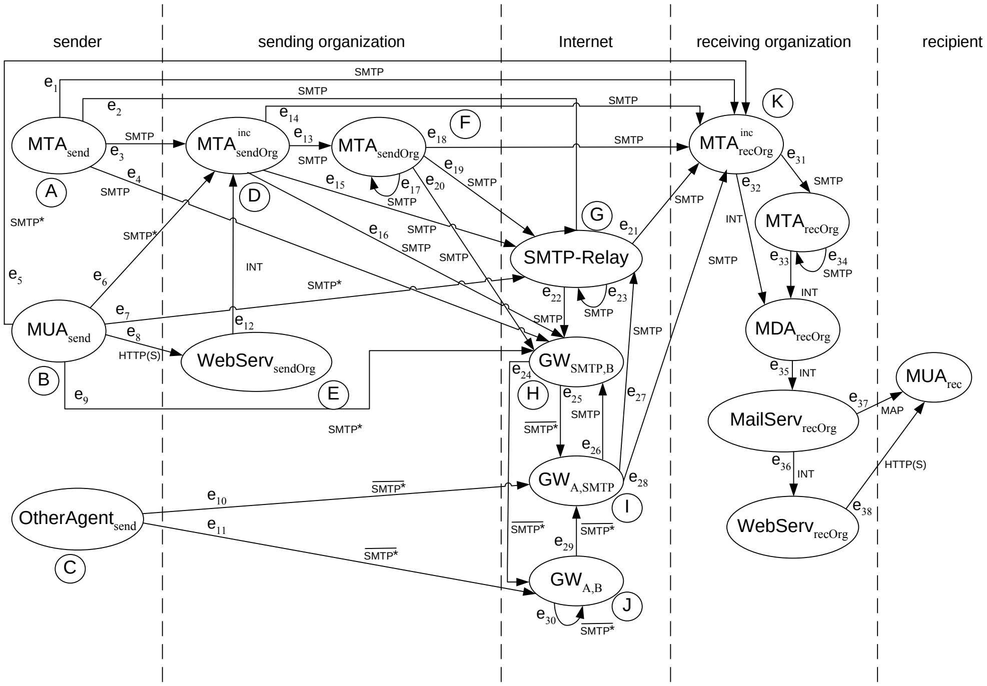

Figure 1: Internet e-mail infrastructure as directed graph

2.2 Appropriateness

The meaning of G in the context of e-mail delivery is given by the fact that the different types of e-mail delivery can be described by a set of specific (directed) paths in G. Before this issue is formally addressed in proposition 1, it will be shown that each type of e-mail node corresponds to a vertex in G:

Lemma 1 Let T be the set of types of e-mail nodes involved in e-mail delivery. Then there is a bijection between T and V.

The proof of lemma 1 is given in the Appendix. It should be noted that the e-mail nodes are logical nodes representing software, some of which may fall in one physical node in a particular instance of e-mail delivery (e.g. MTA^{inc}_{recOrg} and MDA_{recOrg}).

Given Lemma 1, we can now proof the following **Proposition 1** Let $s = (s_1, ..., s_k), k \in N^{\geq 2}$, be a sequence of types of e-mail nodes involved in a complete[1] e-mail delivery and S the set of all feasible sequences, i.e. all types of e-mail delivery. Let p be a (directed) path in G between two vertices $v_{start} \in V_{start} = V_1 \cup \{MTA_{sendOrg}, MTA^{inc}_{sendOrg}\}$ and $v_{end} \in V_{end} = \{MailServ_{recOrg}, MTA_{recOrg}, MTA^{inc}_{recOrg}\}$ and P the set of all these paths. Then there is a bijection f between S and P.

The proof of proposition 1 is given in the Appendix. Proposition 1 provides the basis for formally representing modes of sending an (spam) e-mail. System security violations, caused by, e.g., viruses, Trojan horses, or worms, which may affect e-mail nodes belonging to a sending or receiving organization are not considered here. Possible violations in users' systems, which are more relevant anyway, are, however, included.

Each of the possibilities of sending one e-mail also permits the sending of a number of e-mails, as is the case with the millions of e-mails spammers send. So, to identify all the possibilities of sending spam e-mails we must consider the set of possibilities of sending one e-mail. Since these are represented by the set of all paths in G from V_{start} to V_{end}, we are now provided with a formal approach to the modes of spamming available.

3. Model and classification of spamming options

The graph G will serve as a basis for deriving all modes of spamming. It provides a formal framework within which the effectiveness of (present and future)

[1] By "complete" I mean that the e-mail has reached the recipient's e-mail box on his e-mail server or the MTA of the receiving organization which applies a forwarding rule or rejects the message. That is, forwarding an e-mail and sending a bounce e-mail starts a new sequence. Furthermore, only those e-mail deliveries are regarded which are either initiated by a sender's client or, in case of bouncing or forwarding e-mails, by an ESP's MTA.

technological anti-spam measures can be theoretically analyzed. It also shows all possible spam delivery routes which any holistic anti-spam measures would need to cover.

The modes of spamming are formally presented in the first subsection. A classification of them follows in the second subsection.

3.1 Modes of spamming

For the sake of simplification, we consider all paths from V_{start} to $MailServ_{recOrg}$, denoted as P'. For simplification, all paths from V_{start} to $MailServ_{recOrg}$ are regarded, denoted with P'. Now that P' is identified, we obtain, by simply extensions, all the paths from V_{start} to V_{end}, i.e. P. P' is arrived at applying some basic ideas from automata theory: the graph G is transformed into a Deterministic Finite Automaton (DFA) $A=(S,\Sigma,\delta,Start,F)$ where S is a finite set of states, Σ is an alphabet, *Start* is the initial state, $F \subseteq S$ is the set of final states, and δ is a function from $S \times \Sigma$ to S. This automaton recognizes a language that (bijectively) corresponds to P', such that $w=(w_{1,\ldots,}\ w_n)\in L(A) \Leftrightarrow (w_1,\ldots,w_n)\in P$, where L(A) is the language recognized by A. The construction is self-evident and can be described informally as follows. The set of states S corresponds to the nodes of G extended by an artificial state *Start* which serves as the initial state, $\sum$ corresponds to the nodes of G, as well. An edge (v_1,v_2) means that the transition function δ includes $\delta(s_1,s_2)=s_2$, i.e. state s_2 is reached if and only if the symbol s_2 is "read" by A. In order to account for the starting node, δ also needs to include $\delta(Start,s_2)=s_2$ with s_2 being a state corresponding to any node of the set of starting nodes V_{start}. F only contains the state corresponding to the node MTA^{inc}_{recOrg}.

The language recognized by the DFA A – and thus P' – can be described with a regular expression because of the equivalence between DFAs and regular expressions. For simplicity, the states are labeled with capital letters which are assigned to the corresponding nodes (see figure 1); elements of $\sum$ are set in lower case letters. Given two regular expressions r_1 and r_2, ~ denotes the relationship between r_1 and r_2 with $r_1 \sim r_2 :\Leftrightarrow L(r_1)=L(r_2)$; Λ be the regular expression with $L(\Lambda)=\varepsilon$. Using the edges of G we get L(A)=L(Start) with

$$\begin{array}{lll}
Start & \sim aA \vee bB \vee cC \vee dD \vee fF & (1)\\
A & \sim kK \vee gG \vee dD \vee hH & (2)\\
B & \sim dD \vee gG \vee eE \vee hH & (3)\\
C & \sim iI \ \vee jJ & (4)\\
D & \sim kK \vee \ fF \vee \ gG \vee hH & (5)\\
E & \sim dD & (6)\\
F & \sim kK \vee gG \vee \ fF \vee hH & (7)\\
G & \sim kK \vee gG \vee \ hH & (8)\\
H & \sim jJ \ \vee iI & (9)\\
I & \sim hH \vee gG \vee kK & (10)\\
J & \sim iI \ \vee jJ & (11)\\
K & \sim \Lambda & (12)
\end{array}$$

Let α, β, γ be regular expressions, then recursive relationships can be dissolved using the rule

$$\frac{\alpha \sim \beta\alpha \vee \gamma,\ \varepsilon \notin L(\beta)}{\alpha \sim \beta^*\gamma} \qquad (13)$$

(1) can be solved by application of simple substitutions and multiple application of rule (13), yielding:

$$\begin{aligned}
Start \sim\ & ak \vee agg^*k \vee agg^*hH \vee ad\,(k \vee ff^*k \vee \\
& ff^*gg^* \vee ff^*gg^*hH \vee ff^*hH \vee gg^*k \vee \\
& gg^*hH \vee hH) \vee ahH \vee (bd \vee bed)\,(k \vee ff^*k \\
& \vee\ ff^*gg^*k \vee ff^*gg^*hH \vee ff^*hH \vee gg^*k \vee \\
& gg^*hH \vee hH) \vee bgg^*k \vee bgg^*hH \vee bhH \vee \\
& cihH \vee cigGH \vee cigg^*k \vee cik \vee cjj^*ihH \vee \\
& cjj^*igg^*H \vee cjj^*igg^*k \vee cjj^*ik \vee d\,(\ k\ \vee \\
& ff^*k \vee ff^*gg^*k \vee ff^*gg^*hH \vee ff^*hH \vee gg^*k \vee \\
& gg^*hH \vee hH) \vee ff^*k \vee ff^*gg^*k \vee ff^*gg^*hH \vee \\
& ff^*hH \qquad (14)
\end{aligned}$$

with

$$\begin{aligned}
H \sim\ & (jj^*I\,(\ h \vee gg^*h\ ) \vee I\,(\ h \vee gg^*h\ ))^* \\
& (jj^*igg^*k \vee jj^*ik \vee igg^*k \vee ik) \qquad (15)
\end{aligned}$$

As (14) shows, the infrastructural modes of sending a(n) (spam) e-mail are numerous and should therefore be grouped appropriately.

3.2 Spamming categories

Grouping can be done by subsuming those options in which the same types of organizational units participate. As the *receiving organization* participates in each complete e-mail delivery process and the *recipient* does not affect the delivery process these two units are not integrated into the classification. Accordingly, the set of spamming categories results from the integration of a local sender, an ESP and the Internet (application level infrastructure) which provides eights options. The options (groups, categories) are defined and explained in table 1. The cases in which neither a local sender nor an ESP is involved are only theoretical ones. The model requires a sender or sending organization to appear before any Internet e-mail node: an e-mail either starts from a node residing on ESP's side or on local side included those cases in which computers were

No.	Scenario	Sender	Sending organization	Internet
-	-			
-	-			x
I	Provider itself spams or its MTAs were corrupted; direct connection to MTA_{recOrg}		x	
II	Provider itself spams or its MTAs were corrupted; use of intermediate Internet nodes like relays		x	x
III	Spammer uses local client; direct connection to MTA_{recOrg} (via dial-in or LAN connection)	x		
IV	Spammer uses local client; use of intermediate Internet nodes like relays (via dial-in or LAN connection)	x		x
V	Spammer uses local client; use of e-mail provider (via dial-in or LAN connection)	x	x	
VI	Spammer uses local client; use of e-mail provider (via dial-in or LAN connection) which uses intermediate Internet nodes like relays	x	x	x

Table 1: Spamming categories

infected or controlled remotely. Scenarios I and II refer to ESPs being corrupt or featuring corrupt MTAs. As the number of ESPs and their MTAs is significantly lower than the number of people these spamming options should be addressable effectively. In all other scenarios the spammer uses a local client which seems much more likely and easier for the spammer. Scenario III describes those cases in which a spammer does not use an ESP but, of course, an Internet service provider (ISP) operating not higher than on transport layer, i.e. simple forwarding Transmission Control Protocol (TCP) packets. As in this scenario the spammer connects directly with an MTA of the receiving organization he is restricted to the e-mail ports implemented there. Usually, port 25 or 587 is used. This makes it easy for ISPs to prevent most spam e-mails sent that way by simply blocking TCP packets to these ports. Scenario 4 is much harder to address as spammers use Internet nodes including gateways and thus are not restricted to ports 25 and 587. In Scenario V the spammer uses an ESP to send spam e-mails thereby abusing the e-mail service. Even if the number of e-mails per day and account is restricted it remains challenging to prevent the spammer from setting up new accounts automatically. Scenario VI seems quite unlikely. It refers to a spammer who uses an ESP who forwards e-mails by sending them to intermediate nodes in the Internet. This might occur if the ESP supports spamming activities of customers. Grouping the spamming options of (14) according to the six categories shown in table 1 yields after some transformations:

$$
\begin{aligned}
\text{Start} \quad \sim \quad & \\
& \text{I: } d\,(k \vee ff^{*}k) \vee ff^{*}k \vee \\
& \text{II: } (d \vee \Lambda)\,(ff^{*}gg^{*}k \vee ff^{*}gg^{*}hH \vee ff^{*}hH) \vee \\
& \quad d\,(gg^{*}k \vee gg^{*}hH \vee hH) \vee \\
& \text{III: } ak \vee \\
& \text{IV: } (a \vee b)\,(gg^{*}k \vee gg^{*}hH \vee hH) \vee cj^{*}i(hH \\
& \quad \vee gg^{*}H \vee gg^{*}k \vee k) \vee \\
& \text{V: } (ad \vee bd \vee bed)\,(k \vee ff^{*}k) \vee \\
& \text{VI: } (ad \vee bd \vee bed)\,(ff^{*}gg^{*}k \vee ff^{*}gg^{*}hH \vee \\
& \quad ff^{*}hH \vee gg^{*}k \vee gg^{*}hH \vee hH)
\end{aligned}
$$

with

$$
\begin{aligned}
H \quad & \sim (jj^{*}I\,(h \vee gg^{*}h) \vee I\,(h \vee gg^{*}h))^{*} \\
& (jj^{*}igg^{*}k \vee jj^{*}ik \vee igg^{*}k \vee ik)
\end{aligned}
$$

4. Effectiveness of anti-spam approaches

The effectiveness of the most discussed and deployed anti-spam approaches is assessed on a theoretical and qualitative level, any quantitative and empirical results are out of scope of this manuscript.

Table 2 shows which spamming options can be effectively addressed by which anti-spam approach. An "x" indicates an effective coverage, a blank entry means that no effective coverage is possible.

Some anti-spam approaches have general impacts on spam, i.e. their effectiveness do not depend on the way the spam was sent. Filter, blocking mechanisms basing on gray lists, and address obscuring techniques belong to these approaches and are thus not included in table 2. Filter (programs) heuristically classify incoming e-mails into ham and spam e-mails. They prescind from the spamming option and solely focus on the e-mail document which has already received by the organization's incoming mail server MTA_{recOrg}^{inc} or even by the user's mail client MUA_{rec}.

Spam filters can be categorized according to the type of filtering method (e.g. statistical filters [4] or neural network-based filters – version 3 of the open-source spam filter software SpamAssassin uses a neural network) and according to the e-mail parts inspected by the filter: the header and/or the body can be filtered. All filters suffer from the same drawbacks:

- As they are heuristic approaches they might classify incorrectly: spam e-mails might pass the filter ("false-negatives"), ham e-mails might be filtered and probably even deleted ("false-positives").
- Spammers are forced to send even more e-mails to compensate filters' success.

			Anti-spam approaches					
Scenario	Regular expression	Nodes involved	IP blocking	Limited # of e-mails	Blocking port 25	Digital signature	LMAP	sTLD
I	d (k ∨ ff*k) ∨ ff*k	MTAs of provider	x					x
II	(d ∨ Λ)(ff*gg*k) ∨ dgg*k	MTA of provider, then relay(s)						x
II	(d ∨ Λ) (ff*gg*hH) ∨ dgg*hH	MTA of provider, then relay(s) and gateway(s)						
II	(d ∨ Λ) (ff*hH) ∨ dhH	MTA of provider, then at least gateway(s)						
III	ak	Local MTA			x		x	x
IV	(a ∨ b) gg*k	Local MTA or MUA, then relay(s)			x	x	x	x
IV	(a ∨ b) (gg*hH ∨ hH)	Local MTA or MUA, then relay(s) and gateway(s)						
IV	cj*i (hH ∨ gg*H ∨ gg*k ∨ k)	Local agent other than MTA or MUA, then at least gateway(s)						
V	(ad ∨ bd ∨ bed) (k ∨ ff*k)	Local MTA or MUA, then MTA(s) of provider						
VI	(ad ∨ bd ∨ bed) (ff*gg*k ∨ gg*k)	Local MTA or MUA, then MTA(s) of provider, then relay(s)					x	x
VI	(ad ∨ bd ∨ bed) (ff*hH ∨ hH)	Local MTA or MUA, then MTA(s) of provider, then at least gateway(s)						
VI	(ad ∨ bd ∨ bed) (ff*gg*hH ∨ gg*hH)	Local MTA or MUA, then MTA(s) of provider, then relay(s) and gateway(s)						

Table 2: Effectiveness of anti-spam approaches

- The goal of filters is not to prevent spam but to detect it. But when spam e-mails are detected on recipient's site resources (e.g. band width, storage capacity, CPU time reserved for filter software) already have been consumed.
- Spammers continue upgrading their skills and it is doubtful if filters can be created which are effective in the long-term.

Gray lists [6] are used for temporarily blocking incoming e-mails: They take advantage of the fact that spamming e-mail MTAs often don't implement the standardized "resume feature" according to which a once rejected e-mail is sent again after some minutes. If the same e-mail arrives a second time in a specific time window it may pass. For example, the RWTH Aachen University has implemented a gray list system that stores the IP of the sending host, the sender address, and the recipient address. However, once spammer can access a pool of valid e-mail addresses which might have been stolen, bought, or harvested, they certainly will implement the resume feature thus bypassing the gray list approach.

Resource-based approaches to prevent spam have been proposed in [1] and [2]: before an e-mail can be sent a function has to be calculated which is time- or memory-consuming, respectively. These approaches faces at least two problems: (1) How can solicited bulk e-mail been sent? (2) How can spammers be stopped from allocating sufficient resources?

Address obscuring techniques (AOTs) generally aim at either hiding e-mail addresses or restricting them to a limited use. An example of the former type is the proposal of Hall [5] according to which a user can have an arbitrary number of unguessable addresses (channels). An approach of the latter group is Ioannidis' proposal of a single-purpose address (SPA) which encapsulates a policy in the e-mail address [8]. This policy defines the acceptable use of the address, e.g. the sender allowed to direct an e-email to this address. SPAs are cryptographically protected to shield them from tampering. AOTs only refer to machine-to-person e-mail communication as the addresses are difficult to handle for humans. A second restriction of AOTs' usefulness refers to the complicated first-time person-to-person contacting as a recipient cannot anticipate all regular e-mail connections.

Blocking e-mails is a broadly used mechanism where an e-mail passes or is rejected due to the IP address of the sending node. IP addresses of nodes known to send spam e-mails are listed on local and/or public black lists. For example, ordb.org provides a list of open relays, the Spamhaus Project (www.spamhaus.org) provides a realtime blacklist of IP addresses of verified spam sources as well as an Exploits Block List which is a database of IP addresses of illegal 3rd party exploits, including open proxies, worms/viruses with built-in spam engines, and other types of Trojan-horse exploits. Black lists are not effective in first-time spamming, where nodes send spam e-mails for the first time. As spammers tend to change their IP addresses frequently by switching to another ISP or misusing exploits on unsuspicious 3rd party nodes black lists can only stop corrupt ESPs, as they usually do not change their IPs frequently. Blocking whole IP ranges – e.g. belonging to ISPs or even countries known to host spammers – can address "repeated spamming" but easily leads to a digital divide.

A straightforward option easy to implement is to restrict the number of e-mails which can be sent from a user's account. This approach is only applicable when an ESP is involved, but even then the automatic set-up of an arbitrary number of new e-mail account is possible. Some ESPs apply CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) procedures which provide a numbers or a word which has to be retyped. However, current implementations are not effective as they are insecure, e.g. [10] presents a procedure which is able to detect the content of 92% of all pictures created by the Yahoo CAPTCHA process. Another attack on visual CAPTCHA processes proceeds as follows: the spammer puts the ESP's picture on his own web site and asks users to read the text and enter it in a text field manually to gain access to adult information. The spammer puts the text field's content in the corresponding text field of the ESP's form. All this can be done automatically.

Blocking all (uncontrolled) TCP traffic on port 25 is an option for ISP which is easy to implement and prevents spammers' and exploited computers to send spam e-mails when port 25 is used. Activities of agents using other ports and connecting with gateways are not covered. It must be noted, that this approach means to exclude the operation of an MTA running on a user's or company's computer.

Environments enabling the recipient to authenticate the sender or at least the sending organization have been proposed. Public Key Cryptography provides the mathematical and algorithmic instruments for digital signing documents, Public Key Infrastructures (PKIs) feature the organizational framework. As most e-mail user's do not have a personal key pair, current implementations aim at authenticating at least organizations and (second level) domains. The procedure is straightforward: the sending organization signs an e-mail with its private key, once the sending organization's domain can be verified by applying the public key, it can be compared to the domain used by the sender in the From: field of the message to detect forgeries. An example is Yahoo's "DomainKeys" [14] which is the currently most discussed one. An PKI-based approach presumes that the sending organization is not corrupt and that its MTAs do not suffer from any exploit. A problem with this approach arises when spammers get a key pair for a domain which is used by them temporarily just for spamming purpose. PKI-based approaches are helpful when exploits of unsuspicious computers, e.g. spamming machines residing on the user's computer and being remotely controllable due to a Trojan horse, try to send spam e-mails bypassing the MTAs and the signing software of the sender's organization. Then, no authentication on recipient's side is possible or successful. Quite a serious challenge appears when spamming software on infected computers reads misuses the user's account information for outgoing e-mails including a password. The sending organization can not differ between user's regular e-mails and e-mails sent by a local spamming machine.

Another authentication-based approach is the use of an Lightweight MTA Authentication Protocol (LMAP) [9]. The LMAP family subsumes a set of DNS-based approaches where it is checked whether a message that claims to be from buffy@sunnydale.com was actually sent from an MTA of the sunnydale.com organization. If not, the e-mail is a forgery or an intermediate MTA was used as external e-mail relay. Reverse MX (RMX) Designated Mailers Protocol (DMP), Sender Policy Framework (SPF), and SenderID belong to most discussed approaches, however, no standardization was made and the IETF working group MARID was dissolved in 2004. LMAP approaches effectively address the unauthorized use of relays and gateways but feature similar weaknesses as the PKI-based ones.

ICANN has presented an organizational and technological framework proposed by Spamhaus [7] in which a new sponsored top level domain (sTLD), e.g., .mail, is used. The proposed sTLD will be limited for use by the registrant only during the process of sending e-mail. A registrant must have already registered for a domain key, e.g. icann.org, to get the domain key.sTLD, e.g. icann.org.mail. More demands to be met by a registrant were proposed, e.g. validated whois information must be available, the key domain must have been already registered for at least 6 months, and an appropriate technological anti-spam protection has to be implemented. The registrant must inform the central (sponsoring) organization (SO) of the IPs and hostnames of the sending mail server. Then, the SO enters A records in the DNS for the new domain which enables recipient MTAs to use an LMAP or a PKI-based authentication. Abuse messages for *key.sTLD* are received by a central organization (sponsoring organization) providing a control mechanism. Although the framework promises to be effective against many spamming procedures one problem remains: How can an appropriate technological anti-spam protection be achieved? The framework does not cover the case when zombie PCs – often exploited by spammers with Trojan horses – send spam e-mails.

Summing up the effectiveness of anti-spam approaches which is illustrated in table 2 it must be stressed that no anti-spam approach is currently capable of effectively stopping spammers activities which aim at exploiting the ESPs' infrastructure (scenario V). Main problems are still the spammers' option to set-up and then misuse e-mail accounts automatically and spammers activities regarding the exploits of unsuspicious computers. The latter trouble becomes worse as spammers use "botnets" which are networks of compromised machines that can be remotely controlled by an attacker [13].

5. Summary and outlook

The modes of spam delivery can be described formally with 12 regular expressions. Comparison of what are currently the most important anti-spam approaches with the modes of spamming show that exploitation of PCs and ESP infrastructures are not effectively dealt with. Moreover, most activities focus on the detection rather than the prevention of spam. But this may in fact be counterproductive because spammers are encouraged to send even more e-mails in order to compensate for losses from detection.

Anti-spam activities should be performed more systematically than is done in current, mainly heuristic, anti-spam approaches. Models and formal procedures are possibly an adequate way to assess the effectiveness of anti-spam approaches and to develop new, holistic approaches which would focus on the prevention of spam e-mails.

6. Acknowledgement

Many thanks to Katrin Ungeheuer who made numerous suggestions for improvement.

7. Appendix

Proof of lemma 1 Technical e-mail nodes can be assigned to the organizational unit that acts as sender of an e-mail, the sender's organization (sender's ESP), the recipient, the recipient's organization (recipient's ESP), and the Internet subsuming all other organizational units. On application layer the sender can use an MTA, an MUA as defined in RFC 2821, or any other agent. These nodes correspond to the nodes in G denoted as MTA_{send}, MUA_{send}, and $OtherAgent_{send}$, respectively. If a sending organization participates in e-mail delivery, it accepts incoming e-mails with an SMTP-based MTA, denoted as $MTA_{sendOrg}^{inc}$ in G. Alternatively, e-mails may be sent to a sending organization by way of the web environment, meaning that all e-mails are passed to an internal MTA by a receiving web e-mail server, denoted as $WebServ_{sendOrg}$. A sending organization may make internal SMTP-based delivery using two or more consecutive MTAs, denoted as $MTA_{sendOrg}$. No other e-mail nodes are generally used by ESPs, exceptions being proprietary e-mail nodes. However, since any internal non-standard processing of an (outgoing) ESP is required by interorganizational e-mail delivery agreements to be completed with an MTA, such e-mail nodes are of no relevance in the overall e-mail delivery chain and can be ignored. Receiving organizations as a rule take only SMTP-based e-mail deliveries, and although exceptions do exist, they are so uncommon as to be likewise negligible. As in the case of the sending organization, the MTA responsible for incoming SMTP connections, denoted as MTA_{recOrg}^{inc}, may be followed by two or more consecutive MTAs, denoted as MTA_{recOrg}, before the Mail Delivery Agent (MDA), denoted as MDA_{recOrg}, deposits the message in a "message store" (mail spool) which a mail server, denoted as $MailServ_{recOrg}$, accesses in order to deliver it to the recipient's MUA directly, denoted as MUA_{recOrg}, or via a web-based e-mail server, denoted as $WebServ_{recOrg}$. E-mails terminating in a system other than SMTP require the existence of a mail gateway, but, like the analogous situation at the sending organization's site, this issue is beyond the scope of this model. When the Local Mail Transfer Protocol (LMTP, RFC2033) is used to relay messages to the MDA, the MDA is termed as Local Delivery Agent (LDA). Before an e-mail passes the first MTA of the receiving organization it may be relayed by an intermediate SMTP relay which accepts an e-mail sent by a node residing on the sender's site or on the sending organization's site and transfers it to another e-mail node. This includes the scenario where an e-mail is forwarded to another e-mail node because of a mailbox-specific forwarding rule. The SMTP relay represents an intermediate Internet e-mail node using SMTP both at the incoming and the outgoing interface. When other interfaces are used, three further intermediate types are possible which are used, e.g., for SMTP tunneling and known as gateways: $GW_{SMTP,B}$ nodes accept SMTP e-mails and transfer e-mails with a protocol other than SMTP; $GW_{A,SMTP}$ performs the inverse process at incoming and outgoing interfaces; nodes of type $GW_{A,B}$ use SMTP neither for incoming nor for outgoing messages where A and B can be the same protocol (when A=B we usually talk about a proxy, but for simplicity we subsume this under gateway). No other (important) types of e-mail node are involved in Internet e-mail deliveries. Different types are mapped onto different vertices (injection), and no vertices other than those mentioned exist (surjection), i.e. there is a bijection between T and V. ■

Proof of proposition 1 The proof is in three steps. (1) A function $f:S \rightarrow P$ is defined, (2) injection of f is shown, (3) surjection of f is shown.

(1) Defining f is straightforward: Given a sequence of e-mail nodes $s = (s_1,...,s_k) \in S, |s| := k \in N^{\geq 2}$ then s is associated with $f(s) = (p_1,..., p_k)$, where p_i is the corresponding vertex to s_i for all i=1,…,k; s(i) exists due to the bijection of lemma 1. It still has to be shown that the image of f is a subset of P, i.e. that f(s) is a (directed) path in G between V_{start} and V_{end} for all $s \in S$. This is true if and only if for all $s \in S$, k=|s| holds: $(p_i,p_{i+1}) \in E$ for all i=1,…,(k-1), i.e. G contains directed edges between all vertices which correspond to two e-mail nodes appearing consecutively in any sequence $s \in S$. It is easy to see that the latter condition is fulfilled. To prove that the first condition is fulfilled it is sufficient to show that the design criteria for edges in G (see above) are met: an edge $e=(v_1,v_2)$ exists if and only if the Internet e-mail infrastructure allows e-mail flow between the corresponding node types. To this end, each node v of G is explored with reference to the edges incident upon v:

MTA_{send}. With a local MTA on the user's side only SMTP connections are possible. SMTP connections can be established with an ESP's incoming MTA (e_3), with Internet nodes accepting SMTP connections, namely an SMTP relay (e_2) and a gateway (e_4), or with an MTA of the receiving organization or recipient (e_1). Other connections are not possible.

MUA_{send}. If the e-mail sender operates an MUA, he can connect either with all nodes featuring an SMTP interface for incoming connections or a with a web server of the sending organization. HTTP(S)-based connections with other nodes are covered by the node $OtherAgent_{send}$. In the former case, the MUA can connect to the same nodes as the MTA_{send} (e_5, e_6, e_7, e_9). However, as an MUA is involved the set of protocols has to be extended to $SMTP^*$. If a web server is connected, then either HTTP or the secure version HTTPS may be used. Other connections are not possible and are not modeled.

$OtherAgent_{send}$. Other agents are defined as agents that use connections other than SMTP-based ones ($\overline{SMTP^*}$). ESPs and organizations today generally accept only SMTP-based e-mail connections, such that they can only connect to gateways in the Internet (e_{10}, e_{11}) as modeled.

$WebServ_{sendOrg}$. A web server of an ESP sends its e-mails to an internal MTA (e_{12}). Connections with other nodes generally do not exist.

$MTA^{inc}_{sendOrg}$. The MTA responsible for incoming messages most commonly SMTP-connects with another internal MTA (e_{13}). It may also SMTP-connect with (an MTA of) the receiving organization (e_{18}) – notice that sending and receiving organizations may be identical, in which case the involvement of at least two MTAs of the ESP is assumed without reduction of generality. A third, rarely used possibility is for the MTA to establish a connection to other e-mail nodes on the Internet, to an SMTP relay (e_{15}) or to a gateway (e_{16}). No other connection is possible and modeled.

$MTA_{sendOrg}$. An MTA getting e-mails from another internal MTA can deliver to the same e-mail nodes that $MTA^{inc}_{sendOrg}$ can. Edges e_{17},…,e_{20} model these connections.

SMTP-Relay. An SMTP relay can connect to the same e-mail nodes as $MTA_{sendOrg}$. The only exception to this is $MTA_{sendOrg}$ itself, because a sending organization is either not involved in the process at all or its e-mail environment is already passed. Accordingly, we find edges e_{21},…,e_{23}.

$GW_{SMTP,B}$. E-mail nodes denoted as $GW_{SMTP,B}$ are defined as nodes which make outgoing connections other than SMTP-based ones. The only nodes to be considered are $GW_{A,B}$ (e_{24}) and $GW_{A,SMTP}$ (e_{25}).

$GW_{A,SMTP}$. This denotes gateways with outgoing SMTP connections. That is, they can connect with the same nodes as an SMTP relay (e_{26},…,e_{28}).

$GW_{A,B}$. Regarding outgoing connections, this kind of gateway can be treated same way as a node of type

$GW_{SMTP,B}$. Hence, we find edges e$_{29}$ and e$_{30}$.

MTA^{inc}_{recOrg}. An MTA at the recipient's end accepting SMTP connections can either deliver, forward, or reject an e-mail. If it is delivered, it passes the e-mail either to the local MDA (e$_{30}$) or to another internal MTA (e$_{31}$); in both cases we find internal e-mail processing. If it is delivered, then it either passes the e-mail to the local MDA (e$_{30}$) or to another internal MTA (e$_{31}$); in both cases we find an internal e-mail processing. Because forwarding or rejection of an e-mail initializes a new sequence, as mentioned earlier, edges dedicated to both are not integrated.

MTA_{recOrg}. An internal MTA getting e-mails from MTA^{inc}_{recOrg} either passes an e-mail to another internal MTA (e$_{34}$) using SMTP or to the local MDA (e$_{33}$). This process is denoted internal delivery.

MDA_{recOrg}. The MDA is responsible for storing an e-mail in the recipient's local e-mail box residing on the mail server $MailServ_{recOrg}$ (e$_{37}$). This is the second step of the internal e-mail delivery.

$MailServ_{recOrg}$. Most mail servers provide an interface for recipients' MUAs which access the user's e-mails with a mail access protocol such as IMAP or POP. These protocols are pull protocols, because the MUA initiates the dialogue with the mail server. However, when a connection like the present is established, e-mails are directed to the MUA. Alternatively, a mail server can provide an internal interface for a web server (e$_{36}$).

$WebServ_{recOrg}$. The web server is an intermediate node between the mail server and the MUA and allows an HTTP-based access of e-mails (e$_{38}$). As web browsers are usually installed on users' devices this kind of platform-independent e-mail access is broadly available and convenient.

MUA_{rec}. The destination of an e-mail is the recipient's MUA. MUA_{rec} does not have any outgoing edges, because any outgoing connection relates to the forwarding of an e-mail and is thus treated as a new sequence.

It is proved by this exploration that the image of f is a subset of P.

(2) Injection of f follows directly from lemma 1.

(3) Surjection of f means that each path in G from $v_{start} \in V_1 \cup V_2$ to MailServ$_{recOrg}$ has an inverse image in S. It has been already shown that each edge e=(v,w) ∈ E corresponds to exactly one type of e-mail connection between the e-mail nodes corresponding to v and w. This means, given a path p∈P, that for each edge e=(v,w)∈E with v and w being successive nodes in p there exists one and only one e-mail connection between e-mail nodes represented by v and w. So, each path p∈P has an inverse image in S. ■

8. References

[1] Dwork, C.; Goldberg, A.; Naor, M.: "*On Memory-Bound Functions for Fighting Spam*", Microsoft Research Report, http://research.microsoft.com/research/sv/PennyBlack/demo/lbdgn.pdf, 2002, Accessed 06-11-2005.

[2] Dwork, C.; Naor, M.: "*Pricing Via Processing Or Combatting Junk Mail",* Lecture Notes in Computer Science 740, 1993, 137–147.

[3] Freier, A. O., Karlton, P. and Kocher, P. C., "*The SSL protocol version 3.0*", Internet draft, IETF Network Working Group, 1996.

[4] Graham, P., "*A Plan for Spam",* http://www.paulgraham.com/spam.html, 08-2002, Accessed 06-11-2005.

[5] Hall, R., *Channels: Avoiding Unwanted Electronic Mail,* Proceedings DIMACS Symposium on Network Threats DIMACS, 1996.

[6] Harris, E., "*The Next Step in the Spam Control War: Greylisting*", http://projects.puremagic.com/greylisting/, 2003, Accessed 06-11-2005.

[7] ICANN, "*New sTLD RFP Application .mail*", http://www.icann.org/tlds/stld-apps-19mar04/mail.htm, 2004, Accessed 06-11-2005.

[8] Ioannidis, J., "*Fighting Spam by Encapsulating Policy in Email Addresses*", The 10th Annual Network and Distributed System Security Symposium Conference Proceedings, 2003.

[9] Levine, J.; DeKok, A. et al., "*Lightweight MTA Authentication Protocol (LMAP) Discussion and Comparison*", Internet Draft, http://www.taugh.com/draft-irtf-asrg-lmap-discussion-01.txt, 2004, Accessed 06-11-2005.

[10] Mori, G.; Malik, J., "*Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA*", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.

[11] MessageLabs, *Monthly Report April 2005*, http://www.messagelabs.com/emailthreats/intelligence/reports/monthlies/april05/default.asp, Accessed 06-11-2005.

[12] Symantec, *Spam statistics*, http://www.symantec.com/region/de/PressCenter/spam.html, Accessed 06-11-2005.

[13] The Honeynet Project, "*Know your Enemy: Tracking Botnets - Using honeynets to learn more about Bots*", 03-13-2005, http://www.honeynet.org/papers/bots/, Accessed 06-11-2005.

[14] Yahoo, "DomainKeys: Proving and Protecting Email Sender Identity", http://antispam.yahoo.com/domainkeys, Accessed 06-11-2005.