JISSec 3(2) 2007

Journal of
Information System
Security

www.jissec.org

Do anti-spam measures effectively cover the e-mail communication network? A formal approach

GUIDO SCHRYEN

Institute of Information Systems and Operations Research
RWTH Aachen University
Templergraben 64 52062 Aachen
Germany

Abstract

Spam e-mails have become a serious technological and economic problem. Up to now, by deploying complementary anti-spam measures, we have been reasonably able to withstand spam e-mails and use the Internet for regular communication. However, if we are to avert the danger of losing the Internet e-mail service in its capacity as a valuable, free and worldwide medium of open communication, anti-spam activities should be performed more systematically than is currently the case regarding the mainly heuristic, anti-spam measures in place. A formal framework, within which the existing delivery routes that a spam e-mail may take, and anti-spam measures and their effectiveness can be investigated, will perhaps encourage a shift in methodology and pave the way for new, holistic anti-spam measures.

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 spam delivery routes. The most important anti-spam measures are then described. Methods controlling only specific delivery routes are evaluated in terms of how effectively they cover the modeled e-mail infrastructure; methods operating independently of any particular routes ceive a more general assessment.

Key Words: e-mail, spam, e-mail infrastructure, anti-spam measures, spamming options

Introduction

Spam e-mails have become a serious technological and economic problem. Up to now, we have been reasonably able to withstand spam e-mails, although statistics show a percentage of spam of more than 60% (MessageLabs 2005; Symantec 2005). The availability of the Internet e-mail system for regular e-mail communication is currently ensured by complementary anti-spam measures, that is, mainly by blocking and filtering procedures. However, if we are to avert the danger of losing the Internet e-mail service in its capacity as a valuable, free and worldwide medium of open communication, anti-spam activities should be performed more systematically than is currently the case regarding the mainly heuristic, anti-spam measures in place. A formal framework, within which the different spam delivery routes, and anti-spam measures and their effectiveness can be investigated, will perhaps encourage a shift in methodology and pave the way for new, holistic anti-spam measures. 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 existing delivery routes that a spam e-mail may take are derived and presented as regular expressions. These formal expressions are then grouped into categories according to the types of organization that are participating in e-mail delivery. Section four evaluates what are currently the most important anti-spam measures in terms of how far these effectively cover the delivery routes. Section five summarizes the results presented in this paper and outlines future work.

A Model of the Internet E-mail Infrastructure

The Internet e-mail infrastructure is modeled as a directed graph G, to be defined in the first subsection. In the second subsection, the appropriateness of G of modeling the e-mail infrastructure is discussed and it is shown that types of e-mail delivery are represented by (directed) paths in G. Since any way of making e-mail delivery is obviously also a way of making spam delivery, the set of e-mailing options and the set of spamming options can be regarded as being identical (as can also the corresponding sets of types of delivery) and can hereinafter be understood to be referred to interchangeably.

The Definition

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

The construction of G follows these ideas:

- Graph nodes represent types of e-mail nodes as specified above. Directed edges correspond to e-mail connections between two e-mail nodes, with the edges' direction indicating the orientation "client to server". The edges are assigned a specific value, which is a set of labels representing those protocols which are feasible for the particular edge or connection respectively. Therefore, G can be denoted as a directed, labeled graph.
- The set of e-mail nodes to be modeled is mainly gathered from technological documents, such as RFCs, technological reports in the Internet literature, and practical experience. Hence, completeness cannot be guaranteed. Where necessary, the set has to be extended.
- Each e-mail node can be associated with protocols for incoming connections and protocols for outgoing connections. They are gathered from the same documents and sources as are mentioned above, so again, completeness cannot be guaranteed. Communication between the e-mail nodes (EN) ENA and ENB is possible if, and only if, there is at least one protocol which can be used by ENA for an outgoing connection and by ENB for an incoming connection, i.e. if the intersection of the protocol sets is not empty. Hence, an edge (A;B) is modeled if, and only if, ENA as client can communicate with ENB as server, where ENA corresponds to A and ENB corresponds to B. The assigned labels correspond to the intersection of the protocol sets.

Now we can formally describe G: 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 are then explained in more detail.

The set of vertices can be depicted as the disjoint union of five vertex sets V1,…,V5. 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 (ESP), Internet, receiving organization, and recipient. Where recipients do not use an ESP for the reception of e-mails but run their 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. Let the set of vertices be

V=V1 … V5 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,

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

and let the set of (protocol) labels be $L = \{SMTP, SMTP^*, \overline{SMTP^*}, HTTP(S), INT, MAP\}$.

E and c are not defined formally, but shown graphically in figure 1. Each vertex corresponds to a type of e-mail node. An edge e=(v1,v2), with a label c(e) ∈ 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.

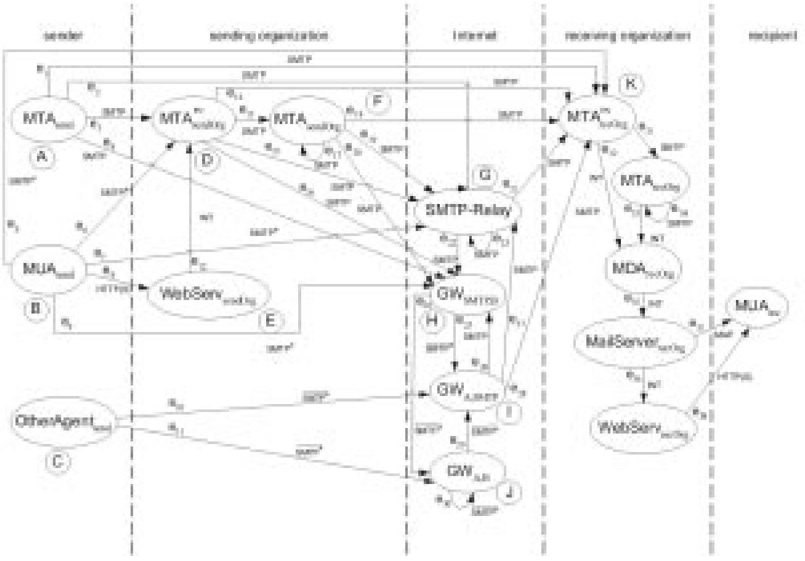

Fig. 1. Internet e-mail infrastructure as a directed graph.

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 which has 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, $MTA_{sendOrg}^{inc}$ can alternatively be denoted as $MSA_{sendOrg}$ and MTA_{recOrg}^{inc} as MSA_{recOrg} respectively. Port 587 is reserved for e-mail message submission. Most e-mail clients and servers can be configured to use port 587 instead of port 25; however, this is not always possible or convenient and, in such cases, port 25 can serve for message submission as well. 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 prior to an SMTP submission session ("SMTP after POP"). $\overline{SMTP^*}$ is the union of three sets of protocols. The first contains all Internet application protocols except SMTP*, and 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 -, consists of 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 or her MUA. Internet Message Access Protocol (IMAP) version 4 (RFC 1730) and POP version 3 (RFC 1939) are among the most deployed protocols here. The set HTTP(S) contains the protocols HTTP (RFC 2616) as well as its secure versions "HTTP over SSL" (Freier et al. 1996) and "HTTP over TLS" (RFC 2818). Finally, the set INT denotes protocols for and procedures in internal e-mail delivery, that is, it is concerned with processes inside the receiving organization, such as getting e-mails from an internal MTA and storing them in the users' e-mail boxes.

The Appropriateness

The appropriateness of graph G in the context of modeling the Internet e-mail infrastructure is given by the fact that different types of e-mail delivery can be described by a set of specific (directed) paths in G. This issue is addressed in three steps:

1. The e-mail nodes modeled are motivated.
2. For the e-mail nodes, possible protocols for incoming connections and protocols for outgoing connections are identified. The edges in G were defined on the basis of these protocols.
3. A set of (directed) paths in G is identified. This models different types of e-mail delivery.

Technical e-mail nodes can be assigned to the organizational unit that acts as the sender of an e-mail, to the sender's organization (sender's ESP), to the recipient, to the recipient's organization (recipient's ESP), and to the Internet subsuming all other organizational units. On the 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 take, as a rule, 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 either 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 an e-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, RFC 2033) is used to relay messages to the MDA, the MDA is termed Local Delivery Agent (LDA). Before an e-mail passes the first MTA of the receiving organization, it may be relayed by an intermediate SMTPrelay 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 (when this node pretends to be the original client it is referred to as SMTP proxy). 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. These are used, for example, for SMTP tunneling and are 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).

Because the term "proxy" is used in different contexts, some remarks on it seem appropriate here. The notion "proxy" generally denotes a service that allows clients to make indirect network connections to other network services. Proxies pretend to act as the original client and do not disclose the actual client; only the access to proxies' log files enables the identification of the actual client. The notion "proxy" does not give any information about the dissimilarity of the protocols used for incoming and outgoing connection. Some MTAs or relays are configured as proxies, meaning that they do not insert a Received entry into the e-mail header. When an MTA or other client on a third party computer is remotely controlled by a spammer, this client acts as a proxy, too. The PC is then called a "zombie PC". Even gateways can implement a proxy function.

Figure 2 provides an overview of existing e-mail nodes, using a class diagram. It should be noted that the e-mail nodes are logical nodes representing pieces of software, several of which might be executed in one physical node in a particular instance of e-mail delivery (for example MTA_{recOrg}^{inc} and MDA_{recOrg}).

Having motivated the nodes and vertices respectively, we now take a look at the protocols and connections by applying the design criteria for edges in G (see above): an edge e=(v1,v2) 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 to an ESP's incoming MTA (e3), to Internet nodes accepting SMTP connections, viz. an SMTP relay (e2) and a gateway (e4), or to an MTA of the receiving organization or recipient (e1). Other connections are not possible.

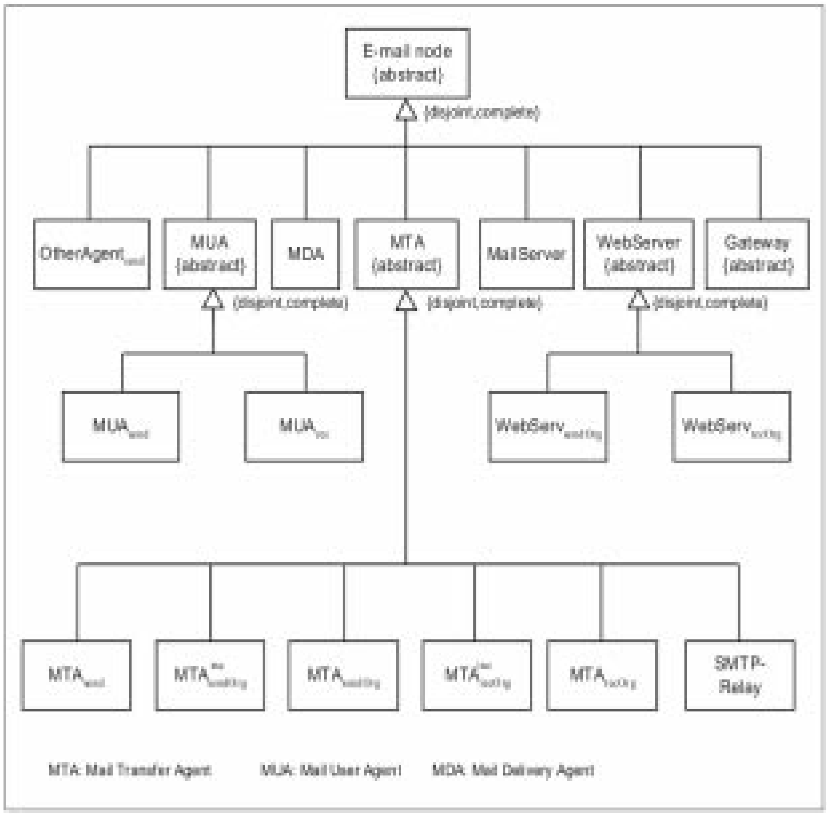

Fig. 2. Internet e-mail nodes.

MUA_{send} : An e-mail sender who operates an MUA can basically connect either to all nodes with an SMTP interface for incoming connections, or to a web server of the sending organization. HTTP(S)-based connections to other nodes are covered by the node OtherAgentsend. In the former case, the MUA can connect to the same nodes as the MTAsend (e5, e6, e7, e9). However, given the involvement of an MUA, the set of protocols has to be extended to SMTP*. If a connection is made to a web server, 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 (e10, e11) as modeled.

$WebServ_{sendOrg}$: A web server of an ESP sends its e-mails to an internal MTA (e12). Connections to other nodes generally do not exist.

$MTA_{sendOrg}^{inc}$: The MTA that is responsible for incoming messages most commonly SMTP-connects to another internal MTA (e13). It may also SMTP-connect to (an MTA of) the receiving organization (e18) - notice that sending and receiving organizations may be identical, in which case we can assume, without compromising the validity of the graph, that at least two MTAs of the ESP are involved. 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 (e15) or to a gateway (e16). Other connections are not possible and are not modeled.

$MTA_{sendOrg}$: An MTA receiving e-mails from another internal MTA can deliver to the same e-mail nodes that $MTA_{sendOrg}^{inc}$ can. Edges e17,…,e20 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 itself, because a sending organization is either not involved in the process at all or its e-mail environment has already been passed. Accordingly, we find edges e21,…,e23.

$GW_{SMTP,B}$: 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 $GW_{A,B}$ (e24) and $GW_{A,SMTP}$ (e25).

$GW_{A,SMTP}$: This denotes gateways with outgoing SMTP connections. They can connect to the same nodes as an SMTP relay (e26,…,e28).

$GW_{A,B}$: Regarding outgoing connections, this kind of gateway can be treated in the same way as a node of type $GW_{SMTP,B}$. Hence, we find edges e29 and e30.

MTA_{recOrg}^{inc} : A recipient MTA that accepts SMTP connections can either deliver, forward, or reject an e-mail. If the e-mail is delivered, it is passed either to the local MDA (e30) or to another internal MTA (e31); in both cases we find internal e-mail processing. Because forwarding or rejection of a n e-mail initializes a new sequence, as mentioned earlier, edges dedicated to both are not integrated.

MTA_{recOrg} : An internal MTA, receiving e-mails from MTA_{recOrg}^{inc} , either passes an e-mail to another internal MTA (e34) using SMTP or to the local MDA (e33). This process is denoted as 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 $MailServ_{recOrg}$ (e37). This is the second step of the internal e-mail delivery process.

$MailServ_{recOrg}$: 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, the MUA initiating the dialogue with the mail server. However, when a connection of this kind is established, e-mails are directed to the MUA. Alternatively, a mail server can provide an internal interface for a web server (e36).

$WebServ_{recOrg}$: The web server is an intermediate node between the mail server and the MUA and allows HTTP-based access of e-mails (e38). This kind of platform-independent e-mail access is widely available and convenient: web browsers are usually installed on users' devices.

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.

According to the construction of G, e-mail delivery routes are represented by paths in G. As it is essential for today's e-mail delivery process that the way in which an e-mail node received an e-mail does not restrict the way it passes the e-mail forward, each path *p* corresponds to a feasible e-mail delivery route. It should be noted that completeness is intended but not guaranteed as is not the completeness of e-mail nodes nor their communication connections. We are only interested in complete e-mail deliveries, which means that the e-mail has reached the recipient's e-mail box on his or her 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 the case of bouncing or forwarding e-mails, by an ESP's MTA.

Each option for sending one e-mail allows, in principle, the sending of many, e.g. millions of e-mails, as spammers do. Thus, the set of options for sending one e-mail has to be taken into account when identifying options for sending spam e-mails. Obviously, the set of all paths $p=(v_{start},\dots,MailServ_{recOrg})$ with $v_{start} \in V_{start}:=V_1 \cup V_2$ gives us all options for sending (spam) e-mails, thus providing a formal approach to spamming options. In the following section, these paths are formally derived and categorized..

Derving and Categorizing the Spam Delivery Reoutes

Graph G will serve as a basis for deriving all spam delivery routes. It provides a formal framework within which the effectiveness of (present and future) 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 spam delivery routes are formally presented with means of automata theory in the first subsection. A categorization of the routes follows in the second subsection.

Deriving the spam delivery routes

The goal of this subsection is to derive the set *P* of all paths $p=(v_{start},\dots,MailServ_{recOrg})$ with $v_{start} \in V_{start}:=V_1 \cup V_2$ *P* is arrived at by applying some basic ideas from automata theory: the graph G is transformed into a Deterministic Finite Automaton (DFA) A=(S, Σ,δ Start,F) where S is a finite set of states, Σ is an alphabet, "Start" is the initial state, Fδ S is the set of final states, and is a function from S x Σ to S. This automaton recognizes a language that (bijectively) corresponds to P, such that w=(w_1,…, w_n)∈ L(A) ⇔ (w1,…,wn) ∈ 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. Σ corresponds to the nodes of G, as well. An edge (v_1,v_2) means that the transition function δ includes δ (s_1,s_2)=s_2, that is, 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, also needs to include δ (Start,s2)=s2 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_{recOrg}^{inc} .

Given the equivalence between DFAs and regular expressions, the language recognized by the DFA A - and thus P - can be described with a regular expression. For simplicity, the states are labeled with capital letters which are assigned to the corresponding nodes (see figure 1). Elements of Σ are set in lowercase letters. Given two regular expressions r_1 and r_2, ~ denotes the relationship between r_1 and r_2 with r_1~r_2: ⇔ L(r1)=L(r2); let Λ be the regular expression with L(Λ)= ε

. Using the edges of G, we get L(A)=L(Start) with

Start~ aA ∨ bB ∨ cC ∨ dD ∨ fF (1)
A ~ kK ∨ gG ∨ dD ∨ hH (2)
B ~ dD ∨ gG ∨ eE ∨ hH (3)
C ~ iI ∨ jJ (4)
D ~ kK ∨ fF ∨ gG ∨ hH (5)
E ~ dD (6)
F ~ kK ∨ gG ∨ fF∨ hH (7)
G ~ kK ∨ gG ∨ hH ∨ (8)
H ~ jJ ∨ iI (9)
I ~ hH ∨ gG ∨ kK (10)
J ~ iI ∨ jJ (11)
K ~ (12)

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

α~βα∨γ,ε∉ L(β) (13)
α~β *γ

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

Start~ ak ∨ agg*k ∨ agg*hH∨
ad (k∨ ff*k ∨ ff*gg* ∨ ff*gg*hH ∨ ff*hH ∨ gg*k ∨ gg*hH ∨ hH) ∨ ahH
(bd ∨ bed) (k ∨ ff*k ∨ ff*gg*k ∨ ff*gg*hH ∨ ff*hH ∨ gg*k ∨gg*hH hH) ∨
bgg*k ∨ bgg*hH ∨ bhH ∨ cihH ∨ cigGH ∨ cigg*k ∨ cik ∨ cjj*ihH
cjj*igg*H ∨ cjj*igg*k ∨ cjj*ik
d (k ∨ ff*k ∨ ff*gg*k ∨ ff*gg*hH ∨ ff*hH ∨ gg*k ∨ gg*hH ∨ hH)

$$ff^*k \vee ff^*gg^*k \vee ff^*gg^*hH \vee ff^*hH \qquad (14)$$

with

$$H \sim (jj^*l\,(h \vee gg^*h) \vee l\,(h \vee gg^*h))^*$$
$$(jj^*igg^*k \vee jj^*ik \vee igg^*k \vee ik) \qquad (15)$$

As (14) shows, (spam) e-mail delivery routes are numerous and call for a categorization of a manageable format.

Categorizing the spam delivery routes

A useful way of proceeding is to place in one category delivery routes which are defined by the same types of organizational unit; the types are "sender", "sending organization" or "ESP", "Internet", "receiving organization", and "recipient" (see figure 1). Because complete e-mail delivery invariably presupposes a receiving organization and the recipient has no influence on the process, these units can be ignored. Categories arise, then, from the respective participation or non-participation of a local sender, an ESP (as the sending organization) and the Internet (application level infrastructure), giving eight possible combinations. The categories are shown in table I.

In an e-mail communication network, as modeled above, an Internet node can never be the first participant in a delivery process: an e-mail goes out from a node in either a sender's or a sending organization's environment, including instances of computers infected or controlled remotely. The types in the first two rows of table I are, therefore, merely theoretical possibilities. Scenarios I and II occur when e-mail providers or their MTAs are corrupt. Given that ESPs and the corresponding MTAs are limited in number in comparison with users, it should be possible to effectively deal with these options to send spam e-mails. In all other scenarios spam e-mails issue from a local client, the obvious starting point, and this is probably what happens most of the time. Scenario III is one in which the spammer does not use an ESP, although of course, he or she uses an ISP operating on layers no higher than the transport layer, that is the ISP generally does simple forwarding of Transmission Control Protocol (TCP) packets or Internet Protocol (IP) packets.

The spammer connects to an MTA of the receiving organization directly and so is restricted to the e-mail ports implemented there. This, however, will usually be port 25 or 587, making it easy for ISPs to stop most spam e-mails sent in this way by simply blocking TCP packets to these ports. Scenario III also contains a specific case of zombie PCs (see below). Spamming in the manner of scenario IV is much harder to tackle because, this time, the spammer may use all Internet nodes, including gateways. In scenario V, to circulate spam, the spammer simply takes advantage of the e-mail service offered by an ESP.

Even if a limit is imposed upon the number of e-mails permitted per day and account, there remains the task of preventing the spammer from setting up new accounts automatically. Scenario V also includes the case of zombie PCs - those PCs which are exploited and controlled remotely by spammers, often via Trojan horses -, which connect to a user's sending organization and ESP respectively. Zombie PCs are also called bots when they belong to a botnet which is controlled by botnet masters. Aided by a botnet and thousands of bots, an attacker is able to send massive amounts of spam e-mail (The Honeynet Project & Research Alliance 2005). More than half of all spam e-mails are assumed to be sent via botnets (Ilett 2004; Sandvine 2004; Sanders 2005), either via a users sending organization or by the usage of a direct connection to the recipient's MTA (scenario III). Scenario VI seems quite unlikely. The spammer uses an ESP which forwards e-mails, sending them to intermediate nodes on the Internet. This might occur if an ESP supported the spamming activities of customers.

No.	Scenario	Sender	Sending organization	Internet
-	-			
-	-			X
I	ESP itself spams or its MTAs are corrupt; direct connection to $\text{MTA}_{\text{recOrg}}$		X	
II	ESP itself spams or its MTAs are corrupt; use of intermediate Internet nodes like relays		X	X
III	Spammer uses local client; direct connection to $\text{MTA}_{\text{recOrg}}$ (via dial-in or LAN connection)	X		
IV	Spammer uses local client and intermediate Internet nodes like relays (via dial-in or LAN connection)	X		X
V	Spammer uses local client and ESP (via dial-in or LAN connection)	X	X	
VI	Spammer uses local client and ESP (via dial-in or LAN connection) that involves intermediate Internet nodes like relays	X	X	X

Table I. Spamming categories.

If we now assign to these six categories the spam delivery routes in (14), we obtain, after a number of transformations:

Start ~
I: d (k ∨ ff*k) ∨ ff*k
II: (d ∨ Λ) (ff*gg*k) ∨ dgg*k
(d ∨ Λ) (ff*gg*hH) ∨ dgg*hH
(d ∨ Λ) (ff*hH) ∨ dhH
III: ak ∨
IV: (a∨ b) (gg*k)
(a∨ b) (gg*hH ∨ hH)
cj*I (hH ∨ gg*H ∨ gg*k ∨ k)
V: (ad ∨ bd ∨ bed) (k ∨ ff*k)
VI: (ad ∨ bd ∨ bed) (ff*gg*k ∨ gg*k)
(ad bd ∨ bed) (ff*hH ∨ hH)
(ad ∨ bd ∨ bed) (ff*gg*hH ∨ gg*hH) (16)

with
H ~ (jj*I (h ∨ gg*h) ∨ I (h ∨ gg*h))* (jj*igg*k ∨ jj*ik ∨ igg*k ∨ ik)

The regular expression in (16) is constructed and represented in a form which permits us to match up each individual line with a corresponding set of delivery routes, defined by the same types of e-mail node. This representation will form the basis for the discussion in section 4. Having formally identified spam delivery routes, we can assess the effectiveness of the most frequently discussed and applied anti-spam measures in the next major section.

Some example delivery routes and their formal representations

To illustrate how (common) options of sending (spam) e-mails are covered by the formal representation in (16), some of the former ones are exemplified:

- A user in the office or at home uses an MUA, e.g. Outlook (Express), and sends an e-mail to the MTA of his or her ESP. The message is then relayed by some consecutive MTAs of this ESP before the message is delivered to an MTA of the recipient's ESP. This route is covered by the regular expression bdf*k (scenario V). It also covers the case in which an MUA is remotely controlled by a botnet master and messages follow the path which is described above.
- An e-mail user can also use a web interface for sending e-mails. This is particularly useful when he or she is abroad and PCs are available in an Internet cafe or in a conference's e-mail room. Then, a message is sent consecutively to the ESP's web server, to at least two MTAs, and, finally, to an MTA of the recipient's ESP. This route corresponds to bedf*k (scenario V).

- Senders of bulk e-mails often use a mass e-mail program residing on their host. When spammers use such a local MTA they often camouflage the spam source by sending the messages to an open proxy which subsequently sends messages to an MTA of the receiving organization (agk, scenario IV).
- Trying to obfuscate the spam e-mail's source, mass e-mail tools can be used that are designed to connect with (a chain of) SOCKS 4 or SOCKS 5 proxies. The chain's last proxy SMTP-connects with an MTA of the receiving organization. This delivery route is covered by the expression cj*ik (scenario IV).
- A script, e.g. a Common Gateway Interface (CGI) script, running on the spammer's or a third party's host, can HTTP-connect to a (misconfigured) web server which provides e-mail services to the public. For example, the entering of "adding new user inurl:addnewuser" into a search engine leads to many web servers which allow anyone to set up a new user account and send an arbitrary number of e-mails on behalf of this account. The web server itself connects to a (usually local) MTA which, subsequently, sends the message to an MTA of the receiving organization. In our model, the bundle consisting of the web server and the (local) MTA is referred to as gateway ($GW_{A,SMTP}$). cik is the regular expression covering this part of scenario IV.

The Effectiveness of Anti-Spam Measures

Anti-spam measures can be distinguished according to whether they control only particular delivery routes of the set derived in section 3, or whether they operate irrespectively of the delivery routes that spam may take. Both types of anti-spam measures require distinct discussions and are addressed, respectively, in the following two subsections. Each anti-spam method is briefly described and assessed in terms of its effectiveness.

Route-specific anti-spam measures

Anti-spam measures controlling only some delivery routes include

- blocking mechanisms accepting or rejecting e-mails on the basis of the IP number of the delivering MTA,
- limiting the number of e-mails per account and unit of time, for example per day,
- blocking TCP port 25 which is used to send e-mails,

- digital signature authentication based on public key cryptography,
- LMAP authentication relying on DNS records and
- an organizational and technological framework which offers a new top level domain.

They can be mapped onto the e-mail infrastructure with the help of the individual lines of the regular expression in (16), each representing delivery routes that are defined by the same type of e-mail node involved. By providing the lines in rows and the anti-spam methods in columns, table II reveals which spam delivery routes are combated by which method. An "x" indicates effective coverage, a blank space indicates the impossibility thereof. The table is explained in the following paragraphs, which dedicated to the individual anti-spam methods.

			Anti-spam measures					
Scenario	Regular expression	types of 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, relay(s)						x
II	(d ∨ Λ) (ff*gg*hH) ∨ dgg*hH	MTA of provider, relay(s) and gateway(s)						
II	(d ∨ Λ) (ff*hH) ∨ dhH	MTA of provider, gateway(s), relay(s) possible						
III	ak	Local MTA	(x)		x	x	x	x
IV	(a ∨ b) gg*k	Local MTA or MUA, relay(s)			x	x	x	x
IV	(a ∨ b) (gg*hH ∨ hH)	Local MTA or MUA, relay(s) and gateway(s)			x			
IV	cj*i (hH ∨ gg*H ∨ gg*k ∨ k)	Local agent other than MTA or MUA, gateway(s), relay(s) possible						
V	(ad ∨ bd ∨ bed) (k ∨ ff*k)	Local MTA or MUA, MTA(s) of provider						
VI	(ad ∨ bd ∨ bed) (ff*gg*k ∨ gg*k)	Local MTA or MUA, MTA(s) of provider, relay(s)					x	x
VI	(ad ∨ bd ∨ bed) (ff*hH ∨ hH)	Local MTA or MUA, MTA(s) of provider, gateway(s), relay(s) possible						
VI	(ad ∨ bd ∨ bed) (ff*gg*hH ∨ gg*hH)	Local MTA or MUA, MTA(s) of provider, relay(s) and gateway(s)						

Table II. Effectiveness of anti-spam measures.

Blocking mechanisms

The blocking of e-mails is a widely used mechanism by which e-mails are accepted or weeded out on the basis of the IP address of the sending node. IP addresses of notorious nodes are listed on local and/or public black lists. Ordb.org., for example, provides a list of open relays, and the Spamhaus Project (www.spamhaus.org) gives a realtime blacklist of IP addresses of verified spam sources, as well as a so-called Exploits Block

List. This is a database of IP addresses of third party exploits, including open proxies, worms/viruses with built-in spam engines, and Trojan horse exploits. Again, there are limits to what can be achieved by all of these. Black lists are weapons against repeatedly used nodes, but spammers tend to change their IP addresses continually, either by switching to other ISPs or by taking advantage of exploits on unsuspected third party nodes. For example, spammers can use relays and gateways running on computers with unsuspected IP numbers. The more permanent IP addresses on black lists tend to be those of corrupt ESPs, so it is mostly spam issued from these which is blocked (scenario I). A sure way to broaden the target is to block a full range of IPs (scenario V), for example of ISPs or even of a country known to harbor spammers. However, this can easily lead to a digital divide, and any measure running this risk hardly seems feasible in the long run, which is why the corresponding "x" in table 2 is bracketed.

Limiting the number of e-mails

A fairly simple method is to limit the maximum number of e-mails which can be sent per account and within a given period. This is only available where an ESP is made use of and, even then, the automatic set-up of infinitely many e-mail accounts presents a wide loophole. Some ESPs apply CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) procedures which require a number or a word appearing in a picture to be retyped before an account can be set up. However, Mori and Malik (2003) show how it is possible to automatically recognize the content of 92% of all pictures created by the Yahoo CAPTCHA process. A different attack on visual CAPTCHA processes is as follows: the spammer places the ESP's picture on his or her own web site and tricks users into believing that reading the text and entering it in a text field will give them access to adult information. The spammer then transfers the retyped text into the corresponding text field on the ESP's form. All this can be done automatically. In short, current implementations to ensure a manual set-up of e-mail accounts - and by this means to keep the number of accounts per user low - are liable to be evaded and are of doubtful value. This renders the quantitative restriction of e-mails a less than effective measure against spam; the corresponding column receives no entries.

Blocking TCP port 25

Blocking all (outgoing) TCP traffic on port 25 is a simple option for ISPs for banishing spam sent from spammers' and exploited computers when port 25 is used (scenario III and the first two versions of scenario IV). It should be noted that this, at the same time, hits deliveries from MTAs running on users' or companies' computers. Spam deliveries involving other ports or gateways, on the other hand, are not covered (the third version of scenario IV is not covered).

Digital signature

A powerful and promising way to secure e-mail communication is that of environments which enable the recipient to authenticate the sender or, at least, the sending organization. Public Key Cryptography supplies the mathematical and algorithmic basis for digitally signing documents, and Public Key Infrastructures (PKIs) provide the organizational framework. At present, implementations serve for the authentication of organizations and (second level) domains: most users do not (as yet) possess a pair of keys. The sending organization signs an e-mail with its private key, and the recipient uses the public key to verify the organization's domain and, to rule out forgery, matches it against the domain in the sender's address shown in the "From:" field. The best known example is probably Yahoo's "DomainKeys" (Yahoo 2005). This PKI-based measure presumes that the sending organization is not corrupt and that its MTAs do not suffer from exploits (scenarios I and II are not covered). It is effective when spammers use MTAs of their own or where exploits on unsuspected computers are concerned, for example, spamming machines remotely controllable via a Trojan horse: an e-mail sent by such a spamming machine, which circumvents the MTAs and the signing software of the sending organization, will fail to be authenticated by the recipient (scenarios III and IV are covered). Misuse of the user's private account information - and of the password in particular - to issue spam, on the other hand, poses quite a serious challenge, since the sending organizations cannot distinguish between genuine and forged e-mails (scenarios V and VI are not covered). A shortcoming of a PKI may also show up on a different plane in that spammers can readily obtain keys for a domain intended, and then used, solely for the temporary purpose of spamming.

Lmap

Another method of authentication is the Lightweight MTA Authentication Protocol (LMAP) (Levine et al. 2004), which in fact constitutes a whole LMAP family, encompassing different DNS-based procedures. These operate by checking whether a message that gives, say, buffy@sunnydale.com as its origin was actually sent from an MTA of the corresponding sunnydale.com organization. A negative result indicates forgery or that the sender has used an external e-mail relay. Among the most extensively reviewed procedures are: Reverse MX (RMX) Designated Mailers Protocol (DMP), Sender Policy Framework (SPF), and SenderID. However, no standardization has been achieved and the IETF working group MARID was dissolved in 2004. The LMAP family is effective in controlling direct e-mail deliveries --- a local MTA is used (scenario III) --- and those indirect deliveries that make use of relays and gateways (scenarios IV and VI). The weaknesses that the LMAP family exhibits are similar to those of the PKI-based measure.

New top level domain

ICANN presents an organizational and technological framework elaborated by Spamhaus (ICANN 2004), which introduces a new, sponsored top level domain (sTLD), for example .mail. This sTLD is intended to serve registrants exclusively for e-mail sending processes. A registrant must already have a registered domain key, say icann.org, which is a prerequisite for the acquisition of the domain key.sTLD - in this case icann.org.mail. There are further requirements which a registrant may have to meet, among them the availability of validated "Whois" information, appropriate technological anti-spam protection, and that the domain key must have been registered for a period of at least six months. Apart from this, the registrant must inform the central (sponsoring) organization (SO) of the IPs and hostnames of the sending mail servers. The SO makes an A records entry for the new domain on the DNS, which enables recipient MTAs to use an LMAP or a PKI-based authentication. The SO also receives any abuse messages concerning key.sTLD and so, at the same time, provides a control mechanism. The framework developed by Spamhaus promises to be effective against a wide range of modes of spamming, yet a fundamental question remains, that is: How can an appropriate technological anti-spam protection be achieved? The framework does not cover cases of spamming zombie PCs - PCs which are exploited and controlled remotely by spammers, often via Trojan horses (scenario V is not covered).

Summary

When summing up the effectiveness of the anti-spam measures indicated in table II, it must be stressed that no anti-spam measure is currently capable of effectively stopping those spam deliveries which take advantage of ESP infrastructures (scenario V). The main problems are third party exploits and that it is all too easy for spammers to set up e-mail accounts automatically. The former is a plague which is becoming more acute as botnets, networks of compromised and remotely controlled machines flourish among spammers (The Honeypot Project & Reseach Alliance 2005).

Comprehensive, non-route-specific anti-spam measures

Some anti-spam measures are independent of the delivery routes that spam may take, which is why they have been excluded from table II. Considered here are

- filters,
- blocking mechanisms based on gray lists,
- resource-based measures and
- address obscuring techniques.

Filters

Filters are interposed at the point where an e-mail reaches the organization's incoming mail server or the user's mail client . They access the e-mail document and heuristically classify it as either ham or spam.

Spam filters can be categorized according to the type of filtering method used (e.g. statistical filters (Graham 2002) or neural network-based filters - version 3 of the open-source spam filter software SpamAssassin uses a neural network) or according to which component of the e-mail is inspected, the header and/or the body. All filters suffer from the same drawbacks:

- Being heuristic, they may misclassify and release spam ("false-negatives") while filtering out and even deleting ham ("false-positives").
- Filters detect rather than prevent spam. But by the time spam e-mails are detected at the recipient's end, valuable resources (for example bandwidth, storage capacity, CPU time reserved for filter software) have been consumed.
- Spammers continue to upgrade their skills and it is doubtful whether filters can ever be effective on a long-term basis.

Additionally, filters may even backfire, beacuse to compensate for losses from spam detection, spammers are likely to send even more e-mails.

Gray list

Gray lists (Harris 2003) are used to block incoming e-mails temporarily. Spammers often do not bother to implement the standard resume feature, whereby a rejected e-mail is re-sent after a few minutes, apparently because many of their hosts of e-mail addresses are of little or no value anyway. Gray lists take advantage of this omission and first reject all arriving e-mails but allow them to pass if they return within a given time window. A gray list system implemented by RWTH Aachen University, for example, does this by recording the IP of the sending host, the sender address, and the recipient address. However, if spammers manage to raise the quality of their address stores, they will certainly implement the resume feature, thus bypassing the gray list technique.

Resource-based measures

Resource-based measures for preventing spam have been proposed in (Dwork et al. 2002) and (Dwork and Naor 1993). The principle is that sending an e-mail presupposes a calculation of a function that is time-consuming or memory-consuming. Such measures face at least two problems: (1) How can solicited bulk e-mail continue to be sent? (2) How can spammers be stopped from laying their hands on sufficient resources?

Address obscuring techniques

Address obscuring techniques (AOTs) generally aim either at concealing e-mail addresses or at restricting their use. One example of the former is the publication of addresses in forms which are easy for humans, though not for software harvesters, to decipher, by displaying them in pictures or obscuring them by "misspellings" (e.g. fooaetfoobar.com represents the address foo@foobar.com). Another is unguessable channel names, a tactic put forward by Hall, in which each legitimate correspondent of a user knows of a different channel (Hall 1996). A use-restrictive AOT is found, for instance, in Ioannidis' concept of a single-purpose address (SPA) with a (security) policy encapsulated therein, which defines the use to which the address may be put, for example who is authorized to e-mail to the address (Ioannidis 2003). A second feature of SPAs is their cryptographic protection. However, what is designed as a shield against tampering also makes SPAs rather difficult to handle, for the human users themselves. The result is that SPAs are primarily applied to machine-to-person communication, Where they are used for interpersonal communication, they are bound to cause some inconvenience. The owners cannot, after all, anticipate all the regular e-mail contacts that they may come to have when they first create the addresses. The establishing of new, subsequent contacts is exceedingly awkward with SPAs.

Summary and Conclusions

This paper is addressed towards evaluating the current effectiveness of important anti-spam measures. This is done by using a formal framework and a method which have a wider applicability, for the purpose of future changes of the Internet e-mail infrastructure and in the development of new anti-spam measures.

The e-mail infrastructure is modeled as a directed graph and a deterministic finite automaton. The appropriateness of the graph as a model of the real world e-mail infrastructure is formally proven. Automata theory, including, in particular, regular expressions, is used to formally derive and represent all possible ways of sending (spam) e-mails. These are categorized on the basis of the types of e-mail nodes involved in e-mail delivery.

The discussed anti-spam measures range from today's most widely applied techniques to promising new and much-discussed methods still awaiting implementation. Some of them are tied to particular delivery routes along which spam can be sent. Among these are blocking mechanisms that limit the number of e-mails per account and unit of time, the blocking of outgoing TCP port 25, digital signature authentication, LMAP authentication, and an organizational and technological framework which introduces a new top level domain. A comparison of these with the existing ways of making spam delivery shows that the exploitation of PCs and ESP infrastructures is not being effectively dealt with.

Some anti-spam measures combat spam in general, irrespective of any particular delivery route. These are: filters, blocking mechanisms based on gray lists, resource-based measures and address obscuring techniques. Filters are heuristic methods and may misclassify. Detecting rather than preventing spam, they do not save resources. Gray lists can be bypassed by implementing SMTP's resume feature. Resource-based measures are promising preventive measures, although some questions remain unanswered: (1) How can solicited bulk e-mail continue to be sent? (2) How can spammers be stopped from obtaining sufficient resources? Address obscuring techniques, which aim either at concealing e-mail addresses or at restricting their use, are not widely applied and little is known about their effectiveness. These techniques require further elaboration, though an effective prevention of address abuse by concealing information seems unlikely.

Today's most significant anti-spam activities are directed mainly at spam detection rather than at its prevention. This, however, may well be counterproductive, since spammers will send even more e-mails in order to compensate for their losses from detection, and valuable resources continue to be consumed; it is rather like shutting the stable door after the horse has bolted.

Generally, anti-spam activities should be performed more systematically than is currently the case with the mainly heuristic, anti-spam measures. Models and formal procedures, such as are used in this paper, are possibly an adequate way of assessing the effectiveness of anti-spam measures and developing new, holistic measures which would focus on the prevention of spam e-mails.

References

Dwork, C., Goldberg, A. and Naor, M. (2002), "On Memory-Bound Functions for Fighting Spam", Microsoft Research Report, http://research.microsoft.com/research/sv/PennyBlack/demo/lbdgn.pdf.

Dwork, C. and Naor, M. (1993), "Pricing Via Processing Or Combatting Junk Mail", Lecture Notes in Computer Science, 740: 137-147.

Freier, A. O., Karlton, P. and Kocher, P. C. (1996), "The SSL protocol version 3.0", Internet draft.

Graham, P. (2002), 'A Plan for Spam', http://www.paulgraham.com/spam.html, August 2002.

Hall, R.. (1996), 'Channels: Avoiding Unwanted Electronic Mail'. DIMACS Symposium on Network Threats. Nov 6-8. Piscataway, N.J.

Harris, E. (2003), 'The Next Step in the Spam Control War: Greylisting', http://projects.puremagic.com/greylisting/.

ICANN (2004), 'New sTLD RFP Application .mail', http://www.icann.org/tlds/stld-apps-19mar04/mail.htm, 19 April 2004.

Ilett, D. (2004), 'Most spam generated by botnets, says expert', http://news.zdnet.co.uk/internet/security/0,39020375,39167561,00.htm, 22 September 2004.

Ioannidis, J. (2003), 'Fighting Spam by Encapsulating Policy in Email Addresses'. 10th Annual Network and Distributed System Security Symposium. February 2003. San Diego, California.

Levine, J. et al. (2004), 'Lightweight MTA Authentication Protocol (LMAP) Discussion and Comparison'. Internet Draft.

Mori, G. and Malik, J. (2003), 'Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA'. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 16-22, Wisconsin.

MessageLabs (2005), 'Monthly Report April 2005', http://www.messagelabs.com/emailthreats/intelligence/reports/monthlies/april05/default.asp,April 2005.

RFC 1730, Crispin, M. (1994), 'Internet Message Access Protocol - Version 4'. IETF Network Working Group.

RFC 1939, Myers, J. and Rose, M. (1996), 'Post office protocol - version 3'. IETF Network Working Group.

RFC 2033, Myers, J. (1996), 'Local Mail Transfer Protocol'. IETF Network Working Group.

RFC 2034, Freed, N. (1996), 'SMTP Service Extension for Returning Enhanced Error Codes'. IETF Network Working Group.

RFC 2476, Gellens, R. and Klensin, J. (1998), 'Message Submission'. IETF Network Working Group.

RFC 2554, Myers, J. (1996), 'SMTP Service Extension for Authentication'. IETF Network Working Group.

RFC 2616, Gettys, J., Mogul, J., Frystyk, H., Masinter, L. and Leach, P. (1999), 'Hypertext Transfer Protocol - HTTP/1.1'. IETF Network Working Group.

RFC 2821, Klensin, J. (2001), 'Simple Mail Transfer Protocol'. IETF Network Working Group.

RFC 2852, Newman, D. (2000), 'Deliver By SMTP Service Extension'. IETF Network Working Group.

RFC 3207, Hoffman, P. (2002), 'SMTP Service Extension for Secure SMTP over Transport Layer Security'. IETF Network Working Group.

Sanders, T. (2005), 'Microsoft takes on spamming botnets', http://www.vnunet.com/vnunet/news/2144976/microsoft-takes-spamming.

Sandvine (2004), 'Zombie PCs spew out 80% of spam'.

Symantec (2005), 'Spam statistics', http://www.symantec.com/region/de/PressCenter/spam.html.

The Honeynet Project & Research Alliance (2005), 'Know your Enemy: Tracking Botnets - Using honeynets to learn more about Bots', http://www.honeynet.org/papers/bots/, 13 March 2005.

Yahoo (2005), 'Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys)', Internet Draft.

Author Biography

Mr. Guido Schryen graduated from the RWTH Aachen University (Germany), where he earned Masters' degrees in Computer Science and in Operations Research. He received his PhD from the Faculty of Business Administration and Economics of RWTH Aachen University where he now holds a postdoctoral position. His current research activities focus on Internet security and anti-spam measures. He may be reached at chryen@winfor.rwth-aachen.de.