Understanding Forgery Properties of Spam Delivery Paths

Fernando Sanchez
Florida State University
sanchez@cs.fsu.edu

Zhenhai Duan
Florida State University
duan@cs.fsu.edu

Yingfei Dong
University of Hawaii
yingfei@hawaii.edu

ABSTRACT

It is well known that spammers can forge the header of an email, in particular, the trace information carried in the `Received:` fields, as an attempt to hide the true origin of the email. Despite its critical importance for spam control and holding accountable the true originators of spam, there has been no systematic study on the forgery behavior of spammers. In this paper, we provide the first comprehensive study on the `Received:` header fields of spam emails to investigate, among others, to what degree spammers *can and do* forge the trace information of spam emails. Towards this goal, we perform empirical experiments based on two complementary real-world data sets: a 3 year spam archive with about 1.84 M spam emails, and the `MX` records of about 1.2 M network domains. In this paper, we report our findings and discuss the implications of the findings on various spam control efforts, including email sender authentication and spam filtering.

1. INTRODUCTION

Due to the weak security design of the Simple Mail Transfer Protocol (SMTP) [13], spammers have immense power and flexibility in forging email headers to mislead email recipients about the real sender of a spam email and to hide the true origin of the email. To ease exposition, in this paper we refer to all categories of unwanted emails as spam emails (including, for example, spam, phishing emails, and email-based extortion and threats) and senders of these emails as spammers. The ability of spammers to forge email headers often complicates the spam control efforts and makes it hard to hold accountable the true spam originators. This presents a great challenge for law enforcement to properly investigate and prosecute email-based criminals [1].

On the other hand, despite its critical importance for spam control and holding accountable the true originators of spam, there has been no systemic study on the forgery behavior of spammers, except anecdotal evidence of spam header forgery. In this paper we provide the first comprehensive study on the forgery behavior of spammers. Given the importance of the trace information carried in the `Received:` header fields in the investigation of the true origin of a spam email, in this paper we concentrate our efforts on the `Received:` header fields of spam emails to investigate, among others, to what degree spammers *can and do* forge the trace information of spam emails [16].

Towards this goal, we perform empirical experiments based on two complementary real-world data sets. The first one is a 3-year spam archive from 2007 to 2009 [9], which contains about 1.84 M spam emails. We extract the `Received:` header fields of each spam email, and refer to the sequence of mail servers carried in these header fields as the *spam delivery path* of the email. The second data set is the mail exchanger (MX) records of about 1.2 M network domains. We use the study on the MX records of the network domains to interpret and confirm the main findings from the first data set. Our main findings (regarding the degree to which spammers can and do forge spam delivery path) are the following.

The number of nodes on spam delivery paths is small and decreased over the 3 year time span. For example, consider the portion of path from the (claimed) origin to the first internal mail server of the recipient network, the average number of nodes on the paths in 2007 is 2.57, which is decreased to 2.34 in 2008 and 2009. Moreover, consider the same portion of the paths, about 45%, 68%, 66% of spam emails have a path of only two hops in 2007, 2008, and 2009, respectively. That is, such emails were directly delivered from the spam originating machines to the recipient side mail servers, without any attempt to fake the spam delivery paths. Such emails were likely sent from compromised machines or members of spamming botnets [22, 4]. Although it is tempting to argue that such spammers do not fake the trace information because they are not concerned with a spamming botnet member being identified, as we will discuss in Section 4, it is hard, if not impossible, for such spammers to hide the true origin even if they fake the trace information.

Our investigation of the MX records of the 1.2 M network domains shows that the majority (90%) of domains only have mail servers in one domain, which means that the majority of network domains on the Internet today do not need a third party to provide the backup relay service; emails destined to these domains should be directly delivered to their own mail servers. The trend of using mail servers in a single domain helps shorten the path that an email traverses from the sender domain to the recipient domain, and makes it hard for spammers to create forged but undetectable trace information.

Our findings have important implications on a broad range of spam control efforts, including email sender authentication schemes and spam filtering. It also helps guide the efforts of law enforcement on investigating email-based crimes.

CEAS 2010 - Seventh annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference July 13-14, 2010, Redmond, Washington, US

As an example, given the short delivery path of emails on the Internet, we may need to re-examine the efforts in developing domain-level signature-based sender authentication schemes [11]. We discuss the detailed implications in Section 6.

The remainder of the paper is organized as follows. In Section 2 we provide the background on email delivery and message format, and discuss the related work. In Section 3 we describe the data sets and the main analysis methodology used in the studies. We study the properties of spam delivery paths in Section 4, and the properties of the MX records of the network domains in Section 5. We discuss the implications of our findings and the limitations of our studies in Section 6. We conclude the paper and discuss future work in Section 7.

2. BACKGROUND AND RELATED WORK

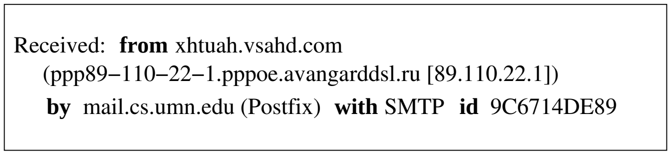
In this section we first provide some background on the Internet email delivery and the message format that are most relevant to the current work (see [13, 16] for a complete treatment). We then briefly discuss the related work.

2.1 Background

The Internet email system consists of two types of machines: Mail User Agents (MUAs) and Mail Transfer Agents (MTAs). MUAs are end user machines where a message is composed and read, and MTAs are mail servers that deliver messages from senders to recipients using the Simple Mail Transfer Protocol (SMTP) [13]. On the way from sender to recipient, a message may traverse a number of intermediate MTAs. From the MTA's perspective, a message contains two pieces of information: a message envelope and a message content. The message content in turn contains a message header and a message body.

MTAs rely on the message envelope (not the message header) to forward a message. More specifically, an MTA uses the `RCPT TO` envelope address instead of the `To:` header field to determine the next MTA to which an email should be forwarded. Recipients only see the message content (header and body); they may not have the complete information of the envelope. The `RCPT TO` envelope address may be included in the `Delivered-To:` header field inserted by the last MTA before an email is delivered into the recipient's inbox.

The design of both SMTP and the Internet message format [16] presents great flexibility but weak security; as a consequence, almost all message header fields can be faked. According to the Internet message format standard [16], only two header fields are required: `Date:` and `From:`. All other header fields are optional. (Technically, the trace header fields `Return-Path:` and `Received:` are also required fields if a message traverses an MTA. SMTP specifies that an MTA must insert a `Received:` field, and the last MTA must insert the `Return-Path:` field.) Moreover, almost all the header fields can be faked, including the two required fields. The only fields that cannot be (completely) faked are the ones inserted by the last MTA such as `Return-Path:` and `Delivered-To:`, and the `Received:` fields inserted by legitimate MTAs on the path. It is important to note that, some of these fields may also contain false information. The `Return-Path:` contains the envelope `MAIL FROM` address; however, a spammer can easily supply a false address. The `Delivered-To:` contains the envelope `RCPT TO` address, which must be correct (otherwise, the message cannot be delivered to the intended recipient). Note that `Delivered-To:` is an optional field; some mail servers including a major mail service provider do not support this feature.



Received: **from** xhtuah.vsahd.com
(ppp89−110−22−1.pppoe.avangarddsl.ru [89.110.22.1])
by mail.cs.umn.edu (Postfix) **with** SMTP **id** 9C6714DE89

Figure 1: An example `Received:` header field.

2.1.1 `Received:` *Header Fields*

As a message traverses an MTA, a trace record `Received:` header field is prepended to the message header (see Figure 1 for an example `Received:` header field). A `Received:` field contains two required clauses `from` and `by`, and a few optional clauses including `with` and `id`. Let $m_1, m_2, \ldots, m_k$ be the sequence of MTA servers that a message traverses (in that order), and hr_i be the corresponding `Received:` field inserted by m_i for $i = 1, 2, \ldots, k$. For convenience, we let m_0 denote the submission machine where the message is composed. Then in hr_i, the `from` clause specifies the upstream MTA m_{i-1}, and the `by` clause the current MTA server m_i.

The `from` clause contains two parts: the name of the sending machine as specified in the SMTP EHLO command, and the host name and IP address of the sending machine as obtained from the TCP connection (more precisely, the host name is obtained via a reverse DNS lookup based on the IP address obtained from the TCP connection). Using the common convention [20], we refer to the host name specified in the EHLO command as the `from-from` field, the host name and IP address obtained from the TCP connection as the `from-domain` and `from-address`, respectively. In general, the `from-from` host name may not be reliable. However, the `from-address` and `from-domain` should be correct if they are inserted by a legitimate mail server. The `by` clause in general only contains the domain name of the current MTA (not IP address), and we refer to it as the `by-domain`. In the example `Received:` field in Figure 1, the `from-from` domain name is `xhtuah.vsahd.com`, the `from-domain` and `from-address` are `ppp89-110-22-1.pppoe.avangarddsl.ru`, and `89.110.22.1`, respectively. The `by-domain` is `mail.cs.umn.edu`.

2.2 Related Work

To the best of our knowledge, there has been no systematic study on the forgery behavior of spammers. In the following we briefly discuss the related work in behavioral characteristics of spammers, email sender authentication, and spam filtering.

A number of studies investigated the behavioral characteristics of spammers at the mail-server level and network level [5, 15], including the number of messages from each mail server, each network domain, and the number of mail servers in each domain, among others. These studies were based on the IP addresses of the first external MTA (i.e., the mail servers that forwarded a message into the recipient network) as observed by the receiving domain. Gomes *et al.* studied the characteristics of spam traffic to identify the features that can distinguish spam from legitimate messages [6]. They found that key email workload aspects including the email arrival process, email size distribution, and distributions of popularity and temporal locality of email recipients can distinguish spam from legitimate messages. These studies did not investigate the forgery properties of the trace information in spam.

Given the proliferation of email address spoofing employed in spam and phishing messages, there have been a number of research and development efforts to improve the email sender authentication situation on the Internet, including Sender Policy Framework (SPF), SenderID, DomainKeys, and DomainKeys Identified Mail (DKIM) [21, 14, 3, 2]. In SPF and SenderID, a special record is published in the DNS database of a network, which specifies the legitimate MTA IP addresses for the domain. A receiver mail server supporting this feature can verify if the sending machine is a legitimate mail server to send messages on behalf of the claimed sender.

In DomainKeys and DKIM, a public key is published using the DNS service, and all outgoing messages are signed using the private key of the domain. A receiving domain can retrieve the public key from the sending domain to verify if the signature carried in the message header is valid to determine if the message is from the claimed sender. We refer to the SPF-like schemes as the mail-server-level authentication schemes (MSLA), and the DKIM-like schemes as the domain-level signature-based authentication schemes (DLSA).

Numerous email spam filters have also been developed [8, 19, 17]. They try to identify spam based on either the content of a message (content-based filters) or the IP addresses or domains of the sending machines (IP-address-based filters), so that users will not spend much time on processing these messages. We discuss the implications of our findings on sender authentication and spam filtering in Section 6.

3. DATA SETS AND ANALYSIS METHODOLOGY

In this section we first describe the data sets we use in analyzing and understanding the forgery properties of spam delivery paths, and then we discuss the methodology for analyzing the data.

3.1 Data Sets

We use two complementary data sets in this study. The first one is a 3 year spam archive with a time span from 2007 to 2009 [9]. This spam archive contains about 1.85 M spam messages in total. We exclude the ones that we cannot use in this study, including the ones that do not have the `Received:` header fields, or do not have any message headers at all. (These cases are likely caused by archiving software or hardware errors.) After removing these messages, the spam archive has about 1.84 M messages. To ease exposition, we refer to the set of emails we can use as the *spam archive*. Table 1 shows the summary of the spam archive. In this table, we also show the number of unique "bait" email addresses and domains used in collecting the spam archive, which were obtained from the `Delivered-To:` fields in the spam archive. Recall that `Delivered-To:` is an optional field. In the spam archive, about 90.5% of messsages contain the `Delivered-To:` field.

Table 1: Number of messages in spam archive.

Year	2007	2008	2009	Total
# of spam	316,746	722,579	802,986	1,842,311
# of bait addr.	45	66	71	90
# of bait domains	7	7	10	10

Table 2: Number of network domains in MX dataset (2008).

Duration	# of domains	# of gTLDs	# of ccTLDs
10/01 – 10/15	1,224,819	19	228

The second data set we use in this study is the mail exchanger (MX) records of a set of network domains. (MX is a resource record specifying mail server(s) responsible for accepting emails on behalf of the corresponding network domain.) In the following we describe how this data set is collected. An email trace was collected at a number of mail servers deployed in the Florida State University (FSU) campus network between 10/01/2008 and 10/15/2008 (inclusive). During the course of the email trace collection, the mail servers received about 53 M messages destined for 82 sub-domains in the FSU campus network, of which about 47 M, or about 88.7%, are spam.

We extract the network domains of the sender's envelope (`MAIL FROM`) email addresses from the email log files. In this way we obtain about 1.8 M distinct network domains. Note that the sender envelope email addresses, and hence, the network domains, can be easily faked. However, our purpose here is to get a relatively large set of representative network domains to study their MX records. Therefore, faked but existing network domains will not affect our study. From the 1.8 M network domains, we can obtain the MX records for about 1.2 M network domains. (The MX records were obtained in July 2009.) The others are either faked non-existing domains, or domains not having an MX record.

We refer to the network domains for which we can retrieve the MX records as the *MX dataset*. To simplify notation, we may also refer to the MX records of these domains as the *MX dataset*, when there is no confusion. As one way to examine the representativeness of the MX dataset, we extract the top level domain (TLD) of each network domain in the dataset. There are 247 unique TLDs in the dataset, which include all the generic TLDs (gTLDs), except `.tel`, and 228 country-code TLDs (ccTLDs) [12]. In addition, we have also manually checked that the MX dataset contains all the major email service providers, including Yahoo! Mail, Hotmail, Gmail, etc. Table 2 shows the summary of the MX dataset.

3.2 Analysis Methodology

In this subsection we describe the main methods we use in analyzing the data sets. We start with the ones on the spam archive dataset.

Given a message m, we let $m_0, m_1, \dots, m_k$ denote the sequence of MTAs included in the `Received:` header fields in that order. Note that m_0 could be the sender MUA instead of an MTA; we do not distinguish them in our study. The sequence of $m_0, m_1, \dots, m_k$ is referred to as the email delivery path. (For a spam email, the sequence is referred to as the spam delivery path.) We let hr_i be the corresponding `Received:` field inserted by m_i for $i = 1, 2, \dots, k$.

3.2.1 First External and Internal MTA Servers

Consider a message m with $m_0, m_1, \dots, m_k$ as the se-

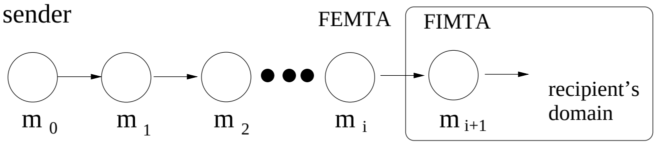


Figure 2: Illustration of message path, FEMTA, and FIMTA.

Algorithm 1 Identifying first external MTA server

```
1: Given a message $m$
2: Extract Received: fields $hr_i$ $(i = 1, 2, \ldots, k)$
3: d = Recipient's network domain
4: mx = set of MX mail servers of domain d
5: // to simplify, mx also contains domains of the servers
6: for ($i = k$ to 1) do
7:     if ($hr_i$'s by-domain $\in$ mx) then
8:         break
9:     end if
10: end for
11: for ($j = i$ to 1) do
12:     if ($hr_j$'s from-address $\notin$ mx) then
13:         break
14:     end if
15: end for
16: Return from-address of $hr_j$
```

quence of nodes on the delivery path of m (see Figure 2). For $i = k, k-1, \ldots, 1, 0$, if m_i is the first MTA that does not belong to the recipient's domain, then m_i is referred to as the *first external MTA server* (FEMTA), and m_{i+1} as the *first internal MTA server* (FIMTA). FEMTA plays a critical role in the investigation of spam origins, as it is the only external MTA whose information we can reliably retrieve, based on the `from-address` or `from-domain` inserted by FIMTA, which we can trust. However, given an email, it is not always trivial to reliably identify which node is FEMTA. In the following we discuss the algorithm we use to identify FEMTA of an email, which is developed based on the algorithm proposed in [7].

Algorithm 1 summarizes the algorithm to identify FEMTA. Given a message m, we first extract all the `Received:` header fields hr_i, for $i = 1, 2, \ldots, k$. We then extract the recipient's network domain d. We extract d in the following manner. If `Delivered-To:` field exists, we assign d the domain of email address contained in this field. Otherwise, we let d be the network domain of the `by-domain` of the last `Received:` field (hr_k). Note that we do not use the `To:` fields to extract the domain information, as it can be faked. After identifying the network domain d of the recipient, we look up the MX records of d (line 4), and store the set of mail servers in mx. We then search all the `Received:` fields hr backwards starting from the recipient side, until we encounter an hr_i, whose `by-domain` is either in mx or matching the domain of a mail server in mx (lines 6 to 10). After we identify hr_i, we continue searching until we find the first hr_j, whose `from-address` is not either in mx or matching the domain of a mail server in mx (lines 11 to 15), which is taken as the FEMTA (line 16).

We identify the FEMTA in this way for the following reason. Certain network domains rely on a third-party network to provide the mail service. For example, one of the "bait" domain used in the spam archive is `untroubled.org`, and its mail server is `mx.futurequest.net`. Therefore, we cannot use the simple heuristic that the first m_i (for $i = k, k-1, \ldots, 0$) whose `from-domain` is not `untroubled.org` as FEMTA. After FEMTA is determined, FIMTA can be obtained trivially, which is the downstream MTA along the path from sender to recipient.

3.2.2 Length of Delivery Path

A key property of the trace information of an spam email is the number of nodes on the spam delivery path, or the path length. In order to eliminate the impacts of the internal message delivery structure in different recipient domains, we only consider the portion of the path up to the FIMTA. More precisely, let $m_0, m_1, \ldots, m_k$ be the nodes on the path of a message m, and m_j the FIMTA, then we exclude the nodes from m_{j+1} to m_k in the calculation of the path length. We consider two different paths in our study. The first one is *raw path*, that is, the path from m_0 to m_j. Note that, this portion of path may contain forged nodes. The motivation of using raw paths is to investigate, without removing any invalid nodes, how the trace information looks like in spam emails.

The second type of paths we consider is the *network-level consistent* (NLC) paths, which is defined as follows. Let hr_i $(i = 2, 3, \ldots, k)$ be the `Received:` header fields of message m (hr_k is inserted by the last MTA at the recipient side). Let f_i and b_i be the MTA servers specified in the `from-domain` and `by-domain` of hr_i, respectively. That is, b_i is the current mail server, and f_i is the upstream mail server along the path. Then f_i and b_{i-1} should be the same machine. Typically, they have the same IP address; however, some mail servers may have multiple IP addresses, and the incoming process and forwarding process may use different IP addresses. Therefore, f_i and b_{i-1} may have different IP addresses. If f_i and b_{i-1} have different IP addresses, but they are in the same network prefix, then we say f_i and b_{i-1} are network-level consistent (NLC). If f_i and b_{i-1} do not have the same network prefix, they are not network-level consistent. For simplicity, we only consider the /16 network prefix in this study.

An NLC path is a portion of a raw path, which satisfies the condition that all f_i and b_{i-1} should be NLC. Formally, let hr_i be the first `Received:` header field whose f_i is not NLC with b_{i-1}, then the corresponding NLC path is from m_{i-1} to m_j (the first internal MTA server). We refer to m_{i-1} as the NLC origin of the message. Note that some `by-domain`'s do not contain a host name (but only the domain name) and cannot be mapped into an IP address. This is used by certain legitimate mail servers. To incorporate this situation, we consider f_i and b_{i-1} to be NLC if they are in the same network domain based on their domain names, even if we cannot obtain the IP address of b_{i-1}. NLC paths enable us to understand to what degree a simple heuristic can help to eliminate obvious forgeries on spam delivery paths.

3.2.3 Studies on MX Dataset

The Internet email system was designed in the early days of the Internet developments, at which time the Internet connectivity was not reliable. For this reason, many network domains needed others to provide the backup relay service for the domains. In case a sender cannot directly connect to the mail server of the recipient domain, it can

Algorithm 2 Determining domain of host name

```
Given a hostname $h$
if ($length(tld) > 2$) then
    $n = 2$
    // except $h$ in domain $dyndns.biz$, for which $n = 3$
else
    // $length(tld) == 2$
    if ($sld == ac|co|gv|or|tm|com|edu|gov|nom|org$
        $|asso|gouv|info|priv|sport$) then
        $n = 3$
    else
        $n = 2$
    end if
end if
Return the last $n$ part of $h$
```

first send the message to one of the backup relay servers, which hopefully will have a better connectivity to the recipient domain. However, given the relatively high reliability of the current Internet connectivity, the backup relay service is rarely needed. Furthermore, backup relay service for a domain *d*, if not carefully configured, may provide open-relay service to all domains. This can be exploited by spammers to hide their true origins. For these reasons, backup mail relay service is not commonly used on today's Internet. On the other hand, due to the ever-increasing volume of spam messages that a domain has to handle, many network domains deploy an increasing number of mail servers in their networks (or outsource all mail service to a third-party provider) to balance the load of incoming messages.

We primarily perform two kinds of studies on the MX dataset. The first one is the total number of mail servers of each domain as shown in the MX records (in contrast to the number of mail sending machines as observed in incoming messages used in a number of studies [15, 5]). Given that certain domains may deploy many mail servers in one domain to balance the load, the total number of mail servers for a domain does not give us the complete picture of the impacts of mail server number on spammers' forgery behavior. For this purpose, we cluster the mail servers of a domain into groups, each containing all the mail servers with the same domain name. For example, `fsu.edu` has 11 mail servers listed in the MX record. After clustering, it has only one domain. To ease exposition, we refer to each group as a *mail-server cluster*. The second metric we study on the MX dataset is the number of mail-server clusters of each domain.

Given the flexibility in domain name registration, it is not always easy to determine the domain of a given mail server. We use the following heuristic to determine the domain part of a host name (Algorithm 2). Given a host name *h*, let *tld* and *sld* denote the top-level domain and second-level domain of *h*, respectively. If the string of *tld* is longer than 2 (such as `.edu`), we consider the top two level domain as the domain of *h*. For example, the domain of mail server `ms1.ucs.fsu.edu` is taken as `fsu.edu`. One exception is when *h*'s top two level domain is `dyndns.biz`, for which we take the top three level domain as the domain. `dyndns.biz` provides dynamic network services. The sub-domains under this domain likely belong to different entities.

When the string length of *tld* is two (for example, `.cn`), we in general take the two top level domains as the domain of *h*, except when *sld* matches a few patterns such as *co* and *com*, in these cases, we take the top three level domains as the domain of *h*. For example, we take `cctv6.com.cn` as the domain for mail server `mail.cctv6.com.cn` (its *sld* matches *com*), while `sae-cctv.cn` as the domain for mail server `mail.sae-cctv.cn` (its *sld* does not match any specified patterns). The set of *sld* patterns are obtained by an ad hoc examination of a large set of top-level domains with a length of two that appear in the MX dataset.

The above heuristic is not perfect; however, it should be reasonably accurate when it is applied to the MX dataset. More importantly, it is rather conservative in the sense that, it may fail to group two host names belonging to the same domain into a cluster, but it will not group two host names not belonging to the same domain into a cluster. Put in another way, our heuristic may generate more mail-server clusters than a domain may have, but it will not generate fewer mail-server clusters.

4. SPAM DELIVERY PATHS

In this section we study the properties of spam delivery paths, including the path lengths and the network-level distribution of the first external MTA. We postpone the study on the naming structure of the first external MTA to the next section, where we also study the naming structure of the mail servers in the MX dataset.

4.1 Path Length

We first study the length distribution of spam delivery paths, which illustrates to what degree spammers try to insert forged trace information to hide the true origin of spam emails. Figure 3 shows the cumulative distribution function (CDF) of the raw path lengths. Note that the raw path of a message is the portion of the path from the first internal MTA to the (claimed) origin of the message, including all the nodes that the message claims to have traversed as recorded in the `Received:` fields. From the figure we see that, about 45% of messages in year 2007 had two nodes in the trace records. They were directly delivered from the originating machine to the recipient domain, without any attempt to forge the trace records. Importantly, the percentage of these messages increased to 68% and 66% for 2008 and 2009, respectively. One possible reason is that, these messages were sent from spamming botnets, and spammers were increasingly relying on botnets to send spam over the years. The average length of raw paths also decreased over the years, from 2.57 in 2007 to 2.34 in 2008 and 2009.

In order to understand how simple heuristics can help identify the forged trace records, Figure 4 shows the CDF of the length of network-level consistent (NLC) paths. As we can see from the figure, simple heuristics such as NLC can help eliminate a large portion of forged trace records. Compared to Figure 3, we now have about 91%, 97.1%, 96.7% of messages that had only two nodes in the trace records in years 2007, 2008, and 2009, respectively. A manual inspection of some sample messages reveals that, a number of factors contributed to the detection of forged trace records. First, a possible mistake in the spamming software confused the `from-domain` and the `by-domain` when it tried to insert forged trace records. Second, some spam emails inserted a forged `Received:` header field trying to show that the email was originated in a third-party network and had been received by the destination network domain. A more detailed investigation and classification of non-NLC paths is left to

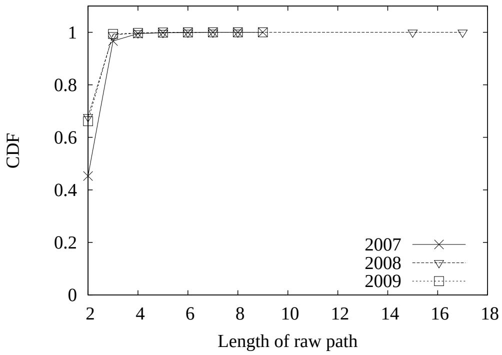


Figure 3: Length of raw paths.

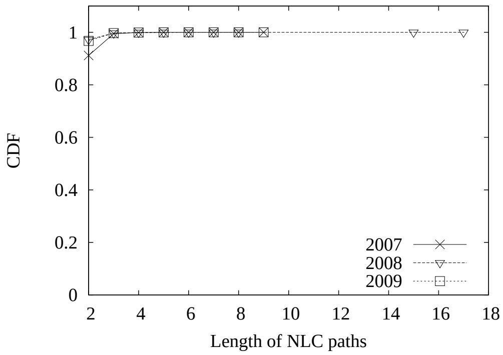


Figure 4: Length of NLC paths.

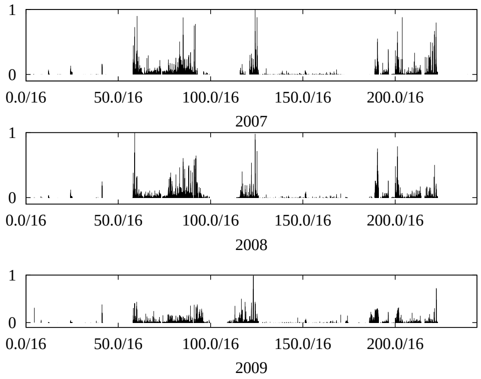


Figure 5: Normalized number of spam delivered from each /16 address space.

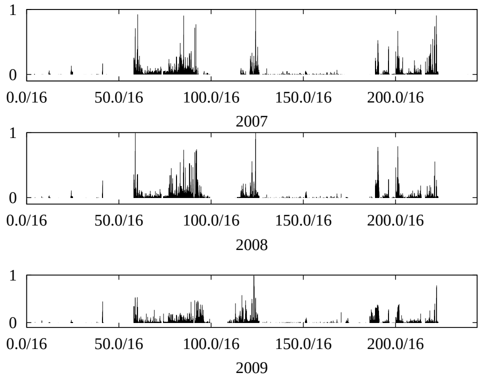


Figure 6: Normalized number of FEMTA IP addresses in each /16 address space.

future work.

From this study we can see that, in recent years spammers seldom attempted to hide the originating machine of a message by forging the trace records of the message. Even when they attempted to do so, we can identify a large part of forged trace records, using simple heuristics (this observation may only apply to the spam archive we use in this study). We discuss the reasons and implications of this behavior in the next section, where we also discuss the limitations of our datasets and methodology.

4.2 Network-Level Distribution

In order to understand the distribution of the messages and the first external MTAs (FEMTAs) across the IP address space, we extract the IP addresses of FEMTAs and the number of messages delivered from each FEMTA. We then classify the messages and IP addresses into each /16 address space (messages are classified based on the IP addresses of FEMTAs).

Figure 5 shows the normalized number of messages from each /16 address space (normalized by the maximum number of messages from a /16 space in each year). From the figure we can see that spam messages were largely delivered from FEMTAs located in three concentrated address space regions. Figure 6 shows the normalized number of FEMTA IP addresses in each /16 address space (normalized by the maximum number of FEMTA IP addresses in a /16 space in each year). From the figure we can see that FEMTAs were also largely located in three concentrated address space regions. Note that the three regions in Figure 6 largely overlap with the three regions in Figure 5. It is also worth noting that the three concentrated address space regions are consistent with what were reported in [5]. To a degree, this confirms the representativeness of the spam archive we use in studying the behavior of spammers.

5. NETWORK DOMAIN MX RECORDS

In this section we study the properties of the MX dataset, in particular, the number of mail servers and mail-server clusters for each network domain. We discuss the implications of the findings in the next section.

Figure 7 shows the cumulative distribution function (CDF) of the total number of mail servers for each domain in the MX dataset. Among the about 1.2 M network domains in the dataset, about 0.7 M domains, or 57% of them only have one single mail server in the MX records. These domains do not have any backup relay services; a normal message destined to recipients in these domains should be directly delivered to the corresponding mail servers, resulting in short delivery path from sender to recipient. Some of the domains have a relatively large number of mail servers. The highest number of mail servers is 94, belonging to zartana.com, which appears to be an online marketing company.

Note that, some of the domains with only a single mail server outsource their mail service all together to a third-party provider. They do not have any mail servers in their own network domains. For example, gturo.com has two

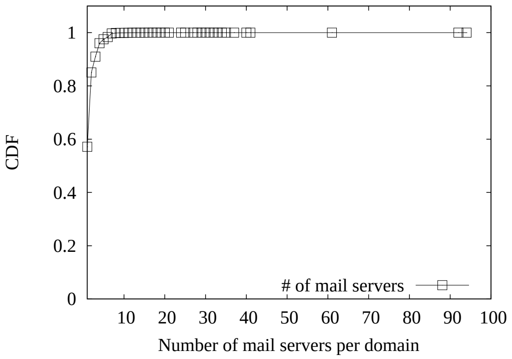


Figure 7: # of total mail servers for each domain.

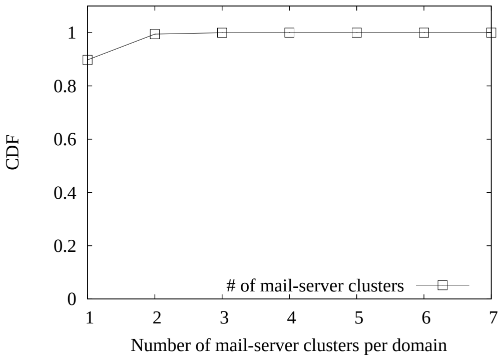


Figure 8: # of mail-server clusters for each domain.

mail servers, both located in the domain `yahoo.com`. This outsourcing arrangement is different from the *backup* relay service from a third party. In the backup relay service, the domain itself also has public mail servers, in addition to the backup relay servers in the third party. For example, `bemac.com` has three mail servers, located in two domains `bemac.com` and `psi.net`. Backup relay service could potentially increase the message delivery path, while outsourcing mail service (to a single provider) still results in a (publicly perceived) short email delivery path.

Figure 8 shows the CDF of the number of mail-server clusters for each domain in the MX dataset. Recall that a mail-server cluster is the set of all mail servers with the same domain name. As we can see from the figure, about 90% of domains only have mail servers in one domain (and about 99% with at most two mail-server clusters). Note that multiple mail servers (as listed in MX records) in the same domain are normally used for load-balancing purpose; they do not provide relay service between themselves. Hence, multiple mail servers in the same domain normally will not increase the path that a message needs to traverse. This is in contrast to the backup relay service, when a domain has multiple mail-server clusters. In this case, a message may be first delivered to one of the backup relay servers (in one cluster) and then forwarded by the backup relay server into a different cluster (hopefully the final destination domain). This relaying process will result in a longer delivery path. The domain with the largest number of mail-server clusters is `mcilmoil.com`, which has 7 clusters (and each cluster only has one mail server).

5.1 Naming Structure

In order to understand the naming convention of mail servers in the MX dataset, we develop a simple method; we simply extract the local name from the host names of the mail servers in the MX dataset. For example, the local name of mail server `mail.cs.fsu.edu` is `mail`. Note that more complicated methods can be developed to look into the patterns of host names of mail servers. There are totally $966,397$ distinct host names of mail servers in the MX dataset. We extract the local names from this set of host names, and there are totally $220,970$ distinct local names. We show the top 10 local names in Table 3. The local names are ranked according to the number of host names containing a particular local name. In the table we also show the number and fraction of host names with a particular local name. For example, $330,553$, or about 34%, of the host names have `mail` as the local name. As we can see from the table, a large portion of mail servers have a host name that contains `mail`, `mx`, `smtp`, etc., from which we can infer if the machine supports the mail service.

Table 3: Top 10 local names in MX dataset (# of host names: $966,397$).

Local name	# of host names	Fraction of host names
mail	$330,553$	0.342047
mail2	$23,665$	0.024488
mx	$20,602$	0.021318
smtp	$17,179$	0.017776
mx1	$14,218$	0.014712
mail1	$10,578$	0.010946
mx2	$9,800$	0.010141
inbound	$7,666$	0.007933
mail3	$4,294$	0.004443
www	$3,512$	0.003634

To understand the naming structure of the first external MTAs (FEMTAs) in the spam archive, we extract the `from-domain` of the FEMTAs. If an FEMTA does not have `from-domain` in the spam message, we treat it as the FEMTA did not have a host name at the time when the message was delivered. In this case, we extract the `from-ip` of the FEMTA. Note that, an FEMTA must have at least the `from-ip` in the message. Note also that we do not use the `from-from` domain name (i.e., the EHLO domain name), given that it can be easily faked.

We process the host names of FEMTAs in the spam archive in a similar manner as we process the mail-server host names in the MX dataset. We simply extract the local names from the host names of FEMTAs, but with three exceptions to better group the local names. First, some host names contain an IP address at the beginning of the host names, for example, `154.88.218.87.dynamic.jazztel.es`, or `83-131-12-156.adsl.net.t-com.hr`. For these host names, we extract the IP addresses and group them into two special local names `a.b.c.d` and `a-b-c-d`, respectively. A third special case we handle is the host names that start with a letter string followed by an IP address (components of an IP address are separated by a dash "-"), as in the example, `oh-71-50-221-149.dyn.embarqhsd.net`. For these domain names, we extract the IP addresses and group them into a

Table 4: Top 10 local names of FEMTA in spam archive (# of host names: 183, 296).

Local name	# of host names	Fraction of host names
a-b-c-d	30, 745	0.167734
xyz-a-b-c-d	25, 027	0.136539
a.b.c.d	20, 333	0.110930
mail	2, 038	0.011119
dsl	1, 909	0.010415
dsl88	1, 324	0.007223
client-201	914	0.004986
ppp-124	877	0.004785
ppp-58	730	0.003983
triband-mum-59	519	0.002831

special local name `xyz-a-b-c-d`. We single out these three simple special cases just trying to better group the local names of FEMTAs. Otherwise, there is no particular dominating local names.

There are totally 183, 296 distinct host names and 153, 078 IP addresses of the FEMTAs in the spam achieve dataset. Put in another way, about 45.5% of FEMTAs only had IP addresses and did not have a domain name. This is a salient feature of FEMTAs compared to the mail servers in the MX dataset. We extract the local names from this set of host names, and show the top 10 local names in Table 4. Similarly, in the table we also show the number and fraction of host names with a particular local name. For example, 30, 745, or about 17%, of the host names have the pattern `a-b-c-d` as the local name. If we combine the top 3 local names (patterns), we can see that a large portion (41.5%) of FEMTAs contains IP addresses at the beginning of their host names. These machines are likely to be home user machines on broadband or dial-up networks. By a manual examination of a sample set of host names of FEMTAs, a large portion of host names contain keywords such as `dsl`, `ppp`, `dynamic` (albeit not necessarily at the beginning of a host name). Note that local name `mail` is ranked 4th in the top 10 list, but with a very small fraction compared to the top 3 local names. This indicates certain spam messages were delivered from mail servers, which could be (open-relay) mail servers of some providers, or mail servers under the control of spammers.

6. IMPLICATIONS AND DISCUSSION

In this section we discuss the implications of our findings on email sender authentication and spam control. We then discuss the limitations of our datasets and analysis methodology.

6.1 Implications on Email Sender Authentication

Given that normal messages are composed on an MUA and then delivered to the sender side MTA before being delivered to the recipient side MTA, they normally need to traverse three nodes. As we have shown in Section 4, a large portion of spam messages have a path of only two nodes. These messages were directly delivered from the spam originating machines to the recipients' mail servers. Based on this, we can infer that most of these sending machines are compromised machines. It is tempting to argue that spammers do not hide the identities of these originating machines (by forging the trace records) because they are compromised machines; however, a closer look reveals that, even if they fake the trace records of the messages sent from compromised machines, they cannot hide the fact that most of these machines are end-user machines, which in general do not serve as mail servers.

Moreover, as we have shown in the previous section, the majority (90%) of network domains only have one mail-server cluster, which also helps to reduce the email delivery path. Regular messages should be directly delivered from the sender domain into the recipient domain. Two legitimate reasons that may increase the path length of a message are mailing lists and message forwarding mechanisms such as the one realized using the `.forward` file on Unix. However, most of mailing lists are moderated nowadays, and moreover, a message to a mailing list should also be directly delivered from the originating domain to the mailing list. Similarly, in the case of `.forward`-like message forwarding scheme, the message should also be directly delivered from the sender domain to the mail server supporting the message forwarding function. In addition, large mail service providers such as Gmail support a message *pull* function, where messages in different accounts can be retrieved to a central account instead of being forwarded from different accounts. This method will not increase the path length of messages. Spammers also try to use mail open-relay servers to hide the originating machines, which can increase the path length. However, as open-relay servers are being actively detected and blocked, they become less useful for spammers.

In summary, we can conclude that the majority of messages (both spam and non-spam), if not all, will be directly delivered from the sender domain to the recipient domain on the Internet. Given this observation, we may need to re-examine the efforts on developing email sender authentication schemes, in particular, the domain-level signature-based authentication schemes such as DKIM. When a message only traverses two domains, we can reliably identify the sender domain, and schemes like DKIM may not be needed. This is particular the case, if mailing lists and `.forward`-like mail forwarding servers can help verify the email sender domain before accepting a message into the system. Note that, similarly, when backup relay service is involved in the delivery of a message, the backup relay server can also help verify the email sender domain before accepting a message for the destination domain.

Given that the majority of spam messages were sent from spamming botnets, mail-server-level authentication (MSLA) schemes such as SPF will continue to play a critical role in blocking spam messages. As pointed out in [5], although spammers can easily turn a compromised machine into a spam mail server, it is much harder for them to fake it as a *legitimate* mail server for a particular network domain. Therefore, MSLA schemes such as SPF can be very effective in blocking messages from compromised machines (see the next subsection on more discussions on spam control).

6.2 Implications on Spam Control

As we have discussed, the majority of spam messages were sent from compromised end-user machines. An effective way to filter spam messages is to distinguish *client IP addresses* from *server IP addresses* and block messages sent from client IP addresses in a remote domain. This is partially the goal of numerous DNSBLs [19, 18] and MSLA schemes such as SPF. Similarly, in the context of countering DoS attacks, Handley and Greenhalgh proposed seven steps in building DoS-resistant Internet architecture [10], and the first step is to separate client and server IP addresses. This proposal was developed based on the observation that worms are propagated among (and infect) the machines with the same vulnerabilities, normally between client machines or between server machines.

Dividing the global IP address space into two different sets of client and server addresses may not be realistic on the current Internet. However, a practical approach to achieving this goal is through community efforts on a common naming structure of host names. In the context of the Internet email system, we have shown that a large portion of mail servers have `mail`, `mx`, and `smtp` as their local names. A manual inspection of the mail server host names reveals that, a much larger portion of mail servers have these kinds of keywords in their host names (albeit not necessarily in the local names). A community effort could be to develop a common naming structure on the local name of mail servers, e.g., all with *mail*, while providing naming flexibility in other parts of the host names, to facilitate the separation of mail servers from end user machines. In this way we can easily identify the client machines and block the messages from them automatically.

Note that a number of free DNS service providers allow users to register any (*but non-conflicting*) host names for machines [23]. However, this service will not help spammers mislead the type of machines, even if spammers register a host name following the common naming structure for a compromised machine. Note that, this service only supports *forward* DNS lookup (mapping host names into IP addresses), but not the *reverse* DNS lookup (mapping IP addresses into host names). Mail servers use reverse DNS lookup to determine the host name of a sending machine.

It is worth noting that, using reverse DNS lookup to determine the host name (and hence the machine type) will not introduce any additional overhead to the mail servers. Currently mail servers already perform the reverse DNS lookup and insert the host name of a sending machine into the trace records. In order to effectively block the spam delivery from end-user machines in a remote domain, the reverse DNS lookup should be performed early in the delivery transaction, ideally immediately after the TCP connection is established (a lookup earlier than this would require the update of the kernel TCP/IP stack). Note also that based on our study on the naming structure of FEMTAs in Section 5, mail servers can start adopting this method to block spam messages delivery from a large portion of end-user machines today. Applying the same principle to the context of DoS control will introduce additional overhead, given that, for example, current web servers do not necessarily perform the reserve DNS lookup to determine the host name of a requesting machine. However, this additional overhead should be negligible from the users' perspective.

Our findings also have other important implications for spam control. For example, content-based spam filters can improve their performance by simple heuristics to detect obvious forged trace records or by examining the structure of trace records, in particular, the length of the trace records. Indeed, the trace records provide valuable information for us to further explore. For example, currently a number of DNSBLs actively probe the Internet as an effort to detect and block mail open-relay servers. A more targeted scheme with less impact on the global Internet is to probe all the nodes on the trace records to examine if any of them is an open relay.

6.3 Discussion

In this subsection we discuss the limitation of our datasets and methodology. One limitation is the spam archive we use in the study, which only provides a small sample of the spamming behavior on the Internet. Diverse spam traces from different vantage points are needed in confirming the findings in this study. However, if spammers do not treat the "bait" addresses used in collecting the spam archive differently from other addresses, the spam archive we use should be representative, and our main findings, in particular, the short path length of spam messages, should be valid. This is cross-validated to a degree using the MX dataset. In addition, the distribution of FEMTA IP addresses is consistent with the findings in other work with a different email trace [5], which also to a degree confirmed the representativeness of the spam archive we use in studying the behavior of spammers. In addition to investigating the path properties of spam messages, it is also important to examine the delivery path properties of non-spam messages, which we plan to do in our future work.

In identifying the forged trace records, we try to be conservative. We adopt this conservative approach because our main objective is to study the path length, and we do not want to under-estimate this value. More aggressive approaches can be adopted by spam filters. In our study of naming structure of host names, we also adopt a conservative approach by only examining the local name of a host name. A more systematic study should be performed by examining other parts of a host name to identify the naming patterns in both mail servers and end-user machines.

7. CONCLUSION AND FUTURE WORK

In this paper, we performed the first comprehensive study on the trace record structure of spam messages to investigate, among others, to what degree spammers *can and do* forge the trace records of spam messages. We performed the study based on two complementary data sets: a 3-year spam archive with about 1.84 M spam messages, and the `MX` records of about 1.2 M network domains. In addition to presenting the findings, we also discussed the implications of the findings on various spam control efforts. As future work we will cross validate the findings using other spam archives; we will also develop more systematic approaches to investigating the inconsistency in spam delivery paths, in addition to the network-level path consistency used in this paper.

8. REFERENCES

[1] S. Aggarwal, J. Bali, Z. Duan, L. Kermes, W. Liu, S. Sahai, and Z. Zhu. The design and development of an undercover multipurpose anti-spoofing kit (unmask). In *23rd Annual Computer Security Applications Conference (ACSAC)*, Miami Beach, FL, Dec. 2007.

[2] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. DomainKeys Identified Mail (DKIM) Signatures. RFC 4871, May 2007.

[3] M. Delany. Domain-based email authentication using public-keys advertised in the DNS (domainkeys). Internet Draft, Aug. 2004. Work in Progress.

[4] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker. Detecting spam zombies by monitoring outgoing messages. In *Proc. IEEE INFOCOM*, Apr. 2009.

[5] Z. Duan, K. Gopalan, and X. Yuan. Behavioral characteristics of spammers and their network reachability properties. In *IEEE International Conference on Communications (ICC)*, June 2007.

[6] L. Gomes, C. Cazita, J. Almeida, V. Almeida, and W. Meira. Characterizing a spam traffic. In *Proceedings of IMC'04*, Oct. 2004.

[7] J. Goodman. IP addresses in email clients. In *Proceedings of First Conference on Email and Anti-Spam (CEAS)*, July 2004.

[8] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):25–33, Feb. 2007.

[9] B. Guenter. Spam archive. `http://untroubled.org/spam/`.

[10] M. Handley and A. Greenhalgh. Steps towards a DoS-resistant Internet architecture. In *Proceedings of ACM SIGCOMM FDNA Workshop*, Aug. 2004.

[11] T. Hansen, D. Croker, and P. Hallam-Baker. DomainKeys Identified Mail (DKIM) Service Overview. RFC 5585, June 2009.

[12] IANA. Top-level domain (tld) list. `http://data.iana.org/TLD/`.

[13] J. Klensin. Simple Mail Transfer Protocol. RFC 5321, Oct. 2008.

[14] J. Lyon and M. Wong. Sender ID: Authenticating e-mail. RFC 4406, Apr. 2006.

[15] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proc. ACM SIGCOMM*, Sept. 2006.

[16] P. Resnick. Internet message format. RFC 5322, Oct. 2008.

[17] SpamAssassin. The Apache SpamAssassin project. http://spamassassin.apache.org/.

[18] Spamhaus. The policy block list. `http://www.spamhaus.org/pbl/`.

[19] Spamlink. DNS and RHS blackhost lists. `http://spamlinks.net/filter-dnsbl-lists.htm`.

[20] A. Spiers. CPAN Mail::Field::Received. `http://search.cpan.org/~aspiers/Mail-Field-Received-0.24/Received.pm`.

[21] M. Wong and W. Schlitt. Sender policy framework (spf): Authorizing use of domains in e-mail, version 1. RFC 4408, Apr. 2006.

[22] Y. Xie, F. Xu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: Signatures and characteristics. In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[23] Zoneedit. Free DNS service. `http://www.zoneedit.com/`.