MDMap: Assisting Users in Identifying Phishing Emails

Patrick Dwyer
Florida State University
dwyer@cs.fsu.edu

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

ABSTRACT

Email-based online phishing is one of the key security threats that greatly deteriorate the trustworthiness of the Internet. Although many spam filters have been developed and deployed, a non-negligible number of phishing emails still sneak into users' inboxes each day. Phishing emails often contain suspicious information that separate them from the legitimate ones; however, average non-expert email users are not acquainted with the details of the Internet email system so as to identify the suspicious information in phishing emails. In this paper we develop a simple yet effective system named MDMap to assist email users in identifying phishing emails. MDMap reveals suspicious information in phishing emails in an intuitive and sensible manner. In particular, in addition to other features, MDMap provides a geographical map showing the message delivery path of an email, which helps to caution the user if the email has been originated from or traversed a suspicious region. In this paper we present the design and development of MDMap and perform a preliminary experiment to illustrate the usefulness of MDMap using real-world phishing emails.

1. INTRODUCTION

Email-based online phishing is one of the key security threats on the Internet, which greatly deteriorate the trustworthiness of the Internet as a global communications platform. In recent years, many spam filters have been developed; however, a non-negligible number of phishing emails still sneak into users' inboxes each day. Moreover, phishing attacks have increased in both numbers and sophistication [7, 15, 17]. For example, a recent report from RSA [17] showed that the number of phishing attacks increased 21% in January 2010 compared to that in December 2009.

Despite the advances in the sophistication of phishing attacks, phishing emails often contain suspicious information that separate them from the legitimate ones. However, the average non-expert email users are not acquainted with the details of the Internet email system. As a consequence, distinguishing phishing emails from legitimate ones presents a great challenge for the average, non-expert email users, who are often the target of online phishing scams and who often fall victim to these attacks. After a phishing email successfully penetrates a spam filter, the recipient of the message is on his or her own to judge the nature of the message.

It is clear that it is impossible for all email users to become an expert on the Internet email system. There is an urgent need to develop more *intuitive and sensible* methods to assist email users in identifying phishing emails, *without* requiring them to completely understand the details of the Internet email system. Towards this goal, we design and develop a simple yet effective system named MDMap to assist email users in identifying phishing emails by revealing suspicious information in a phishing email in a more sensible manner. Amongst other features, MDMap provides a geographical map showing the message delivery path of an email, based on the `Received:` header fields carried in the email [12].

Given that a phishing email is often originated from or traverses suspicious regions with respect to the main theme of the message, MDMap helps caution the recipient in responding to such a message. For example, it looks suspicious even for average email users if a message concerning accounts at the Bank of America was originated from or traversed a foreign country. Note that phishers may insert faked `Received:` header fields into a phishing email; however this behavior will not affect the effectiveness of MDMap because the complete message delivery path instead of only the (claimed) first hop is shown in the geographical map. Indeed, faked `Received:` header fields often cause inconsistency in the message delivery path, working against the interest of the phisher when the complete path is investigated (instead of only the first hop).

In this paper we present the design and development of MDMap and perform a preliminary experiment to illustrate the usefulness of MDMap using real-world phishing emails. A prototype of MDMap has been implemented as a standalone Java program using the MaxMind GeoLite City API (for obtaining the geographical location of an IP address or domain name) [13] and the Google Maps API [10]. Other packages can also be used including the Bing and Yahoo! Maps APIs [14, 20]. Although MDMap is presented as a standalone program in this paper, we envision that it can be incorporated into web-based email systems and provided as a service feature to their users. As an example, should MDMap have been incorporated into Yahoo! Mail, when a user opens a message, an MDMap can be shown along with the content of the message to assist the user in judging the nature of the email. Similarly, MDMap can be adapted as an application for PDA devices such as smart phones [2].

The remainder of the paper is organized as follows. In Section 2 we provide the necessary background on the Internet email delivery and the message format. In Section 3 we present the design and development of MDMap. We per-

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

form a preliminary experiment to illustrate the usefulness of MDMap using real-world phishing emails in Section 4. We briefly discuss the related work in Section 5. We conclude the paper and discuss future work in Section 6.

2. BACKGROUND

In this section we provide some background on the Internet email delivery and the message format that are most relevant to our work (see [12, 16] for a complete treatment).

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) [12]. 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.

A message header contains a number of message header fields. Four header fields Received:, Return-Path:, From:, and Reply-To: are of particular interest to the development of MDMap. These four header fields provide information of the email sender. Before we describe the four header fields, we emphasize that, due to the security weakness in the design of both SMTP and the Internet message format [16], almost all the header fields of a message can be faked, including the four header fields. However, we note that this forgery behavior will not prevent us from identifying suspicious information in a phishing email. First, when examined collectively, faked header fields often present conflicting or suspicious information, which helps identify phishing emails. Second, for certain category of phishing emails that depend on recipients to directly reply to the email senders (instead of re-directing recipients to a phishing website), the header fields of From: and Reply-To: cannot be both faked. Recent phishing scams targeting U.S. colleges and universities belong to this category [17], which asked users to confirm their email account information.

We first describe the Received: field. As an email traverses an MTA, a Received: header field is prepended to the message header. A Received: field contains two required clauses from and by, and a few optional clauses including with and id. 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.

Using the common convention [19], 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. Note that 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 following example Received: header field, the from-from host name is almostcosmic.com; the from-domain host name is n226-h110.gw-net.metromax.ru, and from-address is 83.234.226.110. The by-domain is smtpin.cs.fsu.edu.

```
Received: from almostcosmic.com
          (n226-h110.gw-net.metromax.ru [83.234.226.110])
          by smtpin.cs.fsu.edu with SMTP id o24DvD3r010823
```

Now let us discuss the other three fields: Return-Path:, From:, and Reply-To:. In essence, they all contain email addresses. Return-Path: contains the envelope MAIL FROM address. The From: and Reply-To: header fields specify where a reply message should be sent. If both of them are present in a message, the priority should be given to the Reply-To: email address. That is, the reply message should be sent to the Reply-To: email address if it is present.

3. MDMAP DESIGN AND IMPLEMENTATION

The goal of MDMap is to assist (average) email users in identifying phishing emails by revealing suspicious information in a phishing email in an intuitive and sensible manner. The main idea of MDMap is to visually expose the information related to the email senders. In particular, given a (phishing) email, MDMap provides a geographical map to show the location information related the email senders.

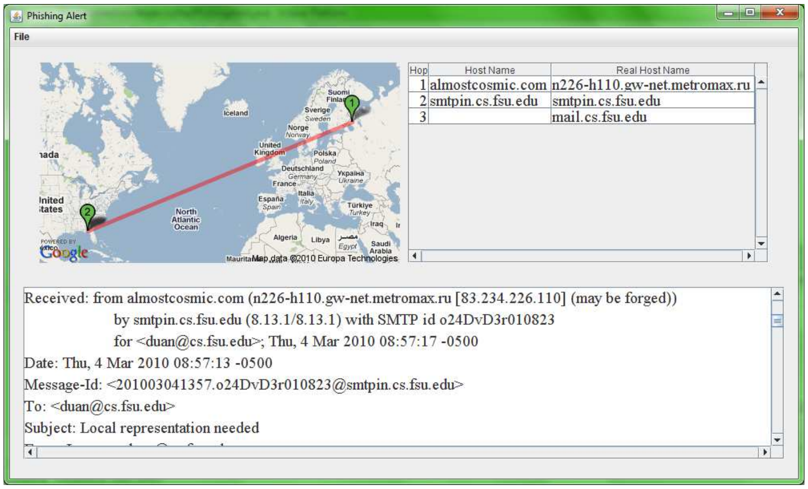
In MDMap, the sender location information is derived from four header fields Received:, Return-Path, From:, and Reply-To:; and URL links carried in the message body. As we have discussed early, the header fields can be faked. However, this will not prevent us from looking for conflicting or suspicious information. In the following we discuss how we derive the location information from the header fields and the URL links. In essence, we extract the domain name or IP address information of each interested field and then map it into the geographical location. We then obtain (and display) a map using the Google Maps API [10].

3.1 Message Delivery Path

The main component of MDMap is a map showing the message delivery path obtained from the Received: header fields. The Received: header fields may contain forgery information; however, the information inserted by the mail servers in the recipient's network domain should be correct. In particular, the from-domain and from-address of the first external MTA server (i.e., the MTA server that delivers the message into the recipient's network domain) *must* be correct. We first describe how we process the Received: header fields in order to obtain the message delivery path.

Removing localhost MTA servers. The from-domain and from-address of a Received: field may be localhost and 127.0.0.1, respectively. In this case, the sending MTA server and the receiving MTA server are the same machine. This is normally caused by the re-delivery of an email. An example is the email forwarding mechanism that uses the .forward file on Unix [4]. Given that this type of Received: header fields do not include a new MTA server, we exclude such header fields in deriving the message delivery path if the from clause refers to a local host.

Removing MTA servers with private IP addresses. The from-address of a Received: header field may be a private IP address [11, 8]. For this kind of IP addresses we cannot determine the geographical location of the MTA servers. We exclude the MTA servers with private IP addresses in deriving the message delivery path. We note that this will not eliminate any conflicting or suspicious information in the message delivery path. An MTA server will accept messages from another MTA server with a private IP address only if they belong to the same network domain. Therefore, excluding MTA servers with private IP addresses



Hop	Host Name	Real Host Name
1	almostcosmic.com	n226-h110.gw-net.metromax.ru
2	smtpin.cs.fsu.edu	smtpin.cs.fsu.edu
3		mail.cs.fsu.edu

```
Received: from almostcosmic.com (n226-h110.gw-net.metromax.ru [83.234.226.110] (may be forged))
                by smtpin.cs.fsu.edu (8.13.1/8.13.1) with SMTP id o24DvD3r010823
                for <duan@cs.fsu.edu>; Thu, 4 Mar 2010 08:57:17 -0500
Date: Thu, 4 Mar 2010 08:57:13 -0500
Message-Id: <201003041357.o24DvD3r010823@smtpin.cs.fsu.edu>
To: <duan@cs.fsu.edu>
Subject: Local representation needed
```

Figure 1: A snapshot of MDMap.

will not remove all MTA servers in a network domain along the message delivery path.

After we exclude the `Received:` header fields whose *from-addresses* are either localhost IP addresses or private IP addresses, we extract the sequence of MTA servers from the remaining `Received:` header fields in the following manner and refer to this sequence as the message delivery path. Let hr_i for $i = 1, 2, \dots, k$ denote the sequence of `Received:` header fields (where hr_k is the one inserted by the last MTA server). We form the message delivery path by sequentially extracting the `from-address` from the header fields hr_i for $i = 1, 2, \dots, k-1$, and the `by-domain` from the header field hr_k. Based on the IP addresses of the MTA servers along the message delivery path (for the last MTA server, we use the domain name), we obtain the geographical locations of the MTA servers in terms of longitude and latitude using the MaxMind GeoLite City API [13]. This longitude and latitude information is then fed to the Google Maps API to obtain a map showing the message delivery path.

In addition to the message delivery map, we also directly show the sequence of the MTA servers along the message delivery path. We show both the `from-from` domain name and the `from-domain` domain name, so that the user can observe the potential discrepancy between the two. Figure 1 shows a snapshot of the MDMap system after a phishing email has been opened. This message traversed three MTA servers (the map has three nodes; the last node is covered by the second one as they are located in the same city). Given that the message was originated from a suspicious region (Russia), the user can safely infer that the nature of the message may be malicious, based on the message delivery map of MDMap and the context of the message.

3.2 Other Sender Location Information

MDMap also relies on other parts of an email to expose potentially conflicting or suspicious sender location information. Three other header fields `Return-Path`, `From:`, and `Reply-To` are used (if they are present in a message). We extract the domain names from the email addresses in these fields, and obtain the geographical location of the email address domains as we have done for the message delivery path. Each email address is shown as a node in the map based on the longitude and latitude information of the corresponding domain name of the email address.

Similarly, for each URL link carried in the message body, we obtain the domain name of the corresponding web server, and then the geographical location in terms of longitude and latitude as we have discussed above. A node is shown in the map for each unique URL link based on the geographical location of the corresponding web server. The location information obtained from the three header fields and the URL link also helps caution email users on if and how they should respond to an email. For example, if an email claims coming from Citibank but the `Reply-To` or the URL link is located in a suspicious region, the user can safely infer that the message could be a phishing email.

4. PERFORMANCE EVALUATION

In this section we perform a preliminary experiment to illustrate the usefulness of MDMap using real-world phishing emails. We randomly selected a set of 100 phishing (or spam) emails that one of the co-authors received in a time period of about four months (from mid 11/2009 to mid 03/2010). The emails were randomly selected in the sense that we did not examine the content of an email before it was being selected from the set of all spam (and phishing) emails that we received in the period. (It is worth noting that "hard" spam messages have been blocked by the departmental spam filter and therefore are not in the candidate spam set.) After the 100 emails have been selected, we manually checked the messages to confirm that they are phishing (or spam) messages.

In this experiment, we classify an email as being suspicious based on three simple heuristics that we will discuss in the following; all three heuristics operate at the *country* level. (H1) An MTA server is located in a suspicious country along the message delivery path based on the context of the message. For example, we flagged a message to be suspicious because it was originated from or traversed an MTA server located in Turkey while the content of the message is about Bank of America. (H2) The domain name of the email address in `Return-Path:`, `From:`, or `Reply-To:` is located in a suspicious country based on the context of the message. For example, we flagged a message because one of

Table 1: Number of phishing emails flagged.

H1	H2	H3	Total Flagged	Total
70	47	19	82	100

the domain names is located in Australia while the message content is about updating email account information. It is also common that when a message is flagged by this heuristic, the three header fields often contain domain names in different countries. (H3) The domain name of an URL link in the message body is located in a suspicious country. For example, we flagged a message to be suspicious because the web server is located in China while the message content is about updating email account information.

Table 1 shows the number of phishing emails flagged by each heuristics and the total number of emails being flagged by all three heuristics. Note that the set of phishing emails flagged by each heuristic may overlap. From the table we can see that out of the 100 phishing (or spam) messages, MDMap helps to flag 82 messages as being suspicious. Among the three heuristics, H1 (message delivery path) flagged the most suspicious messages (70 emails). Results of this preliminary experiment confirm that MDMap is a useful tool in assisting users in identifying phishing emails.

5. RELATED WORK

A large number of spam filters have been developed in recent years [1, 9, 18]. However, they normally target the filtering of general spam emails instead of phishing emails. A few filters are designed specifically for phishing emails. In [3] the authors developed a scheme to filter phishing emails based on the structural properties of phishing emails. A scheme was developed in [6] to filter phishing emails based on the features of phishing emails including IP-based URLs and the age of domain names. However, none of them can achieve 100% filtering rate of spam (phishing) emails. MDMap can work in conjunction with these spam filters, after a phishing email penetrates all spam filters.

Many web browser-based toolbars have been developed (see [21] and references therein). However, as reported in [21], existing anti-phishing toolbars have poor performance in terms of both false positive and false negative rates. Moreover, they only target phishing scams involving websites as part of the attack vector. As we have discussed, many recently popular phishing attacks do not use websites as part of the attack vector. Instead, they ask the recipients to directly reply to the email senders.

EmailTrackerPro [5] tries to determine the possible true originating machine or domain of an email by detecting and eliminating potentially forged `Received:` header fields. A map is used to show the location of the identified originating machine (and the `traceroute` information to the location). EmailTrackerPro and MDMap have different design objectives. While EmailTrackerPro focuses on identifying the originating machine and reporting email abuse, MDMap targets revealing potentially suspicious sender information to assist users in identifying phishing emails. In addition, as tested, it is easy to mislead EmailTrackerPro in identifying the true originating machine of an email by including additional `Received:` header fields (tests were done on the current EmailTrackerPro version v9.0f (Build 3002)).

6. CONCLUSION AND FUTURE WORK

In this paper we developed a simple yet effective system named MDMap to assist email users in identifying phishing emails. In addition we also performed a preliminary experiment to illustrate the usefulness of MDMap using real-world phishing emails. As our future work we will fine tune the design of MDMap; we will also perform thorough experiments to study the performance of MDMap using a larger and more diverse set of phishing emails (instead of phishing emails received by a single user). We also plan to develop an MDMap application for smart phones.

7. REFERENCES

[1] S. Ahmed and F. Mithun. Word stemming to enhance spam filtering. In *Proceedings of CEAS*, July 2004.

[2] Android. `http://developer.android.com/sdk/index.html/`.

[3] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. Phishing email detection based on structural properties. In *New York State Cyber Security Conference*, 2006.

[4] B. Costales. *Sendmail*. O'Reilly, 2002.

[5] EmailTrackerPro. `http://www.emailtrackerpro.com/`.

[6] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *Proceedings of WWW*, Alberta, Canada, May 2007.

[7] Gartner. Number of phishing attacks on U.S. consumers increased 40 percent in 2008, Apr. 2009. `http://www.gartner.com/it/page.jsp?id=936913`.

[8] J. Goodman. IP addresses in email clients. In *Proceedings of CEAS*, July 2004.

[9] 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.

[10] Google. Google static maps api. `http://code.google.com/apis/maps/documentation/staticmaps/`.

[11] IANA. Special-use ipv4 addresses. RFC 3330, Sept. 2002.

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

[13] MaxMind. MaxMind geolite city. `http://www.maxmind.com/app/geolitecity`.

[14] Microsoft. Bing maps platform. `http://www.microsoft.com/maps/developers/`.

[15] E. Mills. Twitter resets passwords after phishing attack, Feb. 2010. `http://news.cnet.com/8301-27080_3-10445898-245.html`.

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

[17] RSA. RSA online fraud report, Feb. 2010.

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

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

[20] Yahoo! Yahoo! maps web services. `http://developer.yahoo.com/maps/`.

[21] Y. Zhang, S. Egelman, L. F. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In *Proceedings of NDSS*, 2007.