Exploiting the Trust Hierarchy among Email Systems

Pablo Ximenes[1], André dos Santos.[1,2]

[1]Information Security Research Team (INSERT) – University of Puerto Rico at Mayagüez (UPRM), USA

[2]Communications Networks and Information Security Laboratory (LARCES) – State University of Ceará (UECE), Brazil

pablo.ximenes@upr.edu, andre@dossantos.org

Abstract. *This paper presents a critique of the current status of the trust hierarchy found among SMTP based email systems. We evaluate current trends and present real evidence that the prevalence of ad-hoc initiatives for trust classification is a potential risk in itself. In that sense, we describe a vulnerability found in Google's free email service (Gmail) that allows an attacker to exploit the current trust hierarchy that exists between email providers in order to assemble powerful spam/phishing attacks. We demonstrate this vulnerability by crafting a proof of concept attack software that is able to send whitelisted open relayed unlimited spam and phishing messages through Google's email servers, thus giving concrete evidence of the presented threat.*

1. Introduction

Email fraud (e.g. spam, phishing) is an increasingly striking problem. Not only it amounts for extreme annoyance to users and system administrators, it also represents serious economic and safety concerns. According to recent statistics [CTC2007] 95% of all electronic mail messages throughout the internet are spam with approximately 8% of these messages being associated to phishing.

One of the early measures that were adopted in the fight against unwanted email messages was the better handling of the way email messages were relayed within the internet SMTP based email infrastructure. Network administrators started blocking access to their otherwise public accessible SMTP relay servers and IP address blacklists were assembled [STR2005]. These protocol level protections were complemented with techniques that would try to classify messages regarding their spam status after delivery based on heuristics that analyze the full contents of the messages [GOO2007]. Protection against spam has since then been divided into two categories: protocol based techniques that try to block messages at the border level before delivery is complete, and techniques that try to classify messages based on an in-depth analysis of their full contents after delivery [COO2006]. Though border level approaches do not solve all the issues related to spam, these measures significantly diminish the computational cost of spam filtering by blocking obvious sources of spam (e.g. through blacklists). Without border level approaches, every single email message would be required to pass through an in-depth scan to assess its spam and phishing status in order to properly filter unwanted junk. This way, the reliance on border level approaches, especially in the form of blacklists and whitelists has been independently adopted by most email providers as a way of cutting down on spam filtering performance bottlenecks. Nevertheless, this ad-hoc trend has created a trust hierarchy among email providers that obeys no specific pattern and might be based on nothing more than mere convenience. This trend might constitute a dormant danger to the very email system it tries to protect. For instance, by compromising just one of the trusted nodes in the hierarchy, and attacker could be able to efficiently subvert the entire security premise thus forcing all other email providers into either dropping the blacklist/whitelist based security premises (and potentially overload their inner spam filters with an unmanageable number messages) or having to bear with attackers until the point of failure is fixed. Given the heterogeneous and widespread nature of the SMTP based email infrastructure that makes it naturally prone to security vulnerabilities and the absurd and increasing number of spam messages, neither of the solutions seems to be acceptable.

This paper intends to contribute to the awareness about the problem of the existing trust hierarchy found among email systems by discussing the security premise and showing the factuality of a concrete threat. As evidence this paper presents a vulnerability in Gmail (Google's free email service) that allows an attacker to abuse Google's SMTP servers in order to send an unlimited number of spam/spoofed email messages. The trust level Google's SMTP servers have within the Internet SMTP based email system trust hierarchy generally grants its messages special treatment, what contributes to make an attack based on this vulnerability even more efficient. This way, attacks that exploit this vulnerability are likely to bypass spam filters, effectively subverting the trust hierarchy among email providers.

The vulnerability described above is related to the possibility of abusing Gmail's message forwarding functionality. This is possible because no restriction or verification is imposed during the setup process of this option. We were able to confirm that this vulnerability is indeed exploitable by assembling a proof of concept (PoC) attack that allowed us to use one single Gmail account to send bulk messages to more than 4000 different email targets (surpassing Gmail's 500 messages limit for bulk messages). Although we have limited the number of messages to a little over 4000, no countermeasures took place that would have prevented us from sending more messages, and for that matter sending an unlimited number of messages. Additionally, this vulnerability allows attackers to forge any of the message body fields (including sender's identity and time/date information) which are not inspected nor modified by Google's servers in any way. Furthermore, we were able to use this vulnerability to forward messages that originally were classified as spam directly to a victim's inbox effectively bypassing filters. Since attack messages are carried by Google's own SMTP servers, the blacklist/whitelist based trust hierarchy that exists between Google's and other Third Parties' email servers is compromised, effectively converting Google's servers into the perfect spam/phishing aid.

This document is organized as follows. Section 2 presents a background discussion regarding current trends in spam filtering. Section 3 specifies the main problem we are discussing. Section 4 details the vulnerability that has been found in Gmail that serves as concrete evidence to our claims. Section 5 shows our experimental findings in assessing the presented vulnerability. Section 6 discusses potential solutions to the presented vulnerability and shows how this vulnerability might an indication of a larger problem. Finally, we draw some conclusions in section 6.

2. Background

In this section we discuss the current state of art regarding protection against unwanted messages through an overview of some of the major techniques. We arrange these techniques in two basic groups by coining the terms 'border level filters' and 'deep inspection filters.' The first term describes the group of techniques that take place at message delivery time, where the second describes techniques that take place after the message is delivered. Although our work concentrates on the former, the interaction between those two groups is paramount to our argument.

2.1 Border Level Filtering

We classify as a 'border level filter' any technique that has low computational cost as to be used during message delivery negotiation. The main purpose of such filters should be identifying all obvious spam messages via protocol level analysis. Border level filters should block or flag messages before they enter the target system as to alleviate and/or complement the work of subsequent filters.

2.1.1 Source authentication

SMTP makes it trivial to spoof (i.e. forge fraudulently) most data fields in an email message, including sender identity and source domain. Spam messages usually take advantage of this fact to conceal the true identities of perpetrators. Verifying if sender identity information was spoofed in an email message is therefore one important step in blocking unwanted messages.

Among the techniques that try to solve the issue of correct sender identification, only the Sender Policy Framework (SPF) [WON2006] can be considered a border level filter. SPF makes use of the DNS system in order to publish DNS records containing lists of IP addresses of the servers that are authorized to send email in behalf of a particular domain. Through a simple DNS look-up, Mail Transfer Agent (MTA) servers are able to determine whether a message was delivery by an authorized server. This technique generalizes the sender verification to the domain level. Although the SPF control can filter out many spoofed messages, it is still possible to spoof senders from the same domain. For instance, if the IP address 10.1.1.1 is authorized to deliver mail for the domain 'example.com', there will be no difference in the SPF check if the mail is said to be from legitimate@example.com or attacker@example.com.

2.1.2 Blacklists

Also known as real-time blackhole lists (RBL) and domain name system black lists (DNSBL) [JUN2004], this technique is structured in the form of individual databases of IP addresses of known and/or suspected spam offenders. Through simple DNS lookups, MTA servers are able to assess if an IP address is listed in one of the various databases.

Blacklists databases are maintained in an ad-hoc fashion by different corporations that have no common policy on how to list or de-list IP addresses. Lists might be focused on including RFC non-compliant servers, open SMTP relays, open proxies (e.g. http proxies, SOCKS), IP addresses that actually have been found as source in spam messages, and even temporary IP addresses usually assigned to broad-band and dial-up internet users (as those addresses are typically not a source of legitimate messages). Additionally, some list managers might have a more aggressive approach when listing IP addresses, possibly increasing the likelihood of false positives [COO2006].

2.1.3 Whitelists

Whitelists are the lists of domains that are given 'carte blanche' to bypass spam filters. Whitelists are usually managed by the local domain (as opposed to a shared database as it happen for blacklists). Entries in whitelists are usually motivated either by performance reasons, where high volume 'trusted' email servers receive special treatment as to cut down on performance bottlenecks caused by spam filters, or by financial incentive where some companies literally buy their way around spam filters [HAN2006][DYS2006]. A variation of the whitelist concept is the challenge response system [PFL2005], where a verification challenge is performed to the first of a series of messages. After the challenge comes out positive proving that the message comes from a legitimate source (i.e. not a spammer), subsequent messages that exhibits the same characteristics as the first one will bypass filters.

Blacklists together with whitelists are the main border level filters used by most email systems in the Internet.

2.2 Deep Inspection Filtering

Deep inspection filters comprise the techniques that analyze the inner contents of the message body in order to assess its status as a spam message. Since they are usually more computationally intensive, deep inspection filters work after messages are delivered and queued. Deep inspection filters' main concern should be accuracy in order to avoid false positives.

The main deep inspection filter technique used by email system throughout the Internet is Bayesian Filtering [PAN1998][SAH1998]. Bayesian (or statistical) filters analyze words inside the message body and calculate a probability of the message being spam. This is done by accounting both words that contribute for the likelihood of being spam as well as words that would indicate the message is not spam. By a composition of the contribution of both types of words a general grade of spam classification is assigned. One of the main drawbacks of deep inspection filters is the computational impact it imposes and the fact that the contents of a message might be especially obfuscated in order to avoid filters. Spammers might use images, garbled text, word distributions containing legitimate text, all in attempt to conceal the actual contents of the message from filters. Nevertheless, this constitutes a cat and mouse chase between spammers and security researchers, where deep inspection filters unavoidably become increasingly computational intensive.

3. Problem Definition

The evolving complexity of deep inspection filters has made room for an increasing dependence on border level filters. If every email message were to be scanned by deep inspection filters, email systems would eventually overload their capacities and potentially become easy targets for denial of service attacks (DoS). This way, border filters are of utmost importance. Since the main type of border level filter currently used is based on the blacklist/whitelist paradigm, this technique ultimately creates a hierarchy of trust among email systems, where email coming from good sources is able to reach inboxes bypassing most of the filtering process, and messages from bad or suspicious sources are queued to be verified by deep inspection filters.

The problem with this trust hierarchy is that the lists that are the basis for the trust relationship have no standardization and are almost completely organized in an ad-hoc fashion. For instance, the mere fact that a server was assigned an IP address that once belonged to a dial-up internet user might completely block this server from being able to act as a legitimate email server. Nevertheless, this is not the biggest concern. Although the DoS risk created by blacklists that blocks legitimate parties as the previous example is certainly a problem, the risk of completely bypassing spam filters of systems that rely on blacklists/whitelists is an actual threat. By compromising just one trusted node within the hierarchy, the entire security premise falls apart. For instance, Google's email servers send email in behalf of millions of users worldwide. This fact alone makes many email providers fearful of directing messages coming from Google's servers to deep inspection filters. Additionally, Google has made a success campaign advertising its spam filters [MIN2006][JAC2007], creating a general sense of trust with regards to messages that come from Google's servers. This trust ultimately became a dependency, as email systems that usually do not filter messages from Google's servers would have to deal with a potentially unbearable load if they were to start applying deep inspection filters to all Google originated messages.

As we will see further in this paper, not even Google is immune to vulnerabilities and the reliance on the current blacklist/whitelist based trust hierarchy might cost many email systems a lot of trouble.

4. Proving the point

The problem with trust hierarchy is an old and already well know problem [LEV1996] [BLZ1999]. However, there is always some that believe that they can minimize known security problems and implement ad-hoc solutions that are based either on security by obscurity or on the ignorance of the complexity of the problem. As already discussed, deep inspection filtering would be too costly to be performed in every email message and therefore whitelists/blacklists are widely in use. Whitelists are a clear example of trust hierarchy and in the experiments performed we were able to demonstrate that abusing a widely used email service (Gmail) that has been whitelisted by many other e-mail services we can:

a) Validate any e-mail as if it were being sent from Gmail and thus receiving all benefits of whitelists;

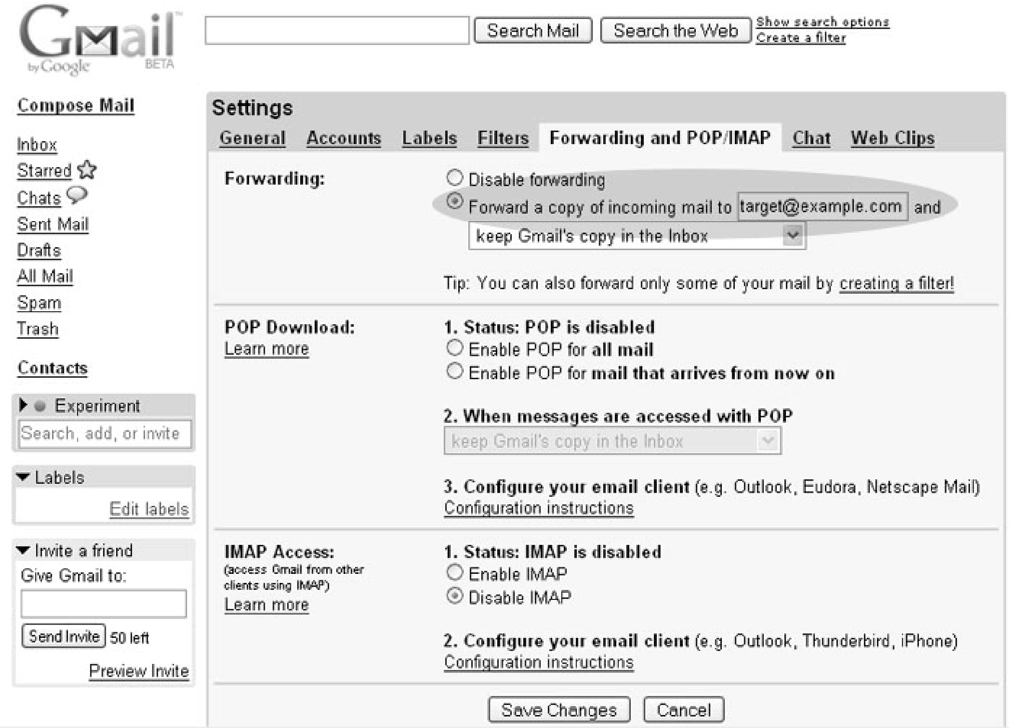
b) Use Gmail as an open relay, thus being able to spoof email fields, while still keeping the whitelist benefits (or at least not being blacklisted);

c) Have unlimited access to a massive distributed SMTP relay infrastructure by effectively circumventing Gmail's policy limitation on the number of messages one account is allowed to send.

The vulnerability we present is related to the possibility of abusing the message forwarding option in Gmail accounts which is facilitated by Gmail's reliance on user input to influence its spam filters. In order to proper describe the problem, we will first review parts of Gmail's structure that are related to the issue to later expose the complete vulnerability.

4.1. Message Forwarding

Gmail's message forwarding option enables a regular Gmail user to have her email messages forwarded to a different email account. It is important to notice that the receiving account can belong to any email system and not necessarily Gmail's. Gmail's email forwarding can be setup up either by activating the global forwarding option or by creating a filtering rule that forwards all messages to a different email address. Figure 1 shows Gmail's configuration interface for its email forwarding option.



Settings

Forwarding:
○ Disable forwarding
● Forward a copy of incoming mail to target@example.com and keep Gmail's copy in the Inbox

Tip: You can also forward only some of your mail by creating a filter!

POP Download:
Learn more

1. Status: POP is disabled
○ Enable POP for all mail
○ Enable POP for mail that arrives from now on

2. When messages are accessed with POP
keep Gmail's copy in the Inbox

3. Configure your email client (e.g. Outlook, Eudora, Netscape Mail)
Configuration instructions

IMAP Access:
(access Gmail from other clients using IMAP)
Learn more

1. Status: IMAP is disabled
○ Enable IMAP
● Disable IMAP

2. Configure your email client (e.g. Outlook, Thunderbird, iPhone)
Configuration instructions

Figure 1. Gmail's message forwarding feature

It is up to the Gmail user to make sure the email address used in the forwarding configuration actually belongs to her since Gmail will not ask for any proof of ownership. The only verification performed by Gmail concerns the format of the address that must comply with the general guidelines for email addresses.

After message forwarding is enabled and configured, the general behavior of this option is to forward messages that are not marked as spam to the email destination chosen by the user. The underlying technical setup that occurs when an e-mail is sent to a forward-enabled Gmail account happens as follows:

1) The message is delivered to one of Google's incoming Mail Transfer Agent (MTA) servers.
2) Google's incoming MTA server will route the message to the intended user account. If the message is addressed to an email account that does not belong to Gmail or to a company hosted by Google, a 500 error message will be displayed and the process is aborted. The abortion is done to avoid the use of Google's MTA servers as SMTP relays by unauthorized external parties.
3) After reaching the Gmail user account, the message is classified to be forwarded (i.e. matching filtering rule and/or not spam)
4) Extra headers are added to the message body to account for the forwarding process.
5) One of Google's outgoing MTA servers establishes an SMTP connection to the forwarding target's input MTA server.
6) Google's outgoing MTA server will set the return path ('Mail From:' SMTP command) to an address that belongs to Gmail's domain. This address is structured in the form 'GMAILUSER+caf_=TARGETUSER=TARGETDOMAIN@gmail.com', where GMAILUSER is the Gmail account's user name, TARGETUSER is the target's email domain user name, and TARGETDOMAIN is the domain name (e.g. example.com) of the forwarding target. Considering the Gmail account attacker@gmail.com and the forwarding target target@example.com, the address would be 'attacker+caf_=target=example.com@gmail.com'.
7) Google's outgoing MTA server will issue a "RCPT TO:" command containing the target's email address as argument.
8) Google's outgoing MTA server will issue a "DATA" command after which it will send the original message body almost intact except for the extra headers added in step 3.
9) The SMTP connection between Google's outgoing MTA server and the target's incoming MTA server is terminated and the message is queued for delivery on the target's system.

4.2. Spam Message Whitelisting

An interesting feature found in Gmail is the ability users have to fine tune their spam filtering settings by whitelisting messages that were classified as spam. The whitelisting is done by displaying the message that was classified as spam and clicking the 'Not Spam' button. After that, the message will be moved to the inbox. In addition to the original message, Gmail will whitelist any message that follows the same patterns by which the whitelisted message was masked as spam. The consequence of the whitelisting is that if the same message is to be sent again, using the same server, it will no longer be classified as spam. The ability to whitelist (mark as 'not spam') and to blacklist ('mark as spam') messages for individual accounts is one of the strongest supporting mechanisms of Gmail's spam filters, since they rely on a collaborative approach that samples the input from all users on blacklisting and whitelisting to correctly identify spam messages [GMA2008].

4.3. The Vulnerability

The vulnerability is exploitable due to three features that in an isolated manner can be thought as inoffensive. That proves another well known point that the combination of otherwise inoffensive features can result in an exploitable feature. Systems, particularly complex ones, have to be well analyzed in their totality so security vulnerabilities can be identified. The three features are:

a) No proof of ownership of the email address that is used as target for the message forwarding option in Gmail accounts is required. A Gmail user can easily setup her account to forward messages to any email address in the Internet, since no verification of ownership is done.

b) No limit is imposed on the number of times a Gmail user can change the email address used as target in the message forwarding configuration.

c) Gmail users can whitelist a spam message.

These functionalities just mentioned can be abused by a malicious user and together can be used to assemble a powerful spam attack. The attack can be performed as to be used by spammers in the following fashion:

1) Whitelist the attack message. Since the attack message will probably originate from a blacklisted IP address and might contain other indicators that will make Gmail flag it as spam, the message needs to first be whitelisted (i.e. marked as 'not spam') before it can be forwarded.
2) Change the email address destination in the message forwarding option.
3) Deliver the attack message addressed to the compromised Gmail account using one of Google's MTA servers.
4) Repeat steps 2 and 3 for every email address in the list of addresses to be spammed effectively sending the attack message to all addresses.

Gmail will deliver the message to the attack targets without modifying any of the message body fields, including sender's identity. This is done so because the attack message is originally disguised as a message that is legitimately destined to the compromised Gmail account. Since these messages are normally expected to come from a different domain and the forwarding process inevitably keeps forwarded messages intact as to be faithful to the original delivery, the attack message will be forwarded to each of the victims preserving the same contents it was originally sent, including spoofed sender's identity. Since attack messages can be performed at the attacker's will and can be forwarded any number of times, this vulnerability is both a spam and a phishing threat concern.

5. Proof of Concept Experiments

To seek proof that the vulnerability described here is indeed exploitable, we have designed a set of experiments that finally led us to the observation that Gmail can be effectively abused by a malicious user almost in the same fashion open relay SMTP servers can be abused.

5.1. Assessing Gmail’s Protections against Bulk Messages

Initially, we decided to assess the workings of Gmail's protections against bulk messages related abuse. The first thing we decided to evaluate was the limit of messages Gmail actually would allow one to send in one day using its system the regular way. To test this feature, we have sent messages via the Web Interface and via the authenticated SMTP interface until we have reached a point where Gmail started blocking out attempts to send more messages. The experiment consisted of sending the same message several times to different email addresses via SMTP and Web. Gmail has effectively locked our test accounts and prevented us from sending any more messages using the regular interfaces (i.e. SMTP, Web) as soon as we reached the 500 limit. In a related experiment we tried to use the SMTP interface to send a message with spoofed sender information. Even though the message was properly delivered, Gmail had rewritten the sender identity information to reflect that of the account’s owner, thus effectively preventing us from spoofing sender identity using Gmail’s regular interfaces.

5.2. Assessing Gmail’s behavior against the described attack

The first test we performed to evaluate Gmail’s reaction to the described attack was to send isolated messages with spoofed (i.e. fraudulent) fields. Particularly we have forged sender identity information and date/time fields. This experiment revealed that indeed Gmail had no protections in place to prevent defrauding the SMTP protocol in this manner.

After we already had been able to effectively send a spoofed message, we needed confirmation to if an attack could be crafted to send a large number of messages. This way, we assembled a program that reads a list of email addresses from a file and sends a forged message to the entire list using the described attack method.

By using a broadband Internet connection and one Gmail account, we were able to use our program to send a bulk message to more than 4,000 email targets (in a domain under our administration) in approximately 6 hours. No measures took place that would have prevented us from keeping sending more messages. Even though the average of 11 messages per minute seems low, it is important to notice that our demonstration exploited only one Gmail account. By exploiting a larger number of accounts the attack could improve its message rate significantly. For instance, by deploying this attack with 100 Gmail accounts simultaneously the message rate would exceed 1,000 messages per minute. This fact shows that by compromising a relatively small number of Gmail accounts it is possible to assemble an attack that would have results similar to those of a botnet based spam, but without the need for thousands of zombie computers.

5.3. Assessing the Trust Relationship between Gmail and other systems

The third part of our experiment was designed to assess the trust relationship between Gmail and other third parties' email providers. This way, we have opened test accounts in two of the other major free email providers: Yahoo and Hotmail. The experiment consisted of sending forged messages from blacklisted IP addresses (our computers) directly to Hotmail's and Yahoo's MTA servers directly and sending the same messages using our proof of concept (PoC) program (i.e. though Gmail's servers). We were able to confirm that indeed messages sent through Gmail’s infrastructure had special treatment by Hotmail and Yahoo.

```
Delivered-To: victim@example.com
Return-Path: <attacker+caf_=victim=example.com@gmail.com>
Received: from fk-out-0910.google.com (fk-out-0910.google.com [209.85.128.184])
        by mx.example.com with ESMTP id i5si4683640mue.2.2008.05.12.09.38.07;
        Mon, 12 May 2008 09:38:09 -0700 (PDT)
Received-SPF: pass (example.com: domain of attacker+caf_=victim=example.com@gmail.com designates
209.85.128.184 as permitted sender) client-ip=209.85.128.184;
Authentication-Results: mx.example.com; spf=pass (example.com: domain of
attacker+caf_=victim=example.com@gmail.com designates 209.85.128.184 as permitted sender)
Received: by fk-out-0910.google.com with SMTP id 18so1981820fks.2
        for <victim@example.com>; Mon, 12 May 2008 09:38:07 -0700 (PDT)
Received: by 10.82.173.1 with SMTP id v1mr831893bue.68.1210610286909;
        Mon, 12 May 2008 09:38:06 -0700 (PDT)
X-Forwarded-To: victim@example.com
X-Forwarded-For: atacker@gmail.com victim@example.com
Delivered-To: attacker@gmail.com
Received: by 10.82.185.4 with SMTP id i4cs82084buf;
        Mon, 12 May 2008 09:38:06 -0700 (PDT)
Received: by 10.100.8.10 with SMTP id 10mr8466460anh.54.1210610285273;
        Mon, 12 May 2008 09:38:05 -0700 (PDT)
Return-Path: <news@reuters.com>
Received: from attacker.domain.name.example.com (attacker.domain.name.example.com [10.20.30.40])
        by mx.google.com with ESMTP id d24si13903701and.24.2008.05.12.09.38.04;
        Mon, 12 May 2008 09:38:05 -0700 (PDT)
Received-SPF: neutral (google.com: 10.20.30.40 is neither permitted nor denied by domain of
news@reuters.com) client-ip=10.20.30.40;
Authentication-Results: mx.google.com; spf=neutral (google.com: 10.20.30.40 is neither permitted
nor denied by domain of news@reuters.com) smtp.mail=news@reuters.com
Message-Id: <48ee726d.1aae110a.13a8.eeffc9b7SMTPIN_ADDED@mx.google.com>
Date: Fri, 02 May 2008 11:30:46 -0400
From: "Reuters News Room" <news@reuters.com>
To: "Happy Victim" <victim@example.com>
Subject: bulk message

Hello,

Help us fighting SPAM! Take part in our experiment and understand where you can improve your
security.

Best Regards,

Experiment People
```

Figure 2. Attack message source code after final delivery

Some messages would not even reach the spam box when sent directly, while when relaying the same messages through Google's servers by using our program the messages were promptly delivered directly in the inbox. We conjecture that this behavior was facilitated by the fact that Gmail changes the return-path email address of forwarded messages to an address belonging to its own domain (as explained in section 4.1). This will result in an SPF successful check, since the message is delivered by one of Google’s server. This can be observed in figure 2 that shows the contents of one of the messages we have forwarded though Google’s system during our experiments.

The observed behavior for the SPF filter is actually the recommended behavior of RFC4408 [WON2006] for message forwarding which states that in order to avoid failed SPF checks, domains should replace the return-path of forwarded messages by an address belonging to the local domain. Additionally, RFC4408 also states that the “SPF authorization check is a check between border MTAs of different domains”, what makes Yahoo and Hotmail potentially ignore SPF headers with non-successful and/or failed results from previous SPF checks inside the message body.

5.4. Assessing the limit on message forwarding setup changes

The way Gmail handles its message forwarding configuration is the main part of the vulnerability we are presenting and the core part of a possible attack. This way, we decided to verify if there was any limit imposed by Gmail on the number of times we would be able to change the email address destination for Gmail's message forwarding option. The experiment consisted of a procedure that changes the message forwarding address destination to a randomly generated email address and a second procedure that verifies if the change is successful.

In our experiment we were able to successfully change the email address destination in Gmail's message forwarding option over 10,000 times in a single run. No countermeasures were taken by Google's systems that would have prevented us from keeping changing addresses.

6. Remarks

6.1. Gmail's Vulnerability Mitigation

The vulnerability in Gmail can be easily mitigated by Google by just requiring proof of ownership of the email target during the setup process of message forwarding.

It is important to notice that although one could propose a solution of limiting the number of changes on the forwarding address, this solution would be incomplete. Solutions that simply limit the number of times a user can change the email destination in Gmail's message forwarding configuration will only avoid the problem of sending messages in bulk while doing nothing to prevent attacks that spoof message body fields (e.g. phishing). Nevertheless, any solution that might require limiting the daily number of forwarded messages would be inappropriate since this approach would unavoidably impair the legitimate use of Gmail's message forwarding feature.

Third party email providers might mitigate the presented vulnerability before Google provides a solution by flagging messages from Google's Servers containing email forwarding message body fields (e.g. "X-Forwarded-For:", "X-Forwarded-To:"). These fields indicate that a message was delivered by Google's servers as a result of a forwarding procedure, and since any of such messages are potentially part of an attack, messages containing these fields should be always queued for further verification until Google provides a definite solution.

6.2. The Trust Hierarchy Problem

Although the mitigation of Gmail's vulnerability presented here is relatively straightforward, the underlying risk of abuse of the current trust hierarchy among email systems has no trivial solution.

Our reliance on a disorganized trust hierarchy system in the form of blacklists/whitelists as the main form of border level filtering makes our email systems particularly vulnerable to attacks similar to the one we have presented. Many email providers have come to realize that their trust on Google's system is not a reflex of security guarantees, but merely a away of conveniently reduce performance bottlenecks by having Gmail originated messages out of the picture. Nevertheless, we cannot simply dismiss border level filters entirely, since we need the performance improvements this type of filter brings but our reliance on blacklists/whitelists needs immediate redesign. This issue becomes even more alarming when we take into account the fact that even many aggressive blacklist operators will refuse to add Google's servers to their lists as we have been able to witness. After all, who would be crazy enough to block Gmail?

7. Conclusion

We have presented a critique of the current trust hierarchy that exists among SMTP based email systems by providing concrete evidence that the widespread ad-hoc based backlist/whitelist model should be replaced or at least improved. In this regard, we have presented a vulnerability that has been found in Google's Gmail that allows attackers to easily subvert the current trust hierarch of the Internet SMTP based email system to send malicious bulk messages more efficiently. This way, we have showed that the current trust model found among SMTP based email systems is vulnerable to the point of acting as a catalyzer to the very attacks it was originally designed to prevent.

8. Acknowledgments

This research was sponsored in part by the United States Army Research Office (ARO) grant number W911NF-07-1-0271.

References

[CTC2007] Comtouch Corporation. **Q3 2007 Email Threats Trend Report**. (2007) Available at: <http://www.commtouch.com/downloads/Commtouch_2007_Q3_Email_Threats.pdf>. Accessed in July 29th 2008.

[STR2005] Strauser, Kirk (2005). **The history and future of SMTP**: SMTP's adaptations to a hostile internet. The Free Software Magazine, Issue 2, USA.

[JUN2004] Jung, Jaeyeon; Sit, Emil (2004), **An Empirical Study of Spam Traffic and the Use of DNS Black Lists**. Internet Measurement Conference. Taormina, Italy

[LEV1996] Levien, R.; McCarthy, L. ; Blaze, M. (1996). **Transparent Internet e-mail security**: Technical Report. AT&T Laboratories, Murray Hill, NJ 07974. (Draft version). USA.

[GOO2007] Goodman, J.; Cormack, G. V.; Heckerman, D. (2007). **Spam and the Ongoing Battle for the Inbox**. COMMUNICATIONS OF THE ACM. Vol. 50, No. 2. USA.

[COO2006] Cook, D.; Hartnett, J.; Manderson, K.; Scanlan, J. (2006). **Catching Spam Before it Arrives: Domain Specific Dynamic Blacklists**. Fourth Australasian Information Security Workshop (AISW-NetSec 2006). Hobart, Australia

[PAN1998] Pantel, P.; Lin, D. (1998). **SpamCop: A Spam Classication & Organization Program**. Fifteenth National Conference on Artificial Intelligence, Workshop on Learning for Text Categorization. USA

[SAH1998] Sahami, M.; Dumais, S.; Heckerman, D.; Horvitz, E. (1998). **A Bayesian Approach to Filtering Junk E-Mail**. Fifteenth National Conference on Artificial Intelligence, Workshop on Learning for Text Categorization. USA

[WON2006] Wong, M.; Schlitt, W. (2006). **RFC4408: Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail**. Available at <www.ietf.org/rfc/rfc4408.txt>. Accessed in July 29th 2008.

[HAN2006] HANSELL, S. (2006). **Postage Is Due for Companies Sending E-Mail**. The New York Times. Issue of February 5, 2006. USA

[DYS2006] Dyson, E. (2006). **You've Got Goodmail**. The New York Times. Issue of March 17, 2006. USA

[PFL2005] Pfleeger, S. L.; Bloom, G. (2005). **Canning Spam: Proposed Solutions to Unwanted Email**. IEEE Security and Privacy Magazine, vol. 3, no. 2, pp. 40-7. USA.

[MIN2006] MINDLIN, A. (2006). **Google's Gmail Learns How to Spot Spam**. The New York Times, Issue of October 2, 2006. USA.

[JAC2007] Jackson, T. (2007). **How our spam filter works**. The Official Gmail Blog. Available at: <http://gmailblog.blogspot.com/2007/10/how-our-spam-filter-works.html>. Accessed in July 29th 2008.

[BLZ1999] Blaze, M.; Feigenbaum, J.; Ioannidis, J.; Keromytis, A (1999). **The Role of Trust Management in Distributed Systems Security**. Chapter in Secure Internet Programming: Security Issues for Mobile and Distributed Objects, Springer-Verlag. Germany.

[GMA2008] Gmail Team (2008). **So much time, so little spam**. Available at: <http://mail.google.com/mail/help/intl/en/fightspam/spamexplained.html>. Accessed in July 29th 2008.