E-Mail Bombs and Countermeasures: Cyber Attacks on Availability and Brand Integrity

Tim Bass and Alfredo Freyre, SAIC, Center for Information Protection
David Gruber and Glenn Watt, USAF, Langley AFB

Abstract

The simplicity of SMTP mail can be combined with the robustness of the *sendmail* MTA program and misused in numerous ways to create extraordinary and powerful e-mail bombs. These e-mail bombs can be launched in many different attack scenarios which can easily flood and shut down chains of SMTP mail servers. *Sendmail*-based SMTP mail relays also can be used covertly to distribute messages and files that could be very damaging to the integrity and brands of victims. This article discusses mail-bombing techniques, automated attack tools, and countermeasures. Also discussed is an actual Internet-based attack that was launched in 1997 on the Langley AFB SMTP e-mail infrastructure. The authors also present an analysis of the cyber attack, graphs illustrating the attack volume, and a statistical e-mail bomb early warning system.

Many variations of sendmail are used on a wide variety of systems in the Internet that facilitate the exchange of electronic mail. The basic design requirement of sendmail is simply this: no e-mail message should ever be lost. Consequently, the sendmail algorithm is extremely robust. For example, if the sendmail sending process cannot confirm that a message was delivered, the process repeatedly attempts to deliver the message [1].

In fact, sendmail is so robust that if the delivery mechanism times out when processing a large mailing list, some versions of sendmail return to the beginning of the list and resend the message to everyone. At other times, however, sendmail has locked up when attempting to deliver to noncompliant remote addresses, effectively denying service to the remainder of the mail queue [1]. The complexity and robustness of the sendmail algorithm makes it very difficult to defend against sendmail-based denial-of-service attacks.

During the first half of 1997, Langley Air Force Base was attacked repeatedly via the Internet with a wide range of automated Simple Mail Transfer Protocol (SMTP) mail bombs. Most e-mail bombs have one primary objective: flood the e-mail server so that it becomes unavailable or is unserviceable. These e-mail attacks may also be used to forge the identity of the attacker, degrade the availability of communications systems, undermine the integrity of organizations, or covertly distribute illicit material.

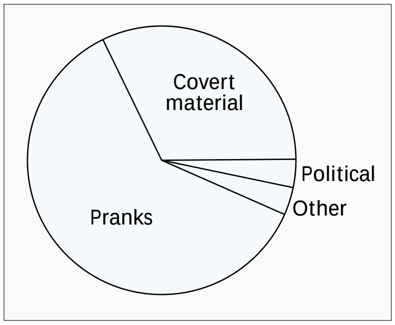

Figure 1. *Sample content of covertly distributed e-mail.*

Langley AFB actively engaged in efforts to stop sendmail-based mail transfer agents (MTAs) from being used as underground SMTP servers. E-mail servers were being used to distribute pornography and other inappropriate e-mail, as illustrated in Fig. 1. Initial countermeasures to shunt the distribution of covert e-mail resulted in large volumes of e-mail bombs directed at the MTA, graphically illustrated in Fig. 2, which became known as the *Langley Cyber Attack*. This article describes the actual Langley Cyber Attack, e-mail bombing techniques, mail-bombing tools, and countermeasures.

The following section provides a brief review of the technology and an in-depth discussion on e-mail bombing techniques. The Langley Cyber Attack, the countermeasures used, and the early warning system designed to alert against the attack are discussed in the section after that, which also presents a brief analysis of the results. In order to be complete, we then briefly discuss cryptographic e-mail bomb countermeasures. An appendix provides an overview of a few automated e-mail bombing programs widely available via the Internet.

Electronic Mail Bombs

There are many e-mail bombing tools and techniques freely available to the public. There are also myriad reasons for cyber-citizens to send e-mail bombs. For example, according to a recent survey [2], 5 percent of the recipients of Internet junk mail retaliate by sending mail bombs

0890-8044/98/$10.00 © 1998 IEEE

and executing other denial-of-service attacks.

In this section we review many of the techniques exploited and used for creating powerful e-mail bombs on the Internet. These techniques include *chain bombs*, *error message bombs*, *covert distribution*, and *mail exploder exploitation*. There are other techniques as well. This section covers a large subset of these e-mail bombing techniques.

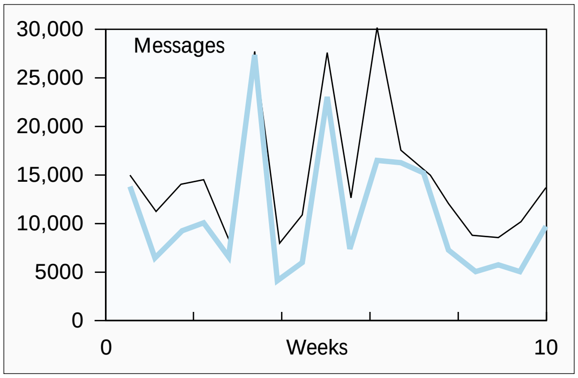

Figure 2. *Daily volume of mail bombs (shaded) vs. total e-mail.*

A Brief Review of E-Mail Concepts

Transferring e-mail between users on a single machine is a relatively trivial process. However, transporting e-mail between a wide array of TCP/IP hosts and servers globally across the Internet can be quite complex. The protocol used to transport mail across the Internet is SMTP [3], but the actual transport management of global e-mail requires a much more complex electronic infrastructure.

Servers and programs with a primary function of storing and forwarding e-mail across the vast canyons of cyberspace are referred to as MTAs. An MTA is a very specialized program which delivers and transports e-mail between mail servers. On the other hand, a *mail user agent* (MUA) may be any one of a vast number of programs users execute to read, compose, reply, and manage e-mail locally. The widely used sendmail program is an MTA [4].

SMTP messages consist of lines of ASCII text of information in a rigid format followed by the main body of the message. The rigidly formated section is known as the *header*. RFC 822 [5] defines the syntax and specification of the SMTP header, created to allow simple parsers to process the general structure of messages without knowledge of the detailed structure of individual header fields.

The sendmail MTA was designed to manage a very complex internetworking environment and is one of the cornerstones of the Internet [4]. One of the myriad technical requirements for sendmail was the necessity to process addresses that are in *route address* syntax [4, 6]. These addresses follow a syntax created for directed source message routing:

$$@MTA_1, @MTA_2, @MTA_n : user@MTA_{n+1}$$

In route address syntax, this expression translates to:

- The originator sends the message to MTA_1
- MTA_1 sends the message to MTA_2
- MTA_2 sends the message to MTA_n
- MTA_n sends the message to MTA_{n+1}
- MTA_{n+1} delivers the message to user

The capability of sendmail to process route addressing is but one of numerous uncountable special rules and expressions which a robust universal MTA must manage. The next section discusses how this syntax is exploited to create extraordinary e-mail bombs. The reader is kindly referred to Brian Costales and Eric Allman's excellent book on sendmail [4] for further reading on the sendmail MTA and its capabilities.

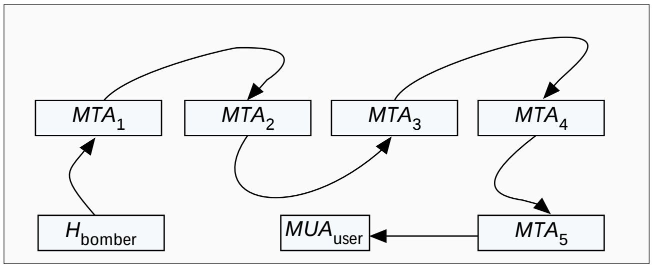

Figure 3. *Sendmail route address chain bombing.*

Chain Bombs

Most sendmail configurations will process e-mail addresses which are in route address syntax, described above. The e-mail bomber exploits the route address functionality to create a very powerful e-mail bomb we refer to as *chain bombing*. Figure 3 illustrates the chain bomb vulnerability.

In the chain bomb scenario, the e-mail bomber, H_{bomber}, executes an automated script with a chain of source routed SMTP messages. The e-mail bombs are delivered and queued on the first MTA in the chain, MTA_1. If the attack volume of the e-mail bomb is sufficient to inhibit or deny service to MTA_1, the remaining messages in the outbound queue of the bombing host will be directed automatically to MTA_2. This process continues for all the MTAs.

If MTA_1 successfully queues the e-mail from the bombing host, the bomb is delivered to the next route address in the chain, MTA_2. The process is repeated through all the MTAs in the chain, either successfully queuing the entire e-mail bomb or queuing a percentage of the volume of the bomb, then moving to flood the next MTA in the chain.

Depending on the configuration of the MTAs, service may be denied due to numerous factors. For example, when the volume of mail in the MTA queue is extremely large, the number of available file descriptors can exceed operating system parameters or the number of open TCP connections reach system limits. The large volume of e-mail in the MTA queue must be systematically moved out of the queue, both operational and malicious e-mail, and the MTA restarted or the system rebooted. Sorting malicious from important business e-mail is difficult and very resource-intensive.

Examples from the Langley Cyber Attack reveal that unsuspecting mail system administrators are often unaware that the MTA has been attacked by a mail bomb and simply reboot the mail server without clearing the malicious messages from the MTA queue. In this case, the sendmail process will reinitiate the process again, attempting to deliver the bomb to the next MTA in the route address chain.

Error Message Bombs

E-mail bombers also exploit the feedback mechanisms of mail systems by using legitimate error messages generated by MTAs. Figure 4 illustrates how an MTA is exploited by masquerading the source address of the sender so that the system responds with error messages delivered to the MTA of the victim.

In this attack scenario, the bomber inserts the e-mail address of the victim's e-mail server, MTA_2, as the origin of the message, and sends the e-mail bomb to MTA_1. MTA_1 was configured to generate feedback messages to the originator when one of many error conditions are generated.

MTA_1 generates an error message or, in the case of an e-mail bomb, large volumes of error messages, and forwards them to the victim, MTA_2. Depending on the robustness and configuration of MTA_2, either MTA_2 is taken out of service, or the end user's mailbox, MUA_{user}, is flooded.

Many well-intended system administrators accidentally configure their systems to be exploited in this manner. One scenario occurs when an MTA has been e-mail-bombed and the system administrator configures the mail server to send rejection messages on receipt of unwanted messages. An Internet-based attacker simply inserts the unauthenticated e-mail address of the victim in the SMTP message and mail-bombs an innocent mail server. The intermediate server innocently responds with error messages, bombing the victim.

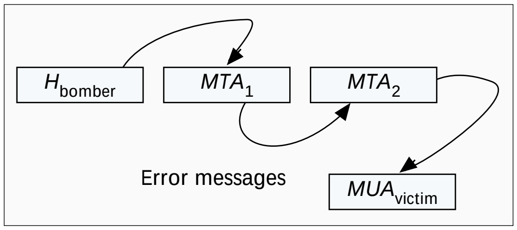

Figure 4. *Error message bombing.*

Covert Distribution Channels

Guluc and Tsudik [7] discuss potential anonymous abuse of remailers for the purpose of "spreading libelous accusations, hate-filled propaganda, pornography, and other unpleasant content." The content of the illicit e-mail uncovered at Langley AFB validated this statement and identified a larger systemic problem.

The technique of anonymously distributing covert files via a neutral intermediate MTA is illustrated in Fig. 5. The covert distributor, $H_{\text{distributor}}$, uses route address syntax and insecure sendmail configurations to relay illicit material to other MTAs. We have seen this technique used for private individuals, the general public, and unsuspecting network users.

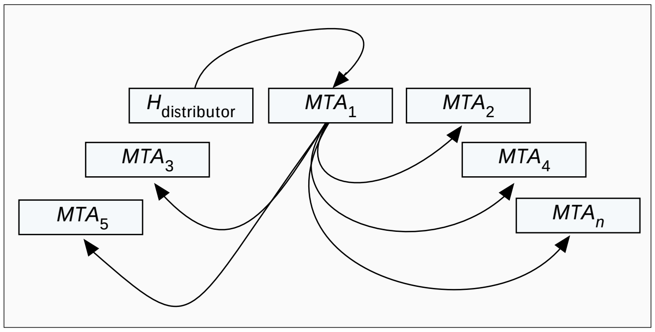

Figure 5. *Covert distribution channels.*

In addition, the recipient of the illicit mail can easily be fooled to believe that the e-mail originated from an innocent victim's host machine. This poses a very real and dangerous method for criminals and malicious agents to victimize the Internet community. For example, an MTA for a large bank in Tokyo could be used as a relay by pornographers. The recipient of the e-mail would more than likely (falsely) believe that the bank was the originator of the illicit mail. This type of accusation is very difficult to defend against and could be extremely damaging to the integrity of the bank's brand and reputation.

Exploiting Mail Exploders and List Servers

A *mailing list* is a community of e-mail addresses that can be reached by sending a single message to one address, known as the *list address*. E-mail sent to the automated mailing list is redistributed to all subscribers to the list [8]. Automated list servers like Majordomo or ListProcessor provide many opportunities for the e-mail bomber to exploit the SMTP infrastructure (Fig. 6).

This attack scenario can be combined with other bombing techniques or executed standalone. In a nutshell, the bomber subscribes the victim, H_{victim}, to numerous mailing lists. Currently, most mailing lists do not authenticate the subscriber; and the list servers which do use weak authentication mechanisms that may easily be subverted.

Herfert [9] discusses security-enhanced mailing list exploders as a way to provide strong authentication to posters on mailing lists. However, these cryptographic techniques are difficult to implement on a global scale, primarily because of the challenges associated with key management. In addition, the processing overhead of encrypting data for thousands of e-mail messages must be considered. The topic of cryptography relative to e-mail authentication is briefly summarized in the fourth section.

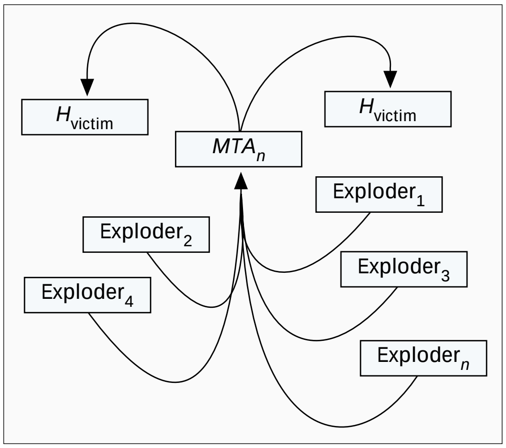

Figure 6. *Abuse of mail exploders.*

The Langley Cyber Attack

In January 1997, a commander within Air Combat Command (ACC) received an inflammatory e-mail message, apparently from President Clinton. The commander immediately understood that someone was using SMTP mail to impersonate the President. The chief of the information protection branch was directed to investigate the situation.

The first reaction to the forged e-mail at Langley was to examine the log files of the sendmail-based MTA. However, like most systems administered with limited resources, the level of auditing and logging of the MTA had been configured to the minimum possible setting to save disk space. The investigators increased the sendmail audit configuration to provide the maximum amount of logging information possible.

The investigation into the Langley SMTP infrastructure uncovered a larger systemic problem. SMTP MTAs, accessible from the public Internet, were being used covertly to distribute large volumes of pornography, bigoted hate mail, and other unacceptable and criminal messaging [10]. This discovery initiated a concentrated effort to stop all malicious use of the SMTP infrastructure while simultaneously ensuring that all legitimate SMTP mail traffic was delivered. To accomplish this objective, Langley AFB installed a simple rules-based filter, which preprocessed all incoming and queued SMTP mail [10]. These countermeasures were successful in preventing illicit use of the MTAs. The rules-based filter is discussed in more detail later.

The results of this investigation into the Langley Cyber Attack have gained national media attention. The results were discussed extensively in the United States Air Force (USAF). In addition, the chairman of the President's Commission on Critical Infrastructure Protection referenced the Langley Cyber Attack as a critical example of an actual international cyber attack [11] via the Internet. Commercial organizations have since reported millions of dollars in damages resulting from forged SMTP mail originating from the Internet [12].

Interviews with the system administrators (SAs) of the SMTP MTAs also uncovered very interesting information. The MTA often “locked up,” according to inexperienced SAs, but they never investigated the cause of the service disruptions nor examined log files to determine the cause of the system failures. Routinely, the SAs casually rebooted the platforms and continued performing other duties that were assigned a higher priority.

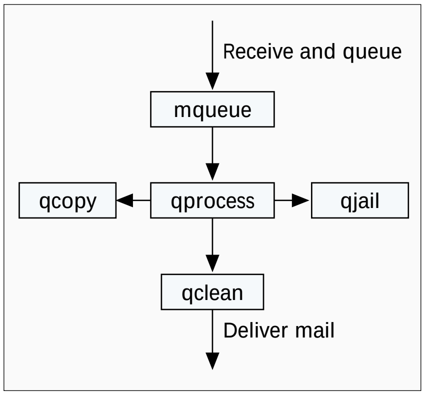

Figure 7. *Process flow diagram for the SMTP filter.*

Examining the log files in real time showed a much larger problem on the enterprise than isolated e-mail spoofing. Large volumes of e-mail originating in the Internet (Fig. 2) were being delivered to other Internet sites via the SMTP MTAs. Further examination of the e-mail being relayed covertly led to the discovery that incredible volumes of pornographic materials was being distributed via the Langley MTA to users at commercial Internet service providers. The largest number of targeted users were America Online subscribers.

Initial Countermeasures

The need to learn the identity of the originator of the *clinton@whitehouse.gov* message was overshadowed by the revelation that an MTA at Langley AFB was being used to covertly and illegally distribute pornography and hate mail. Figure 1 illustrates sample content of the illicit mail covertly relayed via the MTA. The concern for the reputation and *brand* of the enterprise became the primary requirement. If this underground distribution channel gained high media viability, undermined public confidence could result in potential damage to the organization. The basic requirements of the countermeasures were nicknamed the *Black Hole Strategy*:

- Do not provide any feedback or error messages to the hackers or mail bombers.
- Capture and minimize delivery of illegitimate mail using a rules-based filter.
- Copy suspect mail for future analysis, delivering legitimate e-mail robustly and quickly.
- Keep and maintain all captured messages as potential forensic evidence.

The initial reaction of novice network administrators is to use IP firewalls and routers to block the apparent source addresses of the e-mail bombs. However, experienced network managers familiar with mail-exchange records (MXs) [4] and the robustness of the sendmail MTA understand that the address-blocking technique will not work in the vast majority of cases. In fact, chain bombing and other relaying techniques make most attempts to block specific IP addresses relatively futile; traditional firewalls are simply ineffective. Finally, when legitimate relays are used by bombers and hackers, attempts to block these addresses result in self-styled “denial-of-service attacks”; this is an easily exploitable countermeasure (see box).

Filtering Queued E-Mail

The technical strategy of our countermeasure against mail bombing was simply to queue incoming mail messages, filter the mail based on developed rule sets, and forward the *clean* mail. The filtering rule sets triggered on information in the header control files of the mail messages. The message content was not used in the filtering process. All filtered mail was processed via one of two paths. Mail was either sent to “jail,” *qjail*, and not delivered, or copied into *qcopy* for further analysis (Fig. 7). Denying direct feedback to hackers was the cornerstone strategy.

During the initial filter prototyping phase, we copied and delivered all mail with the keyword “whitehouse” in the header fields because it was theoretically possible that valid mail could come from “whitehouse.gov.” This was the prototype of a filter refinement queue which would become *qcopy*. This queue would be used to fine-tune additional rule sets. All captured mail that was *taken prisoner* was stored in *qjail*.

Figure 7 illustrates the flow of events for the filtering process. The SMTP server is started with the **-odq** switch [4], instructing sendmail to receive and queue incoming mail, *mqueue*, only.[1] The filter program is executed by **crond(8)**. The program processes the sendmail mqueue by first copying all messages in the queue to another directory. It is in this directory that the files will be processed. Incoming mail continues to arrive in mqueue and the filter program processes the staging queue, *qprocess*. There are timers and a state machine to avoid moving e-mail which has not been completely received and queued by the MTA.

The filter rule sets first look in the header files (the qf files) for spoofed addresses without the @ character. These files are moved out of the qprocess into qjail and are not delivered. In addition, a subset of sender addresses with @ characters are copied to qcopy and left in the staging queue, qprocess. The remaining messages in qprocess are moved to another queue, *qclean*, and sendmail delivers the mail by executing with the **-q** switch, instructing sendmail to process the mail queue; and the **-Q** switch, specifying which queue to process.[2]

Sendmail Countermeasures

It is possible to configure the sendmail anti-spamming features defined under the check_ or checkcompat() rule sets as countermeasures against many e-mail bombing techniques. For example, these features allow sendmail to be configured; to not perform as a mail gateway, limit the size of messages, and reject certain sites known to send e-mail bombs [4].

The checkcompat() routine requires modification of the sendmail source code, and therefore is quite difficult for the average systems administrator to implement. Beginning with sendmail v. 8.8, limited checking and rejecting can be accomplished with four rule sets: check_mail, check_rcpt, check_relay, and check_compat. For more information on the features, please refer to [4].

During the Langley Cyber Attack, we found the sendmail built-in functions difficult for the average systems administrator to configure. The built-in filtering provisions were not flexible enough to meet all of our Black Hole requirements. In fact, inexperienced users of these rule sets inadvertently configured sendmail to serve as a relay for e-mail error bombs because of the error messages generated by the MTA when under attack. Writing a rules-based filter which processed the mail queue was found to be the best defense against e-mail bombs during our engagement.

However, sendmail is evolving, and it is of quintessential importance to use the latest release of sendmail in all MTAs. Sendmail developers have been refining the software to provide more user-friendly configuration options that can help mitigate e-mail bombs.

[1] *Example: /usr/lib/sendmail -bd -odq.*

[2] *Example: /usr/lib/sendmail -q -oQ/usr/spool/qclean.*

Rules-Based Filtering

Initial efforts to track down spoofed e-mail were the genesis of the rules-based filter, which began very simple and became much more complex over time. The technical strategy was to filter incoming e-mail messages based on refined rule sets and forward the clean mail. Rule sets triggered on information in the header control files of the mail messages. The message content was not used in the filtering process. All filtered mail was processed via one of two paths.

In the early hours of the filter prototype, we discovered that a majority of e-mail bombs contained invalid originator SMTP addresses without the @ sign. Our first prototype filter jailed all messages with invalid SMTP address formats. However, our cyber-bomb opponents adapted, creating bogus addresses with random @ symbols. Our filters were actively probed by bombers across the Internet (and across the globe) in order to understand how they were implemented [10]. A small code fragment (see the box on this page), written in PERL, from the first alpha version filter prototype illustrates simple rule-set processing on the header files.

```
# author: Tim Bass Jan. 23, 1997

# Sendmail will run with the ‘queue only
and
# not deliver’ option, -odq, as a daemon
# process (sendmail -bd -odq).
# This script is executed by crond
# Currently executed every five minutes

# open the qf files

open(QF,”<$Header”)||  die  “Can’t  open
files\n”;

# search the qf files for interesting things
# currently, only doing minimal filtering on
# the sender. It is easy, however, to fil-
ter
# on other header fields, as required.

while(<QF>){

# the /^S/ expression looks in the qf file
# for the line beginning with S, the sender

if(/^S/){

# then we look in that line for senders
# without @ signs

if(!/\@/){

# and we move the suspected header and data
# files to the suspected queue, qjail

if(!/<>/){
     rename($Header,$JailFile);
     rename($Data,$JailFile2);
     }
}

# and we copy these suspected senders with
data # files to the copy queue, qcopy (just
in case)

elsif(/whitehouse/i){
     system(“/bin/cp $Header $CopyFile”);
     system(“/bin/cp $Data $CopyFile2”);
          }
     elsif(/whistleblower/i) {
     system(“/bin/cp $Header $CopyFile”);
     system(“/bin/cp $Data $CopyFile2”);
     }

close(QF);  } } }
```

The bombers were adapting to new filter rule sets within 24 to 48 hours. The most common bomb signature was repeated e-mail with the same sender-receiver pairs. With few exceptions, this fact provided a successful indicator of hostile e-mail. We refined the filtering algorithm based on this observation and provided the programming requirements to the development team. This algorithm became a key element in mitigating numerous types of e-mail bombs during attacks.

It is important to briefly point out that there are occasional exceptions to most filter rules. Trapping all of the exception conditions at the application layer can prove to be a difficult (perhaps impossible) programming challenge. It is impractical, from a resource perspective, to obtain a 100 percent solution. However, the trade-off of jailing a very small percentage of good e-mail vis-a-vis minimizing the risk and exposure to the enterprise was a management decision, not a technical one.

The prototype filter design also provided valuable information which assisted investigators in discovering the content, type, and origin of the criminal e-mail. Trapping and jailing politically charged mail became a high priority as our team began refining filter rules based on illicit e-mail origin, SMTP mailer, originator, sender-receiver pairs, and so on. Our software development team provided numerous enhancements to the original filter prototype which resulted in reduced countermeasures manpower and improved filter granularity. The speed of the sendmail process receiving and forwarding mail in real time made queuing the messages prior to filtering necessary. The additional latency in the mail-filter processing was actually offset by performance improvements in the SMTP infrastructure. However, we did not quantify these observations.

In addition, by queuing the messages the entire SMTP header file and control messages could be used in the filter process. This proved to be extremely valuable in the process of examining mail bombs, understanding the nature of the attacks, and simultaneously insuring all e-mail was correctly delivered with minimal delay.

Mailbomb Early Warning System

As the SMTP filter used against the Langley Cyber Attack matured, SMTP mail denial-of-service (DoS) failures discontinued. However, the probability of overwhelming DoS attacks remained. Thousands of rogue e-mail messages continued to bombard Langley AFB servers. This bombardment created a pseudo steady-state condition of background noise. This steady-state bombardment was the basis for the implementation of our e-mail bomb early warning system.

Using a standard mathematical process we were able to identify an ongoing attack and provide a reasonable basis for predicting non-random future attacks. Beginning with the initial version of the SMTP filter, the investigating team automatically collected e-mail statistics on the daily volume of e-mail (T_d), jailed e-mail (J_d), questionable e-mail, and other variables of interest.

$$\Omega = \frac{\sum_{d=1}^{d=n} J_d}{\sum_{d=1}^{d=n} T_d} \tag{1}$$

By keeping the size of each weekly subgroup, n, constant (seven days), a pattern began to emerge as we graphed and analyzed the raw data. Clustering one-week averages of jailed e-mail as a percentage of total e-mail volume (Ω in Eq. 1) led to the theory that a statistical process control chart with an upper and lower control limit might serve as an e-mail bomb early warning indicator.

$$\bar{\chi} = \frac{\Omega}{n} \tag{2}$$

$$R = \chi_{\max} - \chi_{\min} \tag{3}$$

Daily e-mail statistics were automatically collected and averaged over weekly periods. $\bar{\chi}$ denotes the daily average jailed e-mail percentage based on a one-week clustering interval. In addition, the range of the e-mail bomb volume, R, was calculated (Eq. 3) from the computed average minimum and maximum.

The size of each weekly subgroups, n, remained constant (seven days). The total number of weeks analyzed was represented by K. The overall average daily e-mail bomb volume, $\bar{\bar{\chi}}$, was also tracked (Eq. 4).

$$\bar{\bar{\chi}} = \frac{\sum \bar{\chi}}{K} \tag{4}$$

From this simple statistical process, e-mail bomb upper and lower control limits were established. Figure 8 illustrates ten weeks of these statistics as they were collected and analyzed.

The upper and lower control limits (UCL, LCL) were calculated by taking a 2 percent standard deviation above and below the average traffic (including attack and legitimate e-mail) volume. The decision to use two standard deviations exceeds the generally accepted 1 percent for normal distributions. However, a control limit that is too narrow results in frequent searches for insignificant e-mail bomb attacks (false alarms), which is an inefficient use of human resources.

Control limits (Fig. 8) that are too wide would permit undetected significant e-mail bomb attacks. We simply estimated the initial UCL and LCL from inspection of the graphs. However, as an early warning process matures and each sample outside the prescribed limit is identified and eliminated, the variability from mean to mean should diminish from the initial value. At this point, new upper and lower limits should be calculated. These limits converge to a point where an alarm is sounded at the very start of an attack as the system matures.

A running average exceeding the UCL indicated that a significant e-mail bomb attack was occurring. If the average fell below the LCL, either the number of bogus e-mail had significantly dropped or the filters were failing.[3] At this point, adjustments were made to the filter rules to eliminate the process alarm.

The first indication of a control limit breach surprised the team. Instead of a massive counterattack from the mail bombers as had happened in the past, the first Action Trigger (Fig. 8) occurred on the LCL. This caused the team to examine the queues in an attempt to determine what the hackers were up to. Because the Langley AFB MTA had been hardcoded into several automated tools (see the appendix), those tools could no longer bomb victims because the Langley filter removed the illegitimate messages from the Internet. Langley investigators found that the authors of the tools had removed Langley AFB from the list of MTAs preconfigured into the bombing programs. The Black Hole Strategy proved to be an effective and proven countermeasure which we highly recommend.

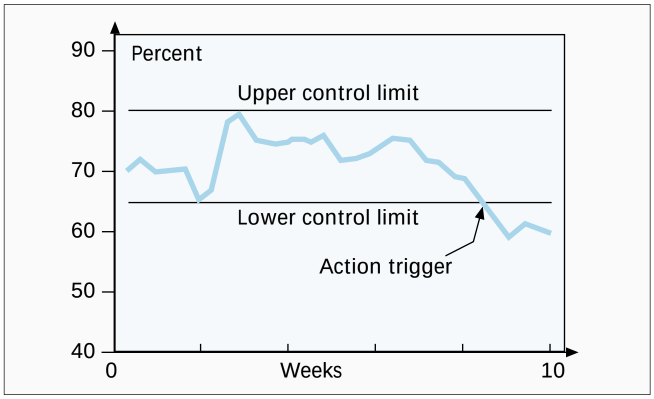

Figure 8. *Early warning system: event trigger.*

We stopped experimenting with control charts at that point in time. However, our next step would have been to develop a control chart based on the variance, standard deviation or the range. This is an excellent area for further research and analysis.

Cryptographic Countermeasures

Many of the technical vulnerabilities which make mail bombing a serious threat could be significantly mitigated by enhancing the security infrastructure of the Internet. Published by Linn in 1988, RFC 1040 [13] discussed SMTP message encryption and authentication procedures. His continuing work in RFC 1115 [14] specified the cryptographic algorithms to support Privacy Enhanced Mail (PEM), including the use of public-key encryption algorithms.

Presently, the PEM specification is contained in four RFCs which are all a part of the "Privacy Enhancement for Internet Mail" series (RFCs 1421-1424):

- Part 1: Message and Authentication Procedures
- Part 2: Certificate-Based Key Management
- Part 3: Algorithms, Modes, and Identifiers
- Part 4: Key Certification and Related Services

The core security problem with e-mail bombs is the *authentication* of the originator. PEM (and similar cryptographic e-mail services) offer both symmetric and asymmetric authentication, supporting a wide variety of cryptographic algorithms[4] [15]. Methods for processing mail addressed to mailing lists are also provided; however, cryptographic authentication remains problematic for e-mail transport [16]. The reader is referred to the many references on the subject [15, 16] for a more detailed technical discussion.

Unfortunately, PEM provides integrity protection only on the body of a message. The header fields of an SMTP message are not protected because MTAs need to modify many of the header fields during e-mail transport [17]. As pointed out earlier, the entire SMTP infrastructure relies on a complex, heterogeneous internetwork of MTAs and MUAs. Therefore, cryptographic solutions which work robustly with intermediate systems are very difficult to design. Scalability and interoperability become complex technical issues which are very expensive to design, implement and sustain.

Scalability and interoperability concerns are also major obstacles in the global management of cryptographic keys, known as the emerging *public key infrastructure* (PKI). Combining a heterogeneous PKI with a robust global MTA infrastructure will provide the developer community with cryptographic tools to address e-mail sender authentication. However, by no

[3] *Perhaps the "Borg" had adapted to our filters again.*

[4] *Many MUAs have built-in or plug-in cryptographic functionality (non-PEM), including Netscape's Messenger, MS Exchange, and Eudora, to name a few.*

means will a PKI solve the e-mail bomb threat without MTA and MUA integration. Finally, PKI–MTA integration may not significantly mitigate the e-mail bomb problem because PKI was designed to address confidentiality, authentication, integrity, and nonrepudiation, not DoS attacks.

All technologies that present new opportunities also propagate new vulnerabilities and risks. Emerging public key cryptosystems are traditionally viewed as defensive mechanisms, strengthening the integrity, confidentiality, and authentication of our electronic infrastructure. However, the widespread availability of cryptosystems creates potential offensive threats to the infrastructure which are normally considered after design and deployment. The interested reader is referred to an excellent paper by Young and Yung [17] which discusses cryptovirology and cryptoextortion, emerging topics beyond the scope of this article.

Summary

E-mail bombing and impersonating the sender have become common crimes in cyberspace. The global networking infrastructure is used as a basis to attack the integrity of unsuspecting victims. Disgruntled employees, terrorists, and industrial competitors can use network-based e-mail bombing techniques to undermine confidence in organizations of trust. This type of attack on brand integrity has the potential to cause major financial damage to institutions, including financial services institutions, banks, insurance companies, pharmaceutical companies, publishing companies, law enforcement, government institutions, *ad infinitum*.

Our team uncovered and stopped a covert channel for the illegal distribution of pornography, hate mail, and pranks via standard operational SMTP MTAs. We actively engaged to protect the reputation, integrity, and brand of the organization. Unfortunately, it is extremely difficult for the general public to differentiate between the abuse of legitimate resources by a hacker, terrorist, or criminal and direct misuse or negligence by an organization or commercial corporation. When the public incorrectly perceives that a large multinational business is distributing illicit material from its e-mail servers, this perception will undermine the integrity, confidence and trust of the business. The results could be devastating.

Tools for launching e-mail based attacks are dangerous, easy to use, and freely available on the Internet. Cryptographic mechanisms to authenticate e-mail are emerging. However, the ease with which an attacker might abuse and misuse the e-mail infrastructure of both commercial and federal organizations puts these organizations at significant risk today.

Finally, there exists a perpetual enigma in the Internet community regarding computer and network security. Many professionals are of the opinion that “security through obscurity” is the better approach to managing information security risks, where containment of vulnerabilities is preferable to open discourse. On the other hand, there are equally passionate opinions for “security without ambiguity”: it pays to have the global community engaged as an open cyber-society, solving security challenges together. It is our hope that this article helps, in some small way, to forward the overall goals and objectives of the Internet community.

Acknowledgments

We are very grateful to the USAF, Air Combat Command, for the interesting and stimulating networking engagements at Langley AFB. In addition, the numerous reviewer comments have significantly improved this article.

References

[1] P. Neumann, “The Computer-Related Risk of the Year: Distributed Control,” *IEEE Proc. Comp. Assurance*, June 1990.
[2] J. Berst, “Front-Line Report from the War on Spam,” *ZDNet*, Aug. 19, 1997.
[3] J. Postel, “Simple Mail Transfer Protocol,” Internet RFC 821, Aug. 1982.
[4] B. Costales and E. Allman, *sendmail*, 2nd ed., Sebastopol, CA: O’Reilly & Associates, 1997.
[5] D. Crocker, “Standard for the Format of ARPA Internet Text Messages,” Internet RFC 822, Aug. 1982.
[6] F. Avolio and P. Vixie, *Sendmail: Theory and Practice*, Newton, MA: Butterworth-Heinemann, 1995.
[7] C. Gulcu and G. Tsuski, “Mixing Email with Babel,” *IEEE Proc. Symp. Networks and Distr. Sys. Security*, 1996.
[8] C. Liu, *et al.*, *Managing Internet Information Services*, Sebastopol, CA: O’Reilly & Associates, 1994.
[9] M. Herfert, “Security-Enhanced Mailing Lists,” *IEEE Network*, May/June 1997.
[10] T. Bass and Lt. Col. G. Watt, “A Simple Method for Filtering Queued SMTP Mail,” *Proc. IEEE MILCOM ’97*, Nov. 1997.
[11] “U.S. Seen as Open to Cyber Attack,” *CNN Interactive*, Associated Press, Oct. 8, 1997.
[12] “Samsung Stung for Millions by Internet Fraud,” *CNN Interactive*, Newsbytes, Aug. 11, 1997.
[13] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part I — Message Encryption and Authentication Procedures,” Internet RFC 1040, Jan. 1988.
[14] J. Linn, “Privacy Enhancement for Internet Electronic Mail: Part III — Algorithms, Modes, and Identifiers,” Internet RFC 1115, Aug. 1989.
[15] W. Stallings *Network and Internetwork Security Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, 1995.
[16] Kaufman *et al.*, *Network Security: Private Communication in a Public World*, Upper Saddle River, NJ: Prentice Hall, 1995.
[17] A. Young and M. Yung, “Cryptovirology: Extortion-Based Security Threats and Countermeasures,” *Proc. 1996 IEEE Symp. Security and Privacy*, 1996, pp. 129–40.
[18] D. Parker, “Automated Crime and Security,” IEE Euro. Conf. Security and Detection, Apr. 1997.

Additional Reading

[1] B. Schneier, *Email Security: How to Keep Your Electronic Messages Private*, New York: Wiley, 1995.
[2] E. Sullivan, “Put a Spammer in the Slammer,” *PC Week*, Dec. 1, 1997.
[3] T. Bernstein *et al.*, *Internet Security for Business*, New York: Wiley, 1996.

Biographies

TIM BASS (bass@silkroad.com) is a technical director with SAIC, Center for Information Protection. He graduated from Tulane University in 1987 with a B.S.E. with Departmental Honors in electrical engineering. He played a principal role in building the SprintLink Integrated Network Management Center, the Sprint Managed Router Network organization, and the original Base Network Control Center (NCC) prototype for the USAF. He directs global efforts with SAIC for engagements with very large financial institutions, publishing enterprises, and insurance companies in complex network architectures and global information protection assessments.

ALFREDO FREYRE (acf@cip.saic.com) is a technical director with SAIC, Center for Information Protection. He graduated from Carnegie Mellon University in 1994 with an M.S. in information networking. He also holds a B.S. with Departmental Honors in electrical engineering from Brown University. He specializes in the identification, application, and prevention of computer crime executed using automated hacker tools. He is currently engaged with very large financial institutions in the area of vulnerability assessments and security architecture analysis and design. His prior experience before coming to SAIC includes Bell Communications Research and AT&T Bell Laboratories.

LT. COL. DAVID GRUBER (david.gruber@langley.af.mil) is chief of the Networks Branch at Air Combat Command and provides operational guidance, policy, and funding for classified and unclassified networks supporting over 100,000 people. He graduated with a B.S. in electrical engineering from the United States Air Force Academy and earned an M.S. in systems management from the University of Southern California. While commanding an Air Force Communications Squadron, he designed and implemented the first network control center in support of the initial Air Expeditionary Wing deployment to Southwest Asia.

LT. COL. GLENN WATT (glen.watt@langley.af.mil) is deputy division chief for USAF HQACC/SCN, Network Systems. He holds a B.S. in mathematics from Kutztown State College and an M.S. in computer science from Lehigh University. In addition, he is a Certified Information Systems Security Professional (CISSP). He has spent the last 10 years involved with defensive information warfare at all levels from national defense organizations to Air Force headquarters. He developed and implemented the original Secret-to-Unclassified Network Guard (SUNG) at USSTRATCOM. For more details on obtaining SMTP filter software, please contact Lt. Col. Watt at the above e-mail address.

Appendix: Automated Tools and Mail Bombs

Parker [18] defines an automated crime as one "executed entirely by one or more sequentially executed computer programs in a computer" or computer network.

"We must anticipate increasingly sophisticated automated crime, and the packaging of easy-to-use, free-ware computer programs that can be executed by almost anybody for fully automated criminal activities." [18]

Mail-bombing tools are examples of automated crime. These tools are dangerous, easy to configure, and widely available on the Internet. The potential e-mail bomber simply points a Web browser at a search engine and performs a keyword search for e-mail bombing programs. A search will return pointers to programs such as Voodoo, Unabomber, KaBoom, Up Yours, and Avalanche, described in this appendix. These examples are just a few of the well-known mail-bombing programs that have surfaced on the Internet in the past few years. Each of these tools has one primary objective: flood the mail server so that it becomes unavailable or is unserviceable.

Automated mail-bombing programs vary in features and functionality. One group is more flexible in the construction and configuration of the mail bombs. Other tools allow random messages and SMTP headers rather than a statically identifiable one. In addition, the bombing programs vary in installation and execution

Many of these bombing tools come with professional-quality graphical user interface (GUI) and very professional documentation. At the other end of the spectrum, there are bombing utilities that are executed via the command line and provide little or no documentation. Most mail-bombing tools attempt to provide full anonymity to the user. The next few subsections describe a few of the well-known and freely available automated bombing tools available on the Internet.

Unabomber

The Unabomber is a Windows 95 mail-bombing program that was developed by *Dead Elvis* in 1996. Upon execution, a picture of the Unabomber (the hooded sweatshirt figure with dark glasses) appears before the main program window opens. Unabomber has a GUI (main program window) that allows a user to send multiple copies of the same message to a single recipient. From the main program window, the user may construct the contents of the mail message and select the number of copies (bomb size) to send. Unabomber also "features" an online Help menu containing detailed information on program installation and use.

KaBoom

KaBoom is a Windows 3.x and Windows 95 mail-bombing program developed by *The Messiah of The Alliance*. Similar to Unabomber, Kaboom allows the bomber to construct an e-mail message and send multiple copies of a message to recipients. Kaboom identifies 62 anonymous servers (i.e., servers that will bounce e-mail anonymously) to assist the bomber to operate covertly. Kaboom also has built-in functionality which can be used to subscribe an SMTP user to 48 different mailing lists.

Up Yours

First released (v. 1.0) in May 1996 as a sample program to illustrate the functionality of Visual Basic (VB) controls, Up Yours is a Windows 95/NT mail-bombing program. The current release (v. 3.0) requires the Microsoft Internet Control Pack and Visual Basic runtime files. Up Yours "features" a random insult generator that can generate 50 million different insults. Similar to KaBoom, Up Yours comes with a mailing list subscription attack and anonymous SMTP server "features." In addition, Up Yours supports *mailcheck*, which assists the bomber in identifying other vulnerable SMTP servers.

Avalanche

Avalanche is a Windows 3.x and Windows 95/NT mail-bombing program that was developed by *H-Master*. Unlike the other bombers, Avalanche comes with a number of configuration files that permit the attacker to customize, create, and select random mail headers and messages. Using a sophisticated GUI, the bomber can select the number of mail messages to send or force the program to send messages continuously until explicitly stopped. For anonymity, Avalanche "features" fake mail headers with several built-in anonymous SMTP servers. Avalanche is distributed with over 20 pages of documentation consisting of a detailed user's guide, a *Tips for Bombing* tutorial, and an *Addon Implementation Guide*. The Addon support functionality is a unique feature of Avalanche, which permits the bomber to add new attacks and functionality to the tool without recompiling the source code. Also, similar to KaBoom and Up Yours, Avalanche can be used to subscribe Internet citizens to numerous mailing lists without their knowledge.

Voodoo

Voodoo is a UNIX mail-bombing program that was released in 1996. It is a small easy-to-use command-line program that is supported on both SunOS and Linux. The current release of Voodoo permits the bomber to send 99 consecutive mail bombs to a victim. However, the basic program can be combined with shell scripts and other software to create much more damaging and comprehensive e-mail bombs.