Global Information Assurance Certification Paper

Advanced Incident Handling and Hacker Exploits

GCIH Practical Assignment: Version 2.1a

Exploit In Action

SMTP Loop Moderate Denial of Service:

InterScan VirusWall NT & Lotus Domino Environment

Submitted By: Brian Roberts
SANS Online Course
January 24, 2002

TABLE OF CONTENTS

EXECUTIVE SUMMARY 3

SECTION 1: THE EXPLOIT 3

LOTUS DOMINO MAIL LOOP DENIAL OF SERVICE VULNERABILITY 3
Name *3*
CVE Candidate *3*
BID *4*
Operating Systems *4*
Application *4*
Protocol *5*
Brief Description *7*
Variants *8*
References *9*
TREND MICRO INTERSCAN VIRUSWALL NT INFINITE LOOP VULNERABILITY 10
Name *10*
TrendMicro Knowledge Base *10*
Operating Systems *10*
Application *10*
Protocol *11*
Brief Description *12*
Variants *13*
References *14*
THE COMBINATION 15

SECTION 2: THE ATTACK 15

DESCRIPTION AND DIAGRAM OF NETWORK 15
Network SMTP Traffic Flow *16*
System Software *16*
PROTOCOL DESCRIPTION 17
HOW THE EXPLOIT WORKS 19
DESCRIPTION AND DIAGRAM OF THE ATTACK 21
SIGNATURE OF THE ATTACK 27
HOW TO PROTECT AGAINST IT 29
Trend Micro InterScan VirusWall NT *29*
Lotus Domino *30*
Protecting Against It In Our Exploit *31*

SECTION 3: THE INCIDENT HANDLING PROCESS 32

PREPARATION 32
IDENTIFICATION 34
CONTAINMENT 36
ERADICATION 39
RECOVERY 40
LESSONS LEARNED 41
What Went Well *41*
What We Can Improve On *42*

REFERENCES 44

Executive Summary

This paper describes an actual incident handled by our improvised incident handling team. As with many organizations, we are in process of formalizing our incident handling processes and procedures. At the time of this incident, our drafted incident handling processes and procedures were under review by senior management, but had not yet been finalized or signed-off. Unfortunately, incidents do not wait until you have finished your preparation, but we leveraged the experience gained and funneled it back into our preparation efforts.

The attack was a combination of exploits and configurations that resulted in a moderate denial of service attack on our SMTP server by an infinite email loop. The denial of service attack was moderate because the infinite email loop was a single message sent back and forth between two servers, and the latency between these two disconnected processes prevented a much more sever attack.

Finally, it is important to note that the attack was triggered by a piece of spam email. The assumed intention of the sender was to sell a product, as linked in the email, and not to create a denial of service specifically on our mail servers. Ironically, the email caught much more attention than the sender would have expected.

Section 1: The Exploit

The exploit was related to two distinct vulnerabilities: Lotus Domino Mail Loop Denial of Service Vulnerability and Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability. Below is a brief overview of each of these vulnerabilities:

Lotus Domino Mail Loop Denial of Service Vulnerability

Name

Lotus Domino Mail Loop Denial of Service Vulnerability

CVE Candidate

CAN-2000-1203 (under review)

BID

3212

Operating Systems

This is an application vulnerability that is operating system independent. Therefore, all operating systems supporting the versions of the Lotus Domino Server listed in the proceeding application section are vulnerable. This would include the following operating systems:

- OS/2 Warp Server Version 4
- OS/2 Warp Server Advanced/SMP Version 4
- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows NT version 3.51
- Microsoft Windows NT version 4.0
- Microsoft Windows 2000
- IBM AIX version 4.1.x
- IBM AIX version 4.2.x
- IBM AIX version 4.3.x
- HP-UX 10.x
- HP-UX 11.x
- Sun Solaris 2.51
- Sun Solaris 2.6
- Sun Solaris 7
- Sun Solaris 8
- Novell NetWare 3.12
- Novell NetWare 4.x
- OS/400 V4R2 or Later
- OS/390 Version 2 Release 4 or Later
- Red Hat Linux 6.0

For further information on operating systems supported by Lotus Domino Server please see Lotus's web site at: http://www.lotus.com.

Application

The following versions of Lotus Domino Server are vulnerable:

- Lotus, Domino, 4.6
- Lotus, Domino, 4.6.1
- Lotus, Domino, 4.6.2
- Lotus, Domino, 4.6.3
- Lotus, Domino, 4.6.4
- Lotus, Domino, 4.6.5

- Lotus, Domino, 4.6.6
- Lotus, Domino, 4.6.7
- Lotus, Domino, 5.0
- Lotus, Domino, 5.0.1
- Lotus, Domino, 5.0.2
- Lotus, Domino, 5.0.3
- Lotus, Domino, 5.0.4
- Lotus, Domino, 5.0.5
- Lotus, Domino, 5.0.6
- Lotus, Domino, 5.0.7
- Lotus, Domino, 5.0.8

Protocol

This vulnerability uses the Simple Mail Transport Protocol (SMTP). SMTP is generally referred to as the Internet's standard protocol for mail transport. SMTP is a two-way host-to-host transmission channel between a SMTP client and SMTP server. The client, having a message to send, initiates communications and is responsible for transferring mail messages to the appropriate SMTP server or report failure if unable to do so. For more information about SMTP, please see the Protocol Description section of this document.

In relation to this vulnerability, there are a few specific aspects of SMTP that need to be highlighted:

1. SMTP is designed to notify the sender if an email cannot be delivered. Once a SMTP client has taken responsibility for an email message, it must either transfer responsibility to the appropriate SMTP server or notify the message sender that it was unable to deliver the message (unless it was addressed to a null address). This is highlighted in section 6.1 Reliable Delivery and Replies by Email of the IETF RFC 2821 where it states:

   "When the receiver-SMTP accepts a piece of mail (by sending a "250 OK" message in response to DATA), it is accepting responsibility for delivering or relaying the message. It must take this responsibility seriously. It MUST NOT lose the message for frivolous reasons, such as because the host later crashes or because of a predictable resource shortage.

   If there is a delivery failure after acceptance of a message, the receiver-SMTP MUST formulate and mail a notification message. This notification MUST be sent using a null ("<>") reverse path in the envelope. The recipient of this notification MUST be the address from the envelope return

path (or the Return-Path: line). However, if this address is null ("<>"), the receiver-SMTP MUST NOT send a notification."[1]

As will be illustrated in greater detail in proceeding sections, this vulnerability relies on the implementation and activation of delivery failure notification as described in IETF RFC 2821.

2. It is important to note that IETF RFC 2821, in Section 6.2 Loop Detection, requires that *"servers MUST contain provisions for detecting and stopping trivial loops"*[2]. SMTP does not inherently detect loops and relies on the SMTP implementations to take appropriate steps to detect and stop trivial loops. This vulnerability, as described in more detail in proceeding sections, illustrates a system that did not have the ability to detect this specific loop. It is worth noting that although SMTP does not inherently detect loops, it does define some requirements to help prevent loops, for instance null return addresses on notification messages (ensuring a SMTP server does not reply to a notification message)[3].

3. Third, the SMTP protocol is not designed to provide strong authentication services, therefore allowing spoofing to be easily done. IETF RFC 2821 states that:

   "SMTP mail is inherently insecure in that it is feasible for even fairly casual users to negotiate directly with receiving and relaying SMTP servers and create messages that will trick a naive recipient into believing that they came from somewhere else. Constructing such a message so that the "spoofed" behavior cannot be detected by an expert is somewhat more difficult, but not sufficiently so as to be a deterrent to someone who is determined and knowledgeable. Consequently, as knowledge of Internet mail increases, so does the knowledge that SMTP mail inherently cannot be authenticated, or integrity checks provided, at the transport level. Real mail security lies only in end-to-end methods involving the message bodies, such as those which use digital signatures."[4]

   Although there are digital signatures, as well as SMTP protocol extensions and configurations that do offer potential authentication services, SMTP does not inherently provide authentication. The ease of spoofing SMTP

[1] IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)
[2] IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)
[3] For more information see IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", "Section 4.5.5 Messages With a Null Reverse-Path", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)
[4] IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", "Section 4.5.5 Messages With a Null Reverse-Path", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)

email, not using digital signatures or authentication protocol extensions and configurations, is a key aspect of the vulnerability and represents what is most typical currently on the Internet. It is worth noting that SMTP does define commands like VRFY and EXPN to validate an address before attempting delivery, but this does not represent authentication or prevent spoofing, this just provides validation that the address is real. Unfortunately, VRFY and EXPN requests are commonly denied by SMTP servers to prevent spammers from acquiring valid email addresses through automated querying.

Brief Description

The Lotus Domino Mail Loop Denial of Service Vulnerability relies on the ability of an attacker to craft an email that has a MAIL FROM field that resolves to the localhost, also referred to as the internal host loopback, and a RCPT TO field that is not found on the target local domain.

In an SMTP message, the MAIL FROM is used to set the envelope return path. In other words, it provides the return or sender's address. For this vulnerability to be successful, the MAIL FROM must resolve to the localhost (127.0.0.1 – 127.255.255.255).

In an SMTP message, the RCPT TO is the list of envelope recipient addresses. In other words, it provides the addresses of the intended message receivers. For this vulnerability to be successful, the RCPT TO must include at least one address that does not exist not exist on the target local domain.

If Lotus Domino Server is sent a message with a MAIL FROM field that resolves to the internal host loopback and a RCPT TO field that includes at least one address that does not exist on its domain, Lotus Domino will attempt to bounce the message (send a delivery failure notification) and, because the MAIL FROM address resolves to the internal host loopback, the message is sent to itself. Lotus Domino Server, after accepting the bounced message, will determine that the email is for an external resource and re-send the message (to itself) causing an infinite loop. Lotus Domino Server does not detect this loop.

This attack will cause a Denial of Service by consuming 100% of the CPU resources on the Lotus Domino Server[5]. With the processor consumed by this looping message, other email messages are prevented from being accepted or sent, therefore denying mail service. Furthermore, this will cause the

[5] IBM, "Domino R5 SMTP "Denial of Service" Attack Caused by Routing Loop", Technote 191746, http://www-1.ibm.com/support/docview.wss?rs=0&org=sims&doc=DA18AA221C3B982085256B84000033EB , (Dec 8, 2002)

console to become unresponsive requiring the application to be stopped before the offending email can be removed from the MAIL.BOX file.

Any SMTP client capable of opening a connection and transferring a message to the target SMTP server, either directly or indirectly through an intermediate SMTP server can perform this attack.

Variants

There are a number of similar SMTP exploits causing infinite loop Denial of Service attacks. Here are a couple examples:

1. As reported in CVE-2000-0738, an email message with a RCPT TO address that has a period at the end will cause an infinite loop in WebShield SMTP 4.5.

   In this vulnerability, WebShield SMTP 4.5 misinterprets the period at the end of the RCPT TO address to be different then its local domain (the same address without the period at the end – mydomain.com compared to mydomain.com.). Believing that the address is not local because of the period at the end, WebShield SMTP 4.5 performs a DNS mail exchange record (MX) lookup on the address. The MX lookup will resolve this address to the WebShield SMTP 4.5 server (itself), as a period at the end is a valid expression of a fully qualified domain name (FQDN), and the message is sent to itself. In other words, WebShield SMTP 4.5 does not realize that mydomain.com and mydomain.com. are equivalent. This loop will continue until the consumption of CPU resources causes the application to crash. For more information on how this vulnerability works, please see SecurityFocus, "Network Associates WebShield SMTP Trailing Period DoS Vulnerability", BID 1589, Aug 18, 2000, http://online.securityfocus.com/bid/1589/info/, (January 14, 2002).

   This exploit is similar to the Lotus Domino Server vulnerability in that the loop will not be detected or stopped by the application, the message can be sent from an external attacker, and the result will be an infinite email loop Denial of Service by consuming CPU resources. Though the outcome is similar, they differ on the cause of the loop, as illustrated above.

2. As reported in CAN-2002-1005 (under review), ArGoSoft Mail Server 1.8.1.7 and prior versions is vulnerable to an infinite loop Denial of Service.

   In this vulnerability, if an authenticated user configures their email account to automatically forward messages to their own account and if they configure an auto-response to emails received, an infinite mail loop will be created resulting in a Denial of Service. ArGoSoft Mail Server 1.8.1.7 and

prior versions are unable to detect and stop this loop. For more information on how this vulnerability works, please see Security Tracker, "ArGoSoft Mail Server Lets Remote Authenticated Users Configure an Endless Loop to Cause Denial of Service Conditions", SecurityTracker AlertID:1004951, http://securitytracker.com/alerts/2002/Aug/1004951.html, (January 14, 2002).

This exploit is similar to the Lotus Domino Server vulnerability in that the loop will not be detected or stopped by the application and the result will be an infinite email loop Denial of Service by consuming CPU resources. It differs in the cause of the loop and the necessity of it to be started by an authenticated users account configuration. Furthermore, the ArGoSoft Mail Server vulnerability would require multiple messages before brining the CPU utilization to 100% and causing the administration console to become unresponsive, where as the Lotus Domino Server vulnerability can jump CPU utilization to 100% from a single email.

References

CVE, "Lotus Domino SMTP server 4.63 through 5.08 allows remote attackers to cause a denial of service (CPU consumption) by forging an email message with the sender as bounce@[127.0.0.1] (localhost), which causes Domino to enter a mail loop." , CVE CAN-2000-1203 (under review), http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-1203, (December 8, 2002).

CVE, "ArGoSoft Mail Server 1.8.1.7 and earlier allows a webmail user to cause a denial of service (CPU consumption) by forwarding the email to the user while autoresponse is enabled, which creates an inifinite loop", CAN-2002-1005 (under review), http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1005, (January 14, 2002)

CVE, "WebShield SMTP 4.5 allows remote attackers to cause a denial of service by sending e-mail with a From: address that has a . (period) at the end, which causes WebShield to continuously send itself copies of the e-mail", CVE-2000-0738, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0738, (January 14, 2002)

IBM, "Domino R5 SMTP "Denial of Service" Attack Caused by Routing Loop", Technote 191746, http://www-1.ibm.com/support/docview.wss?rs=0&org=sims&doc=DA18AA221C3B982085256B84000033EB, (Dec 8, 2002)

IBM, "Domino R5 SMTP Message Is Transferred Repeatedly to 127.0.0.1", Technote 188265, http://www-1.ibm.com/support/docview.wss?rs=0&uid=sim138d6a9cfd3bc072086256abf0075dc73, (Dec 8, 2002)

MARC, “Infinite loop in LOTUS NOTE 5.0.3. SMTP SERVER”, May 20, 2000, http://marc.theaimsgroup.com/?l=vuln-dev&m=95886062521327&w=2, (Dec 8, 2002)

SecurityFocus, “Network Associates WebShield SMTP Trailing Period DoS Vulnerability”, BID 1589, Aug 18, 2000, http://online.securityfocus.com/bid/1589/info/, (January 14, 2002)

SecurityFocus, “Lotus Domino Mail Loop Denial of Service Vulnerability”, BID 3212, Aug 20, 2002 http://online.securityfocus.com/bid/3212, (Dec 8, 2002)

Security Tracker, “ArGoSoft Mail Server Lets Remote Authenticated Users Configure an Endless Loop to Cause Denial of Service Conditions”, SecurityTracker AlertID: 1004951, http://securitytracker.com/alerts/2002/Aug/1004951.html, (January 14, 2002)

Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability

Name

No Official Name, Not in CVE
(For the purposes of this Paper: “Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability”)

TrendMicro Knowledge Base

Solution 10525

Operating Systems

This is an application vulnerability that affects all operating systems that support the versions of Trend Micro InterScan VirusWall NT listed in the proceeding application section. This would include the following operating systems:

- Microsoft Windows NT 4.0
- Microsoft Windows 2000

Application

All versions of Trend Micro InterScan VirusWall NT are vulnerable including all currently supported versions (Trend Micro, “Versions Supported”, http://www.trendmicro.com/en/support/versions/overview.htm, (Dec 8, 2002)):

- Trend Micro InterScan VirusWall NT 3.0x
- Trend Micro InterScan VirusWall NT 3.32
- Trend Micro InterScan VirusWall NT 3.4
- Trend Micro InterScan VirusWall NT 3.5
- Trend Micro InterScan VirusWall NT 3.51
- Trend Micro InterScan VirusWall NT 3.52
- Trend Micro InterScan VirusWall NT 3.53

Protocol

This vulnerability uses the Simple Mail Transport Protocol (SMTP). SMTP is generally referred to as the Internet's standard protocol for mail transport. SMTP is a two-way host-to-host transmission channel between a SMTP client and SMTP server. The client, having a message to send, initiates communications and is responsible for transferring mail messages to the appropriate SMTP server or report failure if unable to do so. For more information about SMTP, please see the Protocol Description section of this document.

In relation to this vulnerability, there are a few specific aspects of SMTP that need to be highlighted:

1. It is important to note that IETF RFC 2821, in Section 6.2 Loop Detection, requires that *"servers MUST contain provisions for detecting and stopping trivial loops"*[6]. SMTP does not inherently detect loops and relies on the SMTP implementations to take appropriate steps to detect and stop trivial loops. This vulnerability, as described in more detail in proceeding sections, illustrates a system that did not have the ability to detect this specific loop. It is worth noting that although SMTP does not inherently detect loops, it does define some requirements to help prevent loops, for instance null return addresses on notification messages (ensuring a SMTP server does not reply to a notification message)[7].

2. Second, the SMTP protocol is not designed to provide strong authentication services, therefore allowing spoofing to be easily done. IETF RFC 2821 states that:

   "SMTP mail is inherently insecure in that it is feasible for even fairly casual users to negotiate directly with receiving and relaying SMTP servers and create messages that will trick a naive recipient into believing that they came from somewhere else. Constructing such a message so that the

[6] IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)
[7] For more information see IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", "Section 4.5.5 Messages With a Null Reverse-Path", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)

"spoofed" behavior cannot be detected by an expert is somewhat more difficult, but not sufficiently so as to be a deterrent to someone who is determined and knowledgeable. Consequently, as knowledge of Internet mail increases, so does the knowledge that SMTP mail inherently cannot be authenticated, or integrity checks provided, at the transport level. Real mail security lies only in end-to-end methods involving the message bodies, such as those which use digital signatures."[8]

Although there are digital signatures, as well as SMTP protocol extensions and configurations that do offer potential authentication services, SMTP does not inherently provide authentication. The ease of spoofing SMTP email, not using digital signatures or authentication protocol extensions and configurations, is a key aspect of the vulnerability and represents what is most typical currently on the Internet. It is worth noting that SMTP does define commands like VRFY and EXPN to validate an address before attempting delivery, but this does not represent authentication or prevent spoofing, this just provides validation that the address is real. Unfortunately, VRFY and EXPN requests are commonly denied by SMTP servers to prevent spammers from acquiring valid email addresses through automated querying.

Brief Description

The Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability relies on the ability of an attacker to craft an email that has a RCPT TO field that resolves to the localhost, also referred to as the internal host loopback, and have this email accepted by the Trend Micro InterScan VirusWall NT Server.

In an SMTP message, the RCPT TO is the list of envelope recipient addresses. In other words, it provides the addresses of the intended message receivers. For this vulnerability to be successful, the RCPT TO must include at least one address that will resolve to the internal host loopback.

If Trend Micro InterScan VirusWall NT is sent a message with a RCPT TO field that resolves to the internal host loopback, it will continually send the message to itself, causing an infinite loop. Trend Micro InterScan VirusWall NT does not detect this loop.

This attack will cause a Denial of Service by consuming CPU resources on the Trend Micro InterScan VirusWall NT Server. One single email will not consume all CPU resources, but multiple emails can create a full Denial of Service by consuming 100% of CPU resources. With the processor consumed by this looping message, other email messages are prevented from being accepted or sent, therefore denying mail service.

[8] IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", "Section 4.5.5 Messages With a Null Reverse-Path", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)

Any SMTP client capable of opening a connection and transferring a message to the target SMTP server, either directly or indirectly through an intermediate SMTP server can perform this attack.

Variants

There are a number of similar SMTP exploits causing infinite loop Denial of Service attacks. Here are a couple examples:

1. As reported in CVE-2000-0738, an email message with a RCPT TO address that has a period at the end will cause an infinite loop in WebShield SMTP 4.5.

   In this vulnerability, WebShield SMTP 4.5 misinterprets the period at the end of the RCPT TO address to be different then its local domain (the same address without the period at the end – mydomain.com compared to mydomain.com.). Believing that the address is not local because of the period at the end, WebShield SMTP 4.5 performs a DNS MX lookup on the address. The MX lookup will resolve this address to the WebShield SMTP 4.5 server (itself), as a period at the end is a valid expression of a fully qualified domain name (FQDN), and the message is sent to itself. In other words, WebShield SMTP 4.5 does not realize that mydomain.com and mydomain.com. are equivalent. This loop will continue until the consumption of CPU resources causes the application to crash. For more information on how this vulnerability works, please see SecurityFocus, "Network Associates WebShield SMTP Trailing Period DoS Vulnerability", BID 1589, Aug 18, 2000, http://online.securityfocus.com/bid/1589/info/, (January 14, 2002).

   This exploit is similar to the Trend Micro InterScan VirusWall NT vulnerability in that the loop will not be detected or stopped by the application, the message can be sent from an external attacker (as an example, even with anti-relaying, an email that causes a delivery failure notification will still work), and the result will be an infinite email loop Denial of Service by consuming CPU resources. Though the outcome is similar, they differ on the cause of the loop.

2. As reported in CAN-2002-1005 (under review), ArGoSoft Mail Server 1.8.1.7 and prior versions is vulnerable to an infinite loop Denial of Service.

   In this vulnerability, if an authenticated user configures their email account to automatically forward messages to their own account and if they configure an auto-response to emails received, an infinite mail loop will be created resulting in a Denial of Service. ArGoSoft Mail Server 1.8.1.7 and

prior versions are unable to detect and stop this loop. For more information on how this vulnerability works, please see Security Tracker, "ArGoSoft Mail Server Lets Remote Authenticated Users Configure an Endless Loop to Cause Denial of Service Conditions", SecurityTracker AlertID:1004951, http://securitytracker.com/alerts/2002/Aug/1004951.html, (January 14, 2002).

This exploit is similar to the Trend Micro InterScan VirusWall NT vulnerability in that the loop will not be detected or stopped by the application, the result will be an infinite email loop Denial of Service by consuming CPU resources, and multiple messages would be required before bringing the CPU utilization to 100%. Though the outcome is similar, they differ on the cause of the loop and the requirement of an authenticated user.

References

CVE, "ArGoSoft Mail Server 1.8.1.7 and earlier allows a webmail user to cause a denial of service (CPU consumption) by forwarding the email to the user while autoresponse is enabled, which creates an inifinite loop", CAN-2002-1005 (under review), http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1005, (January 14, 2002)

CVE, "WebShield SMTP 4.5 allows remote attackers to cause a denial of service by sending e-mail with a From: address that has a . (period) at the end, which causes WebShield to continuously send itself copies of the e-mail", CVE-2000-0738, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0738, (January 14, 2002)

E-Secure-DB, "Antivirus - Trend Micro", Jan 21, 2002, http://www.e-secure-db.us/dscgi/ds.py/View/Collection-51, (Dec 9, 2002)

SecurityFocus, "Network Associates WebShield SMTP Trailing Period DoS Vulnerability", BID 1589, Aug 18, 2000, http://online.securityfocus.com/bid/1589/info/, (January 14, 2002)

Security Tracker, "ArGoSoft Mail Server Lets Remote Authenticated Users Configure an Endless Loop to Cause Denial of Service Conditions", SecurityTracker AlertID: 1004951, http://securitytracker.com/alerts/2002/Aug/1004951.html, (January 14, 2002)

Trend Micro, "Solution 10525", Mar 1, 2002, http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=10525, (Dec 8, 2002)

The Combination

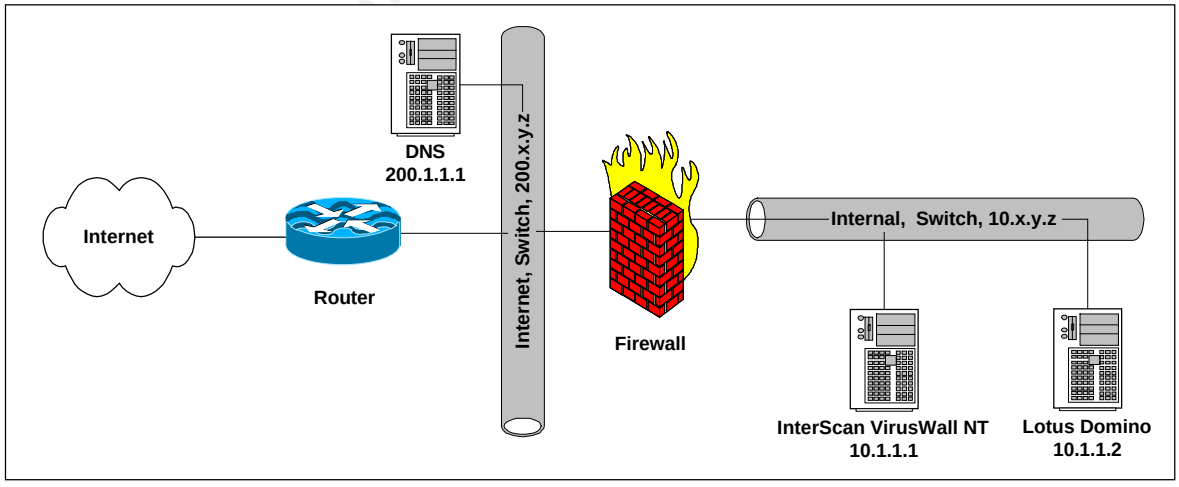
Although the Lotus Domino Mail Loop Denial of Service Vulnerability and the Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability form the basis of the exploit in this paper, it is important to note that the actual exploit is a pseudio combination of both vulnerabilities. Though explained in more detail in the preceding sections, the attack relied on the inability of InterScan VirusWall NT to handle a RCPT TO that resolved to the internal host loopback, the inability of both InterScan VirusWall NT and Lotus Domino to detect the loop that had formed from an attempted delivery failure notification, and the configurations of the two servers – InterScan VirusWall NT was the relay between our internal mail system (Lotus Domino) and the Internet for both inbound and outbound SMTP messages.

Section 2: The Attack

This section will explain how the exploit was used to cause a moderate SMTP Denial of Service attack.

Description and Diagram of Network

The network was configured as illustrated in the figure below. Both the description and diagram will only include systems information required to illustrate the flow of SMTP traffic specific to this paper. Additional details concerning systems are purposefully excluded.



Network SMTP Traffic Flow

External inbound SMTP traffic would first cross the network edge router onto the company's Internet segment (200.x.y.z for the purposes of this paper). From the Internet segment, SMTP traffic was routed, through Network Address Translation (NAT) on the firewall, to the SMTP relay server (InterScan VirusWall NT) on the Internal segment (10.x.y.z for the purposes of this paper). InterScan VirusWall was configured to forward all valid SMTP messages (anti-relaying was configured to accept only messages destined for the local domain) to the Lotus Domino server.

To facilitate the acceptance of SMTP email from the Internet, both the router and the firewall network devices were configured to allow SMTP traffic only between our SMTP server (InterScan VirusWall NT) and the Internet. Furthermore, there were no specific IP exclusions, so anyone on the Internet was permitted to send an email to our SMTP server over inbound port 25 (as well as the network devices allowing the resulting outbound high ports (>1023)). In the reverse, only our SMTP server was permitted to send an email to any SMTP server on the Internet over outbound port 25 (as well as the network devices allowing the resulting inbound high ports (>1023)). The router performed static packet filtering, where as the firewall performed stateful packet filtering in regards to these rules. Neither of these network devices interrogate a packet above OSI Model Layer 4, meaning the only defense against SMTP with the current network device capabilities would be maintaining IP exclusions rules, which as previously mentioned, was not done on a network device.

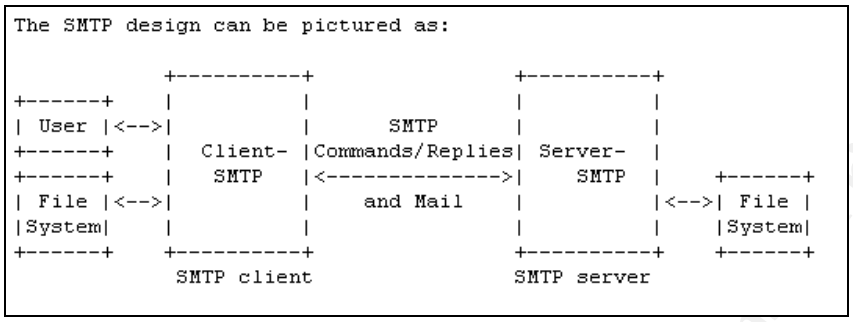
Internal outbound SMTP traffic was sent from the internal network through the Lotus Domino server illustrated above. The Lotus Domino server forwarded all outbound SMTP email to the InterScan VirusWall server. The InterScan VirusWall server resolved the destination through a DNS lookup (primary DNS server is illustrated above) and sent the SMTP traffic to the resolved IP address.

System Software

InterScan VirusWall NT version 3.53 was running on a Microsoft Windows 2000 Server with SP2 and the latest critical security patches. Lotus Domino 5.07 was running on Microsoft Windows 2000 with SP2 and the latest critical security patches. The DNS in the above diagram was the authoritative DNS server for the company's domain (mycompany.com for the purposes of this paper).

Protocol Description

Simple Mail Transport Protocol (SMTP) is generally referred to as the Internet's standard protocol for mail transport. IETF RFC 2821 diagrams the protocol as follows (IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)):



SMTP is a two-way host-to-host transmission channel between a SMTP client and SMTP server. The client, having a message to send, initiates communications and is responsible for transferring mail messages to the appropriate SMTP server or report failure if unable to do so. SMTP servers generally listen on port 25.

SMTP typically works as follows (for more detailed information and variations, please see IETF RFC 2821):

1. The SMTP client has a message that it wants to deliver. At this point, the SMTP client is responsible for the message and wants to either transfer responsibility to the appropriate SMTP server or notify the message sender that it was unable to deliver the message. The SMTP client could be the original sender or an intermediate (SMTP relay or gateway) between the originator and the final destination.

2. Once an SMTP client has a message to be delivered, it client resolves the destination domain. This could be performed by a DNS lookup of the destination SMTP server, or the SMTP client may be configured to forward messages to an intermediate SMTP server, as is common with isolated transport environments or SMTP clients using protocols such as IMAP.

3. Once the SMTP client has the SMTP server's IP address, the SMTP client initiates a connection to the SMTP server and awaits a response. The SMTP server will accept, temporarily refuse, or refuse the connection. Since the SMTP client is responsible for the message, if it is temporarily refused, it will retry connection. If the SMTP client is refused, it will notify the sender. If the connection is accepted, we move onto the next step.

4. After the connection is open, the SMTP client sends a *HELO* request (or *EHLO*) and awaits a reply. The SMTP server will accept, temporarily refuse, or refuse the request. Since the SMTP client is still responsible for the message, if it is temporarily refused, it will retry the request. If the SMTP client is refused, it will notify the sender. If the request is accepted, we move onto the next step.

5. After the HELO request, the SMTP client sends a MAIL request showing the sender's address and waits for a response. If it is temporarily refused, it will retry the request. If the SMTP client is refused, it will notify the sender. If the request is accepted, we move onto the next step.

6. After the MAIL request, the client sends one RCPT request for each message recipient address, waiting for a response after each request. As long as at least one of the RCPT addresses is accepted, we move onto the next step. For each temporarily refused response, the SMTP client retires; for each refused response, the SMTP client notifies the sender.

7. Next, the SMTP client sends a DATA request and awaits a response. If accepted, the SMTP client then sends the encoded message and waits for a response. If the message is accepted, the SMTP client quits (QUIT) and it has successfully transferred responsibility to the SMTP server. For each temporarily refused response, the SMTP client retires; for each refused response, the SMTP client notifies the sender.

An example of a typical SMTP transaction is provided by IETF RFC 2821 (IETF RFC 2821, J. Klensin, Editor, "Simple Mail Transfer Protocol", April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821, (Dec 8, 2002)):

D.1 A Typical SMTP Transaction Scenario

This SMTP example shows mail sent by Smith at host bar.com, to Jones, Green, and Brown at host foo.com. Here we assume that host bar.com contacts host foo.com directly. The mail is accepted for Jones and Brown. Green does not have a mailbox at host foo.com.

S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA

```
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

InterScan VirusWall NT and Lotus Domino are two products that implement SMTP.

How The Exploit Works

This exploit relies on the ability to craft, send, and transfer responsibility of an email message that has a MAIL FROM field that resolves to 127.0.0.1 – 127.255.255.255 (the internal host loopback) to a recipient (RCPT TO) that does not exist on the target local domain and exploit an implementation of SMTP that does not correctly handle this condition when it attempts to notify the sender of a delivery failure.

The first step required for this exploit to be successful is to ensure the SMTP client will resolve the SMTP server to the internal host loopback or if the crafted email has a MAIL FROM field that is an internal host loopback address (e.g. badguy@127.0.0.1), in which case it is already resolved.

There are many methods that an attacker can use to cause this DNS resolution. The simplest is to put an internal host loopback address directly in the MAIL FROM field as above - badguy@127.0.0.1. Other options would include intentionally or unintentionally configuring a DNS MX record on the name server (NS) to resolve to an internal host loopback address or exploiting DNS Spoofing/Cache Poisoning vulnerabilities (an example of this would be CVE-1999-0024, “DNS cache poisoning via BIND, by predictable query IDs”, at http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0024).

For more information on DNS exploits, vulnerabilities, and configurations, please refer to the following resources:

> András Salamon, “DNS related RFCs”, http://www.dns.net/dnsrd/rfc/, (Dec 11, 2002)
>
> Cricket Liu, “Securing an Internet Name Securing an Internet Name Server”, http://www.linuxsecurity.com/resource_files/server_security/securing_an_internet_name_server.pdf, (Jan 14, 2003)

Doug Sax, “DNS Spoofing (Malicious Cache Poisoning)”, November 12, 2000 http://rr.sans.org/firewall/DNS_spoof.php, (Dec 11, 2002)

Jason Coombs, “The Large-Scale Threat of Bad Data in DNS”, August 14, 2002, http://www.linuxsecurity.net/articles/network_security_article-5514.html, (Dec 11, 2002)

In the actual attack on our servers, the name server (NS) for the domain had a MX record resolving to the internal host loopback. Unfortunately, we could not determine if this was an intentional/unintentional configuration or the result of a compromise.

At this point, we will assume that the MAIL FROM field will resolve to the internal host loopback. We will also assume that the RCPT TO field will not be found on the local domain (the email recipient does not exist). These are assumed to keep the focus of this paper on the SMTP exploit of this paper. So an email is crafted having the following attributes:

```
mail from: badguy@bogus.com <mailto:mailto:badguy@bogus.com>
rcpt to: nobody@mycompany.com <mailto:nobody@mycompany.com>
```

Where *bogus.com* (used only in this paper to represent a domain configured by an attacker) will have an MX record that resolves to the internal host loopback and *nobody@mycompany.com* is not an account on the local domain.

The next step is to send the message to the InterScan VirusWall NT SMTP server. This could be done from any SMTP client. As with the DNS methods, the exploit is not dependent on a specific SMTP client; it is only dependent on a SMTP client capable of opening a connection and transferring a message to the target SMTP server, either directly or indirectly through an intermediate SMTP server. If you wanted to do this manually you type the following commands in a telnet session:

```
telnet smtp.mycompany.com 25
helo bogus.com
mail from: badguy@bogus.com
rcpt to: nobody@MyCompany.com
data
blah,blah
.
quit
```

When the InterScan VirusWall NT SMTP server receives this message, it is configured to forward the message to the Lotus Domino server. This is a typical configuration for InterScan VirusWall NT servers that are acting as virus-scanning mail relays. It is important to note that InterScan VirusWall NT has no native ability to filter the MAIL FROM field. This literally means that as long as the message is destined for the local domain (typical anti-relay configurations

checking RCPT TO), InterScan VirusWall NT will forward the message. Furthermore, because InterScan VirusWall NT is configured to be a relay, it will accept messages and assume responsibility even if the RCPT TO field references an account that does not exist.

Once InterScan VirusWall NT receives the message, it forwards it to the Lotus Domino server. Lotus Domino is configured to accept inbound messages from the InterScan VirusWall NT server and, by default, places them in the Inbound Work Queue still in SMTP format. This means that Lotus Domino has assumed responsibility for the delivery of the message. Then, the Inbound Message Conversion task attempts to convert the message and the destination user address into Notes format. If the message cannot be converted or if the address is not deliverable, it will indicate that delivery has failed. A delivery failure notification will be generated.

After the delivery failure notification message is generated, Lotus Domino proceeds with how it routes all messages it is responsible for. In this case, it determines that the message is not on the local domain, and forwards the message to the configured outbound SMTP server (InterScan VirusWall NT).

When InterScan VirusWall NT receives the notification message, it processes the email message as any other outbound message. First, it resolves the message's destination SMTP server through a DNS query, which in this case is its internal host loopback, and then forwards the message to this resolved address. In other words, InterScan VirusWall NT sends the message to itself. Once InterScan VirusWall NT accepts the message from itself, it does as it is configured to do and forwards the message to Lotus Domino. Of course, if InterScan VirusWall NT was not configured to forward messages, this would of caused a Denial of Service on just the InterScan VirusWall NT server as highlighted earlier in this document: *Trend Micro InterScan VirusWall NT Infinite Loop Vulnerability.*

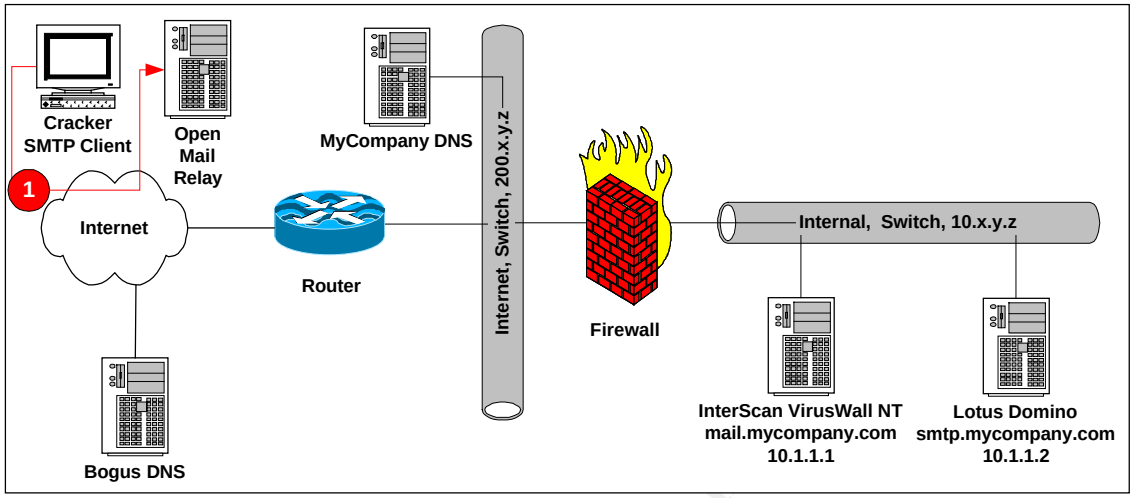
Finally, we see the SMTP loop form. InterScan VirusWall NT forwards the message to Lotus Domino. Lotus Domino accepts the message, realizes the message is for an external domain, and forwards the message to the outbound SMTP server (InterScan VirusWall NT). InterScan receives the message, sends it to its internal host loopback, and the infinite SMTP loop is created.

It is important to note that neither InterScan VirusWall NT of Lotus Domino were able to handle a message with a RCPT TO field that resolves to the internal host loopback (even though Lotus Domino, in this scenario, was not performing DNS resolution for mail relaying) and neither system detected this loop.

Description and Diagram of the Attack

The following diagrams and descriptions illustrate how the attack happened:

Step #1

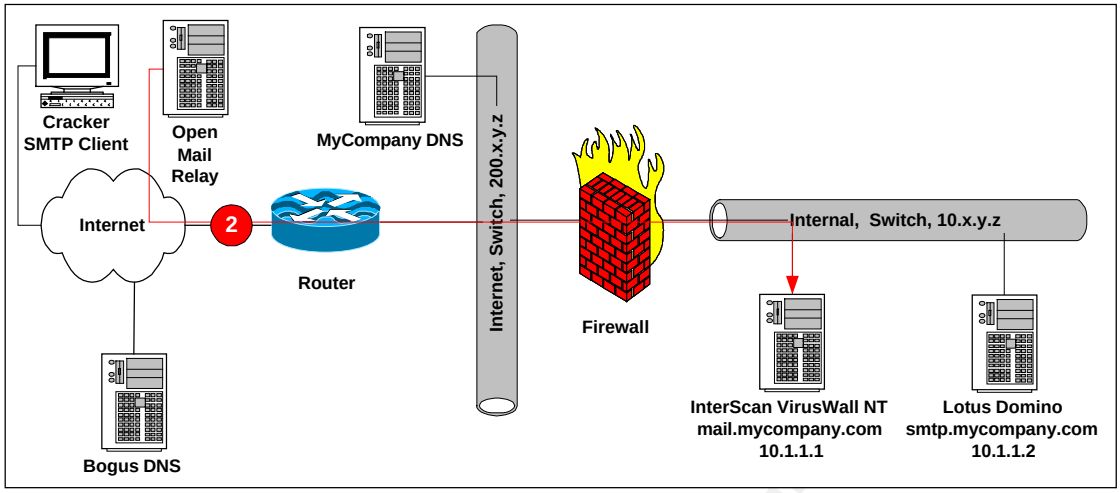


An email message was sent from a cracker (an alternate description is spammer in this case) using an SMTP client to an open mail relay (a mail relay that will forward packets from any SMTP client going to any SMTP server). The identity of the mail relay is intentionally withheld, though it is worth noting that it was not found in the Open Relay DataBase (www.ordb.org). The relay was verified using Truson Technologies Spam Tester (http://www.trusontechnologies.com/services/spam_tester.php). The email message was crafted with the following attributes (altered for security reasons):

```
mail from: badguy@bogus.com <mailto:mailto:badguy@bogus.com>
rcpt to: nobody@mycompany.com <mailto:nobody@mycompany.com>
```

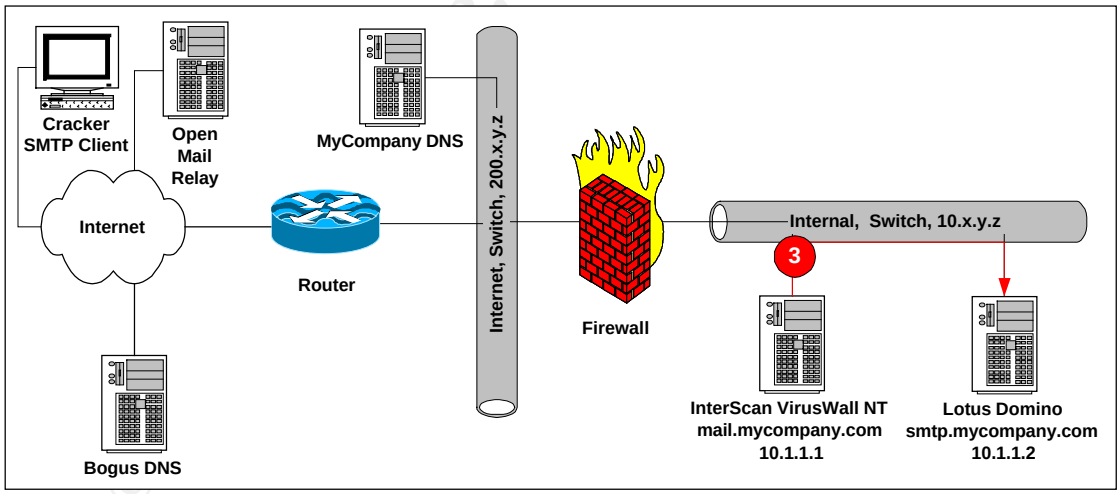
As mentioned previously, the email was an advertisement for a product, *bogus.com* was a fictional domain created solely for this paper that had a MX record that resolved to the local host loopback, and *nobody* was an employee who was no longer with the company and whose email had been removed.

Step #2



The open mail relay forwarded the message to InterScan VirusWall NT (mail.mycompany.com). Because the message's RCPT TO field matched mycompany.com (anti-relaying filters), InterScan VirusWall NT accepted the message. It is important to remember that InterScan VirusWall NT had no way to determine if the RCPT TO was a valid email address, but had still taken responsibility for the message.

Step #3



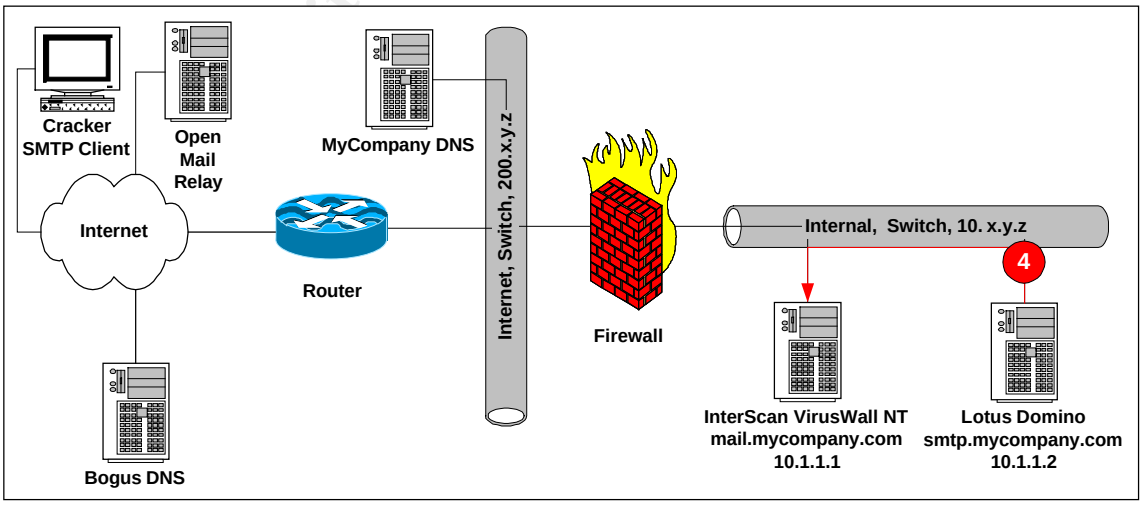
InterScan VirusWall NT forwarded the message to Lotus Domino (smtp.mycompany.com). The message was as follows (certain parts have been removed or altered for security reasons):

```
Received: from mail.mycompany.com ([10.1.1.1])
        by smtp.mycompany.com (Lotus Domino Release 5.0.7)
        with SMTP id 2002120401405849:228429 ;
        Wed, 4 Dec 2002 01:40:58 -0400
Received: from mail.bogus.com by mail.mycompany.com (InterScan E-Mail VirusWall
NT);
        Wed, 04 Dec 2002 01:43:08 -0400
Message-ID: <00004e303ea5$000023ce$00004119 >
To: <nobody@ mycompany.com>
From: "Badguy" <badguy@bogus.com>
Subject: Blah
Date: Mon, 02 Dec 2002 13:07:14 -1700
MIME-Version: 1.0
Reply-To: badguy@bogus.com
```

It is important to note that Lotus Domino accepted the inbound message (including responsibility) and, by default, placed it in the Inbound Work Queue still in SMTP format. Then, the Inbound Message Conversion task discovered that the address is not deliverable, and created a delivery failure notification. The delivery failure part is as follows:

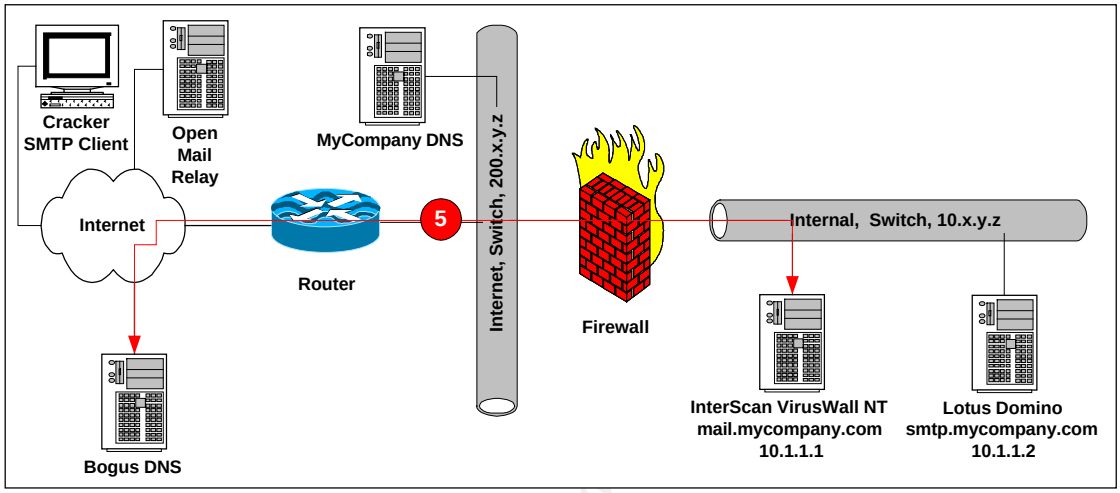
```
Content-Type: message/delivery-status
Reporting-MTA: dns; smtp.mycompany.com

Final-Recipient: rfc822; nobody@ mycompany.com
Action: failed
Status: 5.1.1
Diagnostic-Code: X-Notes; User nobody (nobody@ mycompany.com) not listed in public
Name & Address Book
```

Step #4



Lotus Domino sent the delivery failure notification to the outbound SMTP server (InterScan VirusWall NT) because the RCPT TO is external to the local domain.

Step #5



Upon receiving the outbound message, InterScan VirusWall NT performed a DNS lookup to determine the destination IP address of the SMTP server responsible for handling bogus.com email. To do this, InterScan VirusWall NT sent a DNS MX query to our DNS name server (MyCompany DNS), which resolved the mail exchanger for bogus.com and returned its IP address.

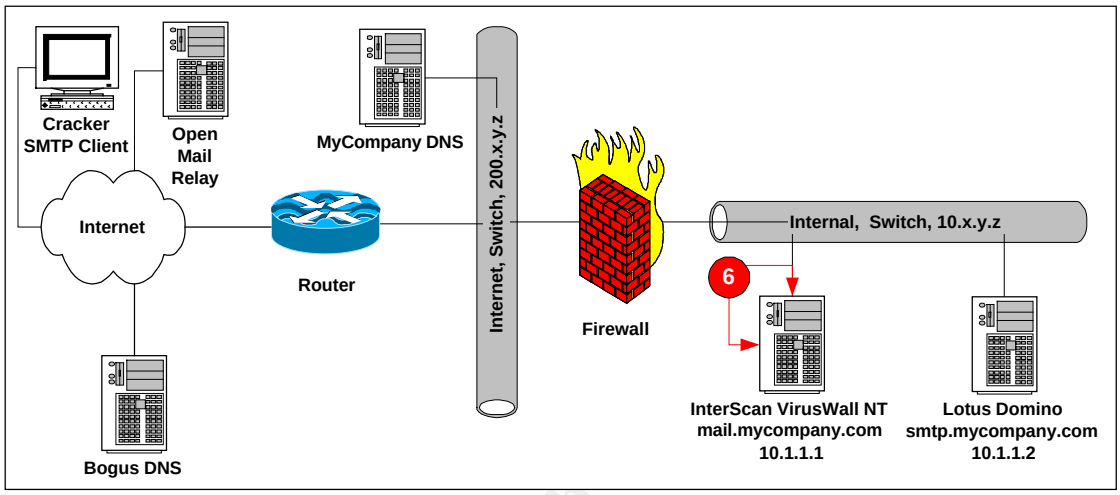
This followed typical DNS resolution where the DNS client (InterScan VirusWall NT) queried its DNS server (MyCompany DNS) for the mail exchange record for bogus.com. Given MyCompany DNS is not authoritative for bogus.com, and assuming the information was not in cache, the MyCompany DNS server would issue a recursive query. In this way, the MyCompany DNS server would ask a root name server for the IP address of a host that is authoritative for the bogus.com, and would then contact the authoritative server provided by the root (or provided by a downstream DNS server depending on the number of levels it needed to go through to obtain the authoritative server for bogus.com) and report back to the DNS client (InterScan VirusWall NT) the IP address of the mail exchange record.

The returned IP address was 127.0.0.1. The diagram is intended to illustrate where the authoritative DNS record was stored and not the full process of DNS resolution. An nslookup command or using a tool like Sam Spade (www.samspade.org) was used to verify this (the data has been altered for security reasons):

From www.samspade.org:

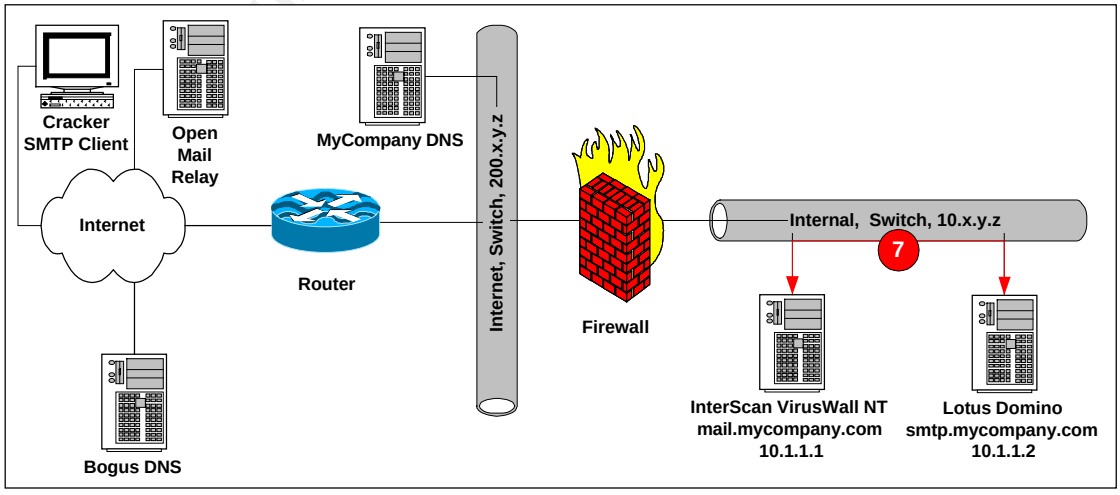
dns bogus.com
Mail for bogus.com is handled by mail.bogus.com (10) 127.0.0.1

Step #6



Once the destination was resolved, InterScan VirusWall NT contacted and sent the message to the resolved SMTP server – 127.0.0.1 (itself). The message was then forwarded to Lotus Domino because it was configured to forward all messages received to Lotus Domino. This is the vulnerability in InterScan VirusWall NT when a message has a RCPT TO that resolve to its internal host loopback.

Step #7



Then InterScan VirusWall NT forwarded the message to Lotus Domino. Lotus Domino accepts the message (as described in more detail previously), realizes the message is for an external domain, and forwards the message back to the outbound SMTP server (InterScan VirusWall NT), thus causing an infinite loop. Below is output from the Lotus Domino log file:

```
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:38 AM  SMTP Server: Message 001F472E received
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:38 AM  Router: Message 001F4671 transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:39 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
12/04/2002 01:41:40 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:40 AM  SMTP Server: Message 001F4805 received
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:40 AM  Router: Message 001F472E transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:41 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
12/04/2002 01:41:42 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
```

When examining the Lotus Domino log file it is important to notice (in bold) that the same message that is delivered to Lotus Domino is sent back out to the InterScan VirusWall NT server, illustrating the loop. This same looping behavior is seen in the corresponding InterScan VirusWall NT log file:

```
12/04/2002 01:41:37 Iscan-maild[254]: Message from: <>
12/04/2002 01:41:37 Iscan-maild[254]: Message to: badguy@bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
12/04/2002 01:41:39 Iscan-maild[196]: Message from: <>
12/04/2002 01:41:39 Iscan-maild[196]: Message to: badguy@bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
```

Signature of the Attack

There are many signatures to this attack that could be both detected and monitored. At the highest level, the signature of the attack is any MAIL FROM destination address that resolves to or is the internal host loopback. There is also

the repetitive nature of the email loop in both the logs and network traffic that could be used to detect and block the loop.

Given the nature of the attack, the options for detecting and blocking are numerous. From a detection standpoint, IDS systems could be configured to perform DNS resolution on MAIL FROM fields in inbound SMTP packets to detect this type of attack. A second example was illustrated in the Lotus Domino log file as illustrated below - the existence of re-occurring paired message IDs (in bold) would indicate a potential attack and serve as a signature that could be monitored by HP OpenView or similar tool to alert on this activity.

```
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:38 AM  SMTP Server: Message 001F472E received
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:38 AM  Router: Message 001F4671 transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:39 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
12/04/2002 01:41:40 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:40 AM  SMTP Server: Message 001F4805 received
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:40 AM  Router: Message 001F472E transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:41 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
12/04/2002 01:41:42 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
```

A third example is the similar pattern is observed in the InterScan VirusWall NT log file that could serve as a signature and be monitored as well. For this specific vulnerability, the repeating of the bolded sections (indicating both delivery failure and the same RCPT TO could be used.

```
12/04/2002 01:41:37 Iscan-maild[254]: Message from: <>
12/04/2002 01:41:37 Iscan-maild[254]: Message to: badguy@bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
12/04/2002 01:41:39 Iscan-maild[196]: Message from: <>
12/04/2002 01:41:39 Iscan-maild[196]: Message to: badguy@bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
```

Once detected, through any of these methods, a rule could be written in an IDS or Monitoring package (HP OpenView) that could write a rule on the firewall,

InterScan VirusWall NT or Lotus Domino to block messages from the offending IP address or a manual process of alerting the appropriate person.

The main issue with the detection and blocking methods above is that they are reactive. Instead of writing reactive rules, our approach was to evaluate and implement a preventative solution to this problem and similar problems through anti-spam software. In this way, we could immediately detect and block this type of email before it was processed by our mail systems. Examples of these include BlackHole, SpamFilter, as well as modules for InterScan VirusWall NT. This evaluation was underway at the time of this paper.

How to Protect Against It

Before considering how to protect against the exploit of this paper, we need to consider the options available for the vulnerability of each application.

Trend Micro InterScan VirusWall NT

According to Trend Micro, we can do one of the following two options to resolve the vulnerability in InterScan VirusWall NT (Trend Micro, "Solution 10525", Mar 1, 2002, http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=10525, (Dec 8, 2002)):

> *"Solution 1: InterScan VirusWall with InterScan eManager plug-in*
>
> > *Create an Anti-Spam rule to block any inbound and outbound mail going to "Domain.com" or similar domains that use a 127.0.0.1 IP address.*
>
> *Solution 2: InterScan VirusWall without InterScan eManager*
>
> > *Configure InterScan VirusWall to forward outbound mail to another SMTP server, which should properly handle the infinite looping effect for mail to be delivered to "Domain.com".*"

If you are someone who is running the vulnerable software and you are unable to expand your Trend Micro solution to include InterScan eManager, your only option offered by the vendor is to forward outbound mail to another SMTP server. There is currently no patch available to fix this vulnerability and this includes the latest version of InterScan VirusWall NT. Unfortunately, this is not fixing a known problem with an existing software solution.

If you do opt for the plug-in, InterScan eManager, you will soon discover that the plug-in depends on domain (text-based) filters, not IP filters. In other words, we would have to constantly be modifying the configuration file hoping to add a troublesome domain before we receive an email from it, instead of simply adding the resolved internal host loopback range. Not a very attractive alternative.

Lotus Domino

Lotus Domino, on the other hand, has fixed their vulnerability in releases after 5.0.8, so an upgrade will remove their vulnerability. The fix ensures that Lotus Domino detects and stops the loop that is caused. Lotus Domino does this by retaining SMTP received headers of the delivery status notification messages to be able to detect the loop. Alternatively, if you are not able to upgrade, Lotus provides the following workaround (IBM, "Domino R5 SMTP "Denial of Service" Attack Caused by Routing Loop", Technote 191746, http://www-1.ibm.com/support/docview.wss?rs=0&org=sims&doc=DA18AA221C3B982085256B84000033EB, (Dec 8, 2002)):

> "1. Add "[127.0.0.1]" (without the quotation marks) to the "Deny messages from the following internet addresses/domains" field (in the Server Configuration document's Router/SMTP, Restrictions and Controls, and SMTP Inbound Controls tabs, and Inbound Sender Controls section). This configuration causes the SMTP Listener task to reject MAIL FROM commands that contain the "[127.0.0.1]" IP address.
>
> 2. Configure the SMTP inbound relay restrictions. If applicable, add the wildcard character "*" to the "Deny messages from external internet domains to be sent to the following Internet domains" (in the Server Configuration document's Router/SMTP, Restrictions and Controls, and SMTP Inbound Controls tabs, and Inbound Relay Controls section). This configuration causes the SMTP Listener task to reject RCPT TO commands that contain external Internet domains, domains not configured through the "Fully qualified Internet host name field" in the Basics tab of the Server document or in the Conversions tab of the Global Domain document(s). This restriction includes the "[127.0.0.1]" IP address."

Although the workaround may seem sufficient, we run into the same problem as we found in InterScan VirusWall NT. The fields that we add these into are text fields that do exact matching. In our scenario, it is not a text field of 127.0.0.1 that we are trying to protect against, but a domain that resolves to the internal host loopback range, including 127.0.0.1, or at least the ability to detect an email loop and stop it. The workaround is not an attractive alternative either. For more information see:

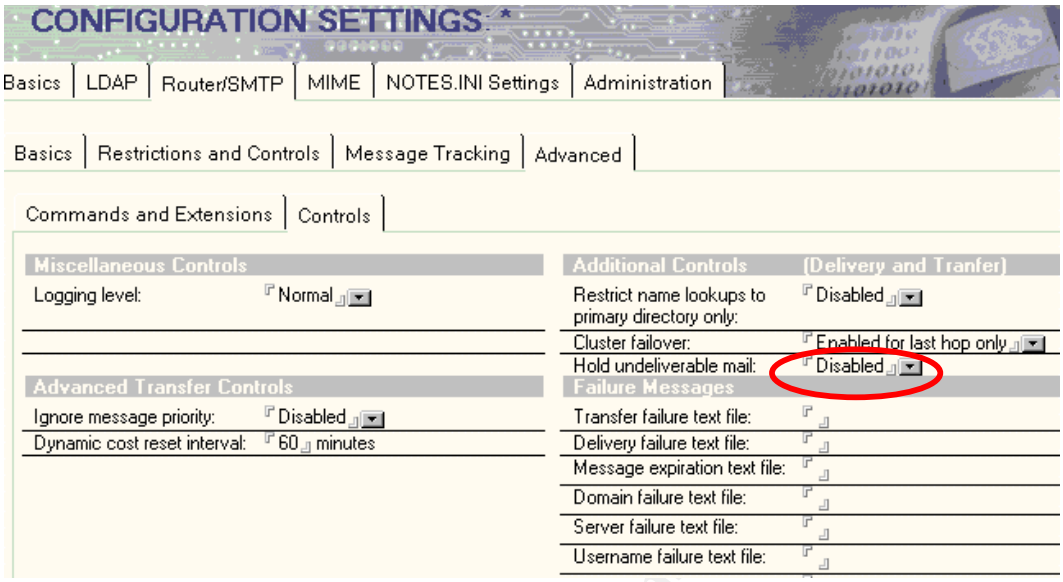
IBM, “Domino R5 Router Restrictions and Controls Explained”, http://www-1.ibm.com/support/docview.wss?rs=1&uid=sim1a786e37762de8341852568da004f9a80, (Dec 11, 2002)

Protecting Against It In Our Exploit

Two options for protecting against this exploit are:

1. Given the configuration of the network and the specific exploit, you can upgrade Lotus Domino to a version later than 5.0.8 and change your Lotus Domino server to become the outbound SMTP server. This will remove the vulnerability from InterScan VirusWall NT (as it should only be acting as an inbound forwarder and not a relay for any outbound messages) and, because Lotus Domino will detect the email loop, the message will be correctly handled on the Lotus Domino server. This option includes following the recommendations from both vendors. It is worth noting that by upgrading Lotus Domino, giving it the capability to detect the loop, you should not need to remove InterScan VirusWall NT as your outbound SMTP server. Theoretically, if a loop were started from the InterScan VirusWall NT vulnerability, Lotus Domino would detect and stop the loop because it retains the SMTP received headers of the delivery status notification messages.

2. If you are unable to upgrade, you can have the Lotus Domino SMTP router hold all messages with failures in the mail.box. You can do this in the Server Configuration Document by selecting the Router/SMTP tab, Advanced tab, then the Controls Section tab. You will need to enable the "Hold undeliverable mail" field:



If you choose this option, make sure that you check the mail.box often so it does not become filled with held messages. When you are checking the mail.box, you will be able to release, forward, or delete any of these held messages.

Section 3: The Incident Handling Process

This section explains the incident handling process used to handle this incident.

Preparation

As with many organizations, we are in process of formalizing our incident handling processes and procedures. At the time of this incident, our drafted incident handling processes and procedures were under review by senior management, but had not yet been finalized or signed-off. Unfortunately, incidents do not wait until you have finished your preparation.

The drafted incident handling processes and procedures was based on the SANS *Computer Security Incident Handling Step-by-Step* guide. This gave us insight into how to apply and adapt proven incident handling policies and procedures into our environment. Having already been through parts of the preparation, some of the framework in the organization was starting to take shape, as interim informal processes and procedures.

We had established relationships with other key departments, including our help desk, legal, audit, human recourses, and information technology operational groups that notified us immediately of suspicious activity. We also had

established relationships with information asset owners (business owners) and management, who would guide and direct the decisions made through the incident handling process. Our security manager would interface with management regularly throughout an incident.

As part of our ongoing awareness program, employees were informed of our role and that anyone who discovers something that appears to be security related, should immediately contact our information security department.

Beyond establishing relationships and raising awareness, we had established informal agreements with different key departments allowing us to take control of any security incident and have immediate cooperation from any required resources to handle the incident as discretely and quickly as possible. Any team member participating in the handling of an incident was made fully aware of their responsibilities of being part of the team. We had also identified resources, by location and expertise, as well as contact numbers across the enterprise as part of our preparation.

On the policy side, we employed proactive techniques to mitigate the risk of incident occurrence and severity. Below are high-level summaries of select policies outlined in the Information Security Policy intended to demonstrate the preparation status (this is provided in lieu of being able to provide direct excerpts):

1. Maintaining current security patches for known system vulnerabilities.

   There was a process for patch notification, analysis, and deployment based on the risk of the vulnerability.

2. Maintaining an Anti-Virus Program.

   This included maintaining current anti-virus software on all systems as well as perimeter anti-virus measures.

3. Secure system configuration procedures.

   Security measures were included in the standard configurations of all systems.

4. Deployment of intrusion detection systems.

   Host and network based intrusion detection systems were deployed and monitored for suspicious activity.

5. Deployment of firewall devices.

   All external network access points were protected by a firewall.

6. Clearly defined system and information ownership.

Every information asset had a clear information owner defined in an information asset database.

7. Business Continuity and Disaster Recovery Plan

   Business continuity plans and disaster recovery plans were maintained and tested.

8. Strong password policies

   Strong password policies were defined and enforced across systems.

One last note - because we were in the preparation stages of our incident handling processes and procedures when this incident occurred, we did have some tools available for our incident handling tool kit. This included an incident laptop, a few notepads/pens and a couple spare hard disk drives. The laptop had Sniffer Pro 4.7, LC4, and Search 2.1.2 installed; we had recently purchased VMWare Workstation 3.2 to run Linux tools, but we had not yet had the opportunity to install the application.

Identification

It was 9:44AM on Thursday December 5, 2002 when I received a call from Bob, one of our system administrators. He said he was coming over and had something to show me. Bob is just down the hall from my office so in a few moments he was at my desk. He told me that he was just alerted to the fact that one of the Lotus Domino mail.box files had been corrupted, and when he went to look at the log files, he noticed something really weird.

So we pulled up the Lotus Domino log file as well as a live Lotus Domino activity terminal, and we read the following (this is just an example from the beginning, but the pattern is the same throughout the log file):

```
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:38 AM  SMTP Server: Message 001F472E received
12/04/2002 01:41:38 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:38 AM  Router: Message 001F4671 transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:39 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
12/04/2002 01:41:40 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 connected
12/04/2002 01:41:40 AM  SMTP Server: Message 001F4805 received
12/04/2002 01:41:40 AM  SMTP Server: 10.1.1.1 disconnected. 1 message[s] received
12/04/2002 01:41:40 AM  Router: Message 001F472E transferred to
MAIL.MYCOMPANY.COM for badguy@bogus.com via SMTP
12/04/2002 01:41:41 AM  Router: Transferring mail to domain MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM [10.1.1.1]) via SMTP
```

```
12/04/2002 01:41:42 AM  Router: Transferred 1 messages to MAIL.MYCOMPANY.COM
(host MAIL.MYCOMPANY.COM) via SMTP
```

As we scrolled through the log file we immediately recognized an email pattern occurring in short time intervals. On first inspection, it appeared as if the same email message was being continually sent to the same external email address from our Lotus Domino server. I immediately thought: “Could we be continually emailing an external SMTP server?”

Upon closer inspection, Bob noticed that the message ID (in bold) being sent out matched the message ID that had just come in. We both immediately thought that this was an email loop or some form of automated external attack continually sending the same message that is somehow being bounced back out.

One last thing that we did notice is that regular email traffic was still being processed around the incident. Though we could not quickly determine how long the attack had been running, given the way in which Lotus Domino breaks log files by default, but we could see that the pattern was going on for at least a few hours.

Having the log file demonstrating either an automated external email attack or a loop in our mail system, I immediately notified the information security manager indicating that we had a security incident. It was about 9:55AM on Thursday December 5, 2002 when I made initial contact with the information security manager.

After explaining the little bit I knew at the time that lead me to determine this was an incident, I was instructed to continue gathering information to identify the problem, identify the information assets involved, gather the necessary team, and prepare a recommendation as quickly as possible, while he notified management. At that point in time, given normal email processing appeared to be occurring around the incident and that the attack appeared to be an automated email attack or loop (and not a security breach), he did not want the mail system taken offline and felt it unnecessary at this point to perform any system, device, or log backups. This would change if we discovered something that made us believe the incident was something more.

It is important to note that this incident began at 1:41AM on Wednesday December 4, 2002 and we were not alerted to it until a mail.box file was corrupted on the Lotus Domino server at about 9:25AM Thursday December 5, 2002. The countermeasures we had defined, including IDS, were not effective because the loop (which we did not know at the time) was occurring between two boxes on the same segment and was not significant enough to completely DoS either server because of latency (though eventually it did play a role in corrupting one of the mail.box files on the Lotus Domino server).

It was nearing 10:00AM on Thursday December 5, 2002, when I grabbed my notebook, and began making notes. I quickly recorded the information assets known to be involved, as well as I began to record the incident (time/date of notification, notified by, steps taken to determine the incident).

We immediately identified the following as evidence to the incident:

1. Lotus Domino server and logs
2. InterScan VirusIWall NT server and logs
3. Firewall and logs
4. IDS and logs

There was no immediate back up made of the systems, devices, or log files, so there was no chain of custody procedures used, as decided by the security manager.

Containment

Once we had identified that a security incident had occurred, a team was brought together to assess and handle the incident at 10:05AM on Thursday December 5, 2002. This included another security analyst, a network analyst, the system administrator (who alerted us to this incident), and myself acting as the lead incident handler. This literally happened in a couple of minutes given the proximity of the necessary resources to my office and their availability at the time of notification.

It was now 10:20AM on Thursday December 5, 2002, and after bringing everyone up to speed on what we knew and the direction provided by the information security manager, we put a plan together to gather the information needed to analyze the incident. Each person was given a task (based on the identified evidence and where we felt relevant information would be), and we agreed to meet back in fifteen minutes to share what we had found. As well, if anybody found anything that could alter our current plan or what we believed the problem was, they would notify me immediately and we would re-group.

Bob, our system administrator, analyzed the Lotus Domino logs and reported back that the incident began on at 1:41AM on Wednesday December 4, 2002 and that the logs indicated a loop every 3-4 seconds based on the message ID. This is illustrated in the log file in the previous section (bolded). He further confirmed that normal email processing was occurring around the loop until the mail.box file was corrupted, which he believed was due in part to the loop.

Jan, our security analyst, analyzed our firewall and IDS logs. There did not appear to be any related suspicious activity on the IDS or in the firewall logs,

certainly none that mimicked such a constant interval from a single source. This ruled out an external automated process as a likely candidate.

Jack, our network analyst, used Sniffer Pro 4.7 that was setup on our incident laptop and captured the traffic between the servers. The actual capture is not included as it would be difficult to sanitize. The capture showed the same message being sent back and forth between the Lotus Domino server and the InterScan VirusWall NT server at a consistent interval.

I analyzed the InterScan VirusWall NT logs and discovered the same pattern Bob had seen in the Lotus Domino log files:

```
12/04/2002 01:41:37 Iscan-maild[254]: Message from: <>
12/04/2002 01:41:37 Iscan-maild[254]: Message to: badguy@bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:37 Iscan-maild[257]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
12/04/2002 01:41:39 Iscan-maild[196]: Message from: <>
12/04/2002 01:41:39 Iscan-maild[196]: Message to: badguy@bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Delivering mail to badguy@bogus.com through
mail.bogus.com
12/04/2002 01:41:39 Iscan-maild[145]: Forwarding mail to badguy@bogus.com to
10.1.1.2 at port 25
```

At this point it was 10:40AM on Thursday December 5, 2002, we re-grouped and discussed all the evidence that had been gathered. It was clear to us that we were dealing with an email loop. This was confirmed in the sniffer capture (still containing the original email received at 1:41AM on Wednesday December 4, 2002), as well as re-enforced with the Lotus Domino and InterScan VirusWall NT logs.

At 10:45AM on Thursday December 5, 2002, I notified the information security manager of our progress. The business wanted operations to continue, but at the same time, they were becoming nervous about the loop. They knew there was still a risk of another mail.box file becoming corrupted or additional emails configured similarly creating a more serious problem. Essentially, we were given a short time period to wrap up the initial analysis and management was expecting some quick recommendations of how the loop could be stopped and prevented.

At 10:50AM on Thursday December 5, 2002, after relaying the message to the team, we began to analyze the email message and three distinct parts caught our attention. The first thing that caught our attention was that the email was a failure of delivery notification:

```
Content-Type: message/delivery-status
Reporting-MTA: dns; smtp.mycompany.com

Final-Recipient: rfc822; nobody@ mycompany.com
Action: failed
Status: 5.1.1
Diagnostic-Code: X-Notes; User nobody (nobody@ mycompany.com) not listed in public
Name & Address Book
```

The second thing that caught our attention was that the original message was sent to a user who no longer worked for the company and their email account had been deleted (which would cause an undeliverable notification to be sent to the sender). The third thing we noticed was a peculiar domain name (illustrated as bogus.com here) and a typical spam header including random numbers at the end of the subject line:

```
Received: from mail.mycompany.com ([10.1.1.1])
        by smtp.mycompany.com (Lotus Domino Release 5.0.7)
        with SMTP id 2002120401405849:228429 ;
        Wed, 4 Dec 2002 01:40:58 -0400
Received: from mail.bogus.com by mail.mycompany.com (InterScan E-Mail VirusWall
NT);
       Wed, 04 Dec 2002 01:43:08 -0400
Message-ID: <00004e303ea5$000023ce$00004119 >
To: <nobody@ mycompany.com>
From: "Badguy" <badguy@bogus.com>
Subject: Blah 5467
Date: Mon, 02 Dec 2002 13:07:14 -1700
MIME-Version: 1.0
Reply-To: badguy@bogus.com
```

Wondering about the validity of the MAIL FROM domain, we did a quick nslookup on the domain and it returned to us 127.0.0.1 (internal host loopback). Using Sam Spade (www.samspade.org) we confirmed that this DNS resolution was not from any modifications to our DNS servers. Below is the output from Sam Spade:

```
From www.samspade.org:

dns bogus.com
Mail for bogus.com is handled by mail.bogus.com (10) 127.0.0.1
```

We now had a working theory about how the loop occurred:

1. A spam message was sent to a user whose account no longer exists. This message entered through InterScan VirusWall NT and was forwarded to Lotus Domino.
2. Lotus Domino, received the message and, because the account did not exist, created a delivery failure message to the sender.

3. Lotus Domino sent this message to the outbound SMTP server (InterScan VirusWall NT).
4. InterScan VirusWall NT received this message, did a DNS resolution (which resolved to its internal host loopback), and sent the message to itself.
5. InterScan VirusWall NT then forwarded the message to Lotus Domino.
6. Lotus Domino, seeing it was not destined for the local domain, forwarded it back to InterScan VirusWall NT, causing the loop.

So by 11:00AM on Thursday December 5, 2002, we understood how the message was looping and what caused the incident, but we still did not fully understand the reaction of Lotus Domino and InterScan VirusWall NT to this condition.

It is worth noting that given the nature of the loop and its consistent frequency between the two SMTP servers in our environment, there was no risk of this spreading to other systems so this was considered a self-contained incident. There was a recognized risk that another email or multiple emails could be sent to cause a more serious Denial of Service attack, but management did not believe this risk warranted taking the mail systems offline and isolating them.

For a more detailed description of these steps, please see the previous section "Description and Diagram of the Attack"

Eradication

Now that we understood the sequence of events, we set forth an action plan to both stop this loop from continuing and research the reaction of Lotus Domino and InterScan VirusWall NT to this condition.

The first discussion was around how we would stop this loop. It was important that we did not modify the configuration of either server involved so they would remain pristine as we continued to determine the reason Lotus Domino and InterScan VirusWall NT did not pick up on this occurrence. It was decided that we would create a DNS zone on our primary DNS server for bogus.com and a MX record pointing to a SMTP server running on our 'incident laptop' (IIS SMTP server). This would stop the loop, allow us to receive the message on our incident laptop that caused this denial of service, and ensure that the servers involved were not altered.

We assigned the task of stopping the loop to the network analyst and myself. In the meantime, the other team members would focus their attention on trying to discover why Lotus Domino and InterScan VirusWall NT reacted this way.

With our plan in place, I called the information security manager at 11:15AM on Thursday December 5, 2002, to bring him up to speed and gain approval for our plan. After a brief discussion, we got the go ahead and put the plan in action. At 11:25AM on Thursday December 5, 2002, we had installed and configured the SMTP service onto the incident handling laptop, created the DNS zone and MX entry, and successfully stopped the loop. We verified that the loop had stopped in the InterScan VirusWall NT and Lotus Domino log files; we then removed the DNS zone and stopped the SMTP server on the incident laptop.

At 11:30PM on Thursday December 5, 2002, after notifying the information security manager that the loop had successfully be stopped, we joined the rest of the team in researching the root cause.

By 12:30PM on Thursday December 5, 2002, we had discovered the vulnerabilities in both products outlined previously in this document, we had discovered the root cause – an email message that has a MAIL FROM field that resolves to the internal host loopback and a RCPT TO field that is to a non-exist local account. These were known vulnerabilities in InterScan VirusWall NT and Lotus Domino server, as well as the inability to detect this loop.

Recovery

Having determined the root cause of the exploit, we decided the best action to prevent the exploit from occurring again would be to have Lotus Domino hold undeliverable mail. This would prevent the loop from occurring, and the messages would be held in the mail.box files that could be checked on a schedule (at 7:30AM, 10:00AM, 12:00PM, 3:00PM and 5:00PM until a long-term solution was implemented) by the system administrator. Although this was not a long-term solution, it was done to buy us enough time to create an upgrade strategy for Lotus Domino. At 12:50PM on Thursday December 5, 2002, I called the information security manager and relayed our plan and had approval.

To do this, we first changed the configuration of our Lotus Domino server in our mirrored production environment at 1:00PM on Thursday December 5, 2002. Although we do not have an exact mirror of our entire production environment, key systems are mirrored, which includes our mail servers and relays. We enabled the “hold undeliverable mail” option as described in the *How to Protect Against It* section of this document.

Since our mirrored production environment does not connect to the Internet and we needed to simulate the problem, we created a DNS zone for bogus.com and created a MX record (mail.bogus.com) that resolved to 127.0.0.1.

Next, we connected to the mirrored SMTP server through telnet and typed the following:

```
telnet smtp.mycompanytest.com 25
helo bogus.com
mail from: badguy@bogus.com
rcpt to: nobody@mycompany.com
data
blah,blah
.
quit
```

We knew would cause an undeliverable notification message destined for an external domain. As expected, the message was correctly held in a mail.box file.

Confident that this solution would prevent the exploit, we documented and deployed the change into our production environment through our change control process at 1:30PM on Thursday December 5, 2002. Given that this was considered an emergency change, it was implemented immediately.

Once we had promoted the change to production, we replayed the same email that had caused the incident, and it was successfully held in the mail.box file on the Lotus Domino server.

After the final change was made, the incident was completely documented and stored in the incident logs. Lotus Domino was monitored hourly during the first day, and then on a schedule (7:30AM, 10:00AM, 12:00PM, 3:00PM and 5:00PM) until a long-term solution was implemented (this was specifically focused around the mail.box files and the holding of undeliverable notifications).

Lessons Learned

In the follow-up, we analyzed what allowed the incident to occur and ways we could improve processes, procedures, or policies to prevent or better handle similar incidents. Below are the outcomes from the standpoint of what went well and what could have gone better.

What Went Well

We concluded the following went well:

1. Response time after notification.

From the time that the incident was discovered until the time the team was fully deployed and assessing the situation was extremely short. People take an incident seriously and react accordingly. The contact process and resource identification drafted in the Incident Handling Process and Procedures worked well.

2. Cooperation from different team members and departments.

   Even though official incident handling processes and procedures do not exist, the team formed quickly with a shared goal. There was no issue from any department during or after the incident for utilizing their resources and pulling them from their other responsibilities.

3. Having the right resources to solve the problem.

   One key aspect of handling an incident as quickly and efficiently as possible is ensuring the correct resources are involved as part of the team. Identifying and including the correct team members across departments went well.

What We Can Improve On

We concluded the following could be improved on:

1. We need to be more proactive and diligent in our efforts in keeping systems patched from known vulnerabilities and our analysis of vulnerability notifications.

   Even though we have a process for patch notification, analysis, and deployment, we often focus on what are regarded as critically, pushing moderate or lower risk patches off and not giving them the same attention. This was escalated to management.

   We also recognized that this vulnerability has escaped our attention, and we need to make sure we full analyze the implications of vulnerabilities in our environment, especially in how they may correlate with other vulnerabilities.

2. We need to have formal incident handling processes procedures.

   This was probably the most discussed issue. From proper notification trees to clear expectations of team members, most people expressed the need for having this formalized and proper training provided. It was generally agreed that we were lucky on this incident, but we need to

continue to push for the formalized processes and procedures as soon as possible.

3. Better monitoring of log files

   Considering the incident took over a day to be discovered, we recognized the need to have better monitoring of our systems log files. We do have an enterprise-monitoring tool that will search log files and as a take-away, our operational group is looking into ways to automate log monitoring for re-occurring events that should be looked into.

4. Technology improvements

   As already planned, the need for deploying a solution to combat spam was highlighted. This was considered a larger issue and escalated to management. Furthermore, there was a recommendation to pursue upgrading Lotus Domino as soon as possible (after following testing procedures) to remove the vulnerability and the manual process of checking the mail.box files on the Lotus Domino server.

References

András Salamon, “DNS related RFCs”, http://www.dns.net/dnsrd/rfc/, (Dec 11, 2002)

CVE, “Lotus Domino SMTP server 4.63 through 5.08 allows remote attackers to cause a denial of service (CPU consumption) by forging an email message with the sender as bounce@[127.0.0.1] (localhost), which causes Domino to enter a mail loop.” , CVE CAN-2000-1203 (under review), http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-1203, (December 8, 2002).

CVE, “WebShield SMTP 4.5 allows remote attackers to cause a denial of service by sending e-mail with a From: address that has a . (period) at the end, which causes WebShield to continuously send itself copies of the e-mail”, CVE-2000-0738, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0738, (January 14, 2002)

CVE, “ArGoSoft Mail Server 1.8.1.7 and earlier allows a webmail user to cause a denial of service (CPU consumption) by forwarding the email to the user while autoresponse is enabled, which creates an inifinite loop”, CAN-2002-1005 (under review), http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-1005, (January 14, 2002)

Doug Sax, “DNS Spoofing (Malicious Cache Poisoning)”, November 12, 2000 http://rr.sans.org/firewall/DNS_spoof.php, (Dec 11, 2002)

E-Secure-DB, “Antivirus - Trend Micro”, Jan 21, 2002, http://www.e-secure-db.us/dscgi/ds.py/View/Collection-51, (Dec 9, 2002)

IETF RFC 2821, J. Klensin, Editor, “Simple Mail Transfer Protocol”, April 2001, http://www.ietf.org/rfc/rfc2821.txt?number=2821 (Dec 8, 2002)

IBM, “Domino R5 SMTP "Denial of Service" Attack Caused by Routing Loop”, Technote 191746, http://www-1.ibm.com/support/docview.wss?rs=0&org=sims&doc=DA18AA221C3B982085256B84000033EB, (Dec 8, 2002)

IBM, “Domino R5 Router Restrictions and Controls Explained”, http://www-1.ibm.com/support/docview.wss?rs=1&uid=sim1a786e37762de8341852568da004f9a80, (Dec 11, 2002)

IBM, “Domino R5 SMTP Message Is Transferred Repeatedly to 127.0.0.1”, Technote 188265, http://www-1.ibm.com/support/docview.wss?rs=0&uid=sim138d6a9cfd3bc072086256abf0075dc73, (Dec 8, 2002)

IBM, “Domino 5 Administration Help”, http://publib-b.boulder.ibm.com/lotus/21011400.nsf, (Dec 11, 2002)

Jason Coombs, “The Large-Scale Threat of Bad Data in DNS”, August 14, 2002, http://www.linuxsecurity.net/articles/network_security_article-5514.html, (Dec 11, 2002)

MARC, “Infinite loop in LOTUS NOTE 5.0.3. SMTP SERVER”, May 20, 2000, http://marc.theaimsgroup.com/?l=vuln-dev&m=95886062521327&w=2, (Dec 8, 2002)

Mice & Men, “DNS Spoofing”, http://www.menandmice.com/9000/9211_dns_spoofing.html, (Dec 11, 2002)

SANS Institute, “Computer Security Incident Handling Step-by-Step, Version 1.5”, May 1998

SecurityFocus, “Lotus Domino Mail Loop Denial of Service Vulnerability”, BID 3212, Aug 20, 2002 http://online.securityfocus.com/bid/3212, (Dec 8, 2002)

SecurityFocus, “Network Associates WebShield SMTP Trailing Period DoS Vulnerability”, BID 1589, Aug 18, 2000, http://online.securityfocus.com/bid/1589/info/, (January 14, 2002)

Security Tracker, “ArGoSoft Mail Server Lets Remote Authenticated Users Configure an Endless Loop to Cause Denial of Service Conditions”, SecurityTracker Alert ID: 1004951, http://securitytracker.com/alerts/2002/Aug/1004951.html, (January 14, 2002)

Trend Micro, “Solution 10525”, Mar 1, 2002, http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=10525, (Dec 8, 2002)

Trend Micro, “Trend InterScan VirusWall 3.5 Administrator’s Guide”, April 30, 2001

W. Richard Stevens, “TCP/IP Illustrated: Volume 1, The Protocols”, January 1994, Addison-Wesley Publishing Company

Zvon RFC Repository, “Simple Mail Transfer Protocol”, http://www.zvon.org/tmRFC/RFC2821/Output/, (Dec 8, 2002)