An Analysis of W32.Bugbear and the Technical and Procedural Controls Needed for Protection

GCIH Practical Assignment Version 2.1

By Edmundo Manrique CISSP
February 3, 2003

The increasing number of successful attacks on corporate and government networks demonstrate the need for continued improvement of security strategy and architecture. A combination of strong security procedures and technical components are often sufficient to stop or prevent an attack. On October 8, 2002, the author was involved in an internal incident that occurred due to an infection by Bugbear. The infection occurred after several controls mitigating the Bugbear worm had been implemented and was due to discrepancies between corporate and home security and to a violation of a security policy that was not enforced technically. The incident was an eye opener on the importance of polices and procedures in maintaining a secure environment.

The following analysis will demonstrate an infection of the Bugbear worm on the author's security environment, analyze its attacking strategies, and provide recommendations and steps needed to prevent Bugbear from infecting a network. The recommendations given include a mixture of security policies, procedures, and technical solutions. More importantly the solutions provided will not only help prevent an attack from Bugbear, but are part of any good security strategy and can be applied to a number of security threats.

It is the purpose of this author to keep all client and customer information confidential. For this reason, all scenarios given are a representation of what occurred during a Bugbear attack. All pertinent information has been modified to prevent any information regarding the network that this author protects from being disclosed.

The following definition and explanation of a worm is used throughout this analysis: A worm is very similar to a computer virus in that it is a computer programs that replicates itself and that often, but not always, contains some functionality that will interfere with the normal use of a computer or a program. The difference is that unlike a virus, a worm exists as separate entities and does not attach itself to other files or programs. A worm can spread itself automatically over the network from one computer to the next. A worm can take advantage of automatic file sending and receiving features found on many computers (Ball State University, "Virus Information", University Computing Services; http://www.bsu.edu/ucs/article/0,1299,6313~4488~1985,00.html).

This analysis is based on the template and requirements outlined in GCIH Practical Option 1 – Exploit in Action versions 2.1

<u>Exploit</u>

Name of Exploit:	W32.Bugbear
CVE Number:	CVE-2001-0154 Microsoft Security Bulletin (MS01-020) – "Incorrect MIME header can cause IE to execute e-mail attachment".

BUGTRAQ: 20010330 Incorrect MIME Header Can Cause IE to Execute E-mail Attachment.

Operating Systems Affected: Win 9X/ME/NT/2000/XP running IE 5.5 SP1 or earlier

Effected protocols, Services, Applications: IE 5.5 usage of malformed MIME and IFRAME when rendering html, SMTP, NetBIOS

Alias: W32/Bugbear-A
WORM_BUGBEAR.A
W32/Bugbear@mm
I-worm.Tanatos
W32/Bugbear
Tanatos

Description:

The W32.Bugbear worm was discovered on September 30, 2002. It exploited vulnerability in IE that allowed certain MIME types to be executed without user interaction. The worm has a payload that includes a backdoor, a key logger, shutdown of critical security functions, and mass emailing and network share propagation. Initially, Network Associates classified the worm as a category 2 by Symantec Security Response and as a medium threat. However, On October 2, 2002, due to Bugbear's rapid infection success, both Symantic Security Response and Network Associates increased the threat level to a '4' and 'high', respectively.

Bugbear is believed to have originated from Malaysia. The worm is written in Microsoft Visual C++ 6 and is compressed with v0.76.1-1.22. Bugbear mainly uses Outlook and Outlook Express to infect its victims. For this reason, Bugbear is mainly considered a mass-mailing worm. The worm takes advantage of a weakness requiring Outlook and Outlook Express to depend on IE for rendering HTML emails. By doing so, Bugbear exploits the MIME vulnerability in IE. Upon infection the worm uses existing email messages and performs "reply's" or "forwards" to distribute and disguise its infected messages. The subject of emails are variable and have been discovered as email attachments in subject lines such as "bad news", "Membership Conformation", Market Update Report, and "Your Gift". In addition to spreading through email, Bugbear also has the capability of spreading on a network through network shares including printers. This means that an infection can occur without receiving email.

Due to the variety of subject headers, Bugbear is difficult to identify. However, the infected attachment file has often been of 50,688 bytes size and the name of the attachment has been variable, with double extension ending in .exe, .scr, or .pif.

One key issue that may not be apparent is that of confidentiality. Because the email randomly forwards existing emails to random addresses in the victims address book, confidentiality of data and information become a key issue. Within my client, there are many discussions, issues, and negotiations that take place that are considered confidential. In addition, many users use email to send, receive, and store company "information" assets. This is essentially the data that as a security contractor, I am hired to protect. Therefore, not only does Bugbear infect systems and cause the security weaknesses that are discussed below, but also the infection has the potential of jeopardizing the confidentiality data. If data confidentiality is jeopardized, then the infection is successful in penetrating and bypassing the security controls that you or your company have implemented.

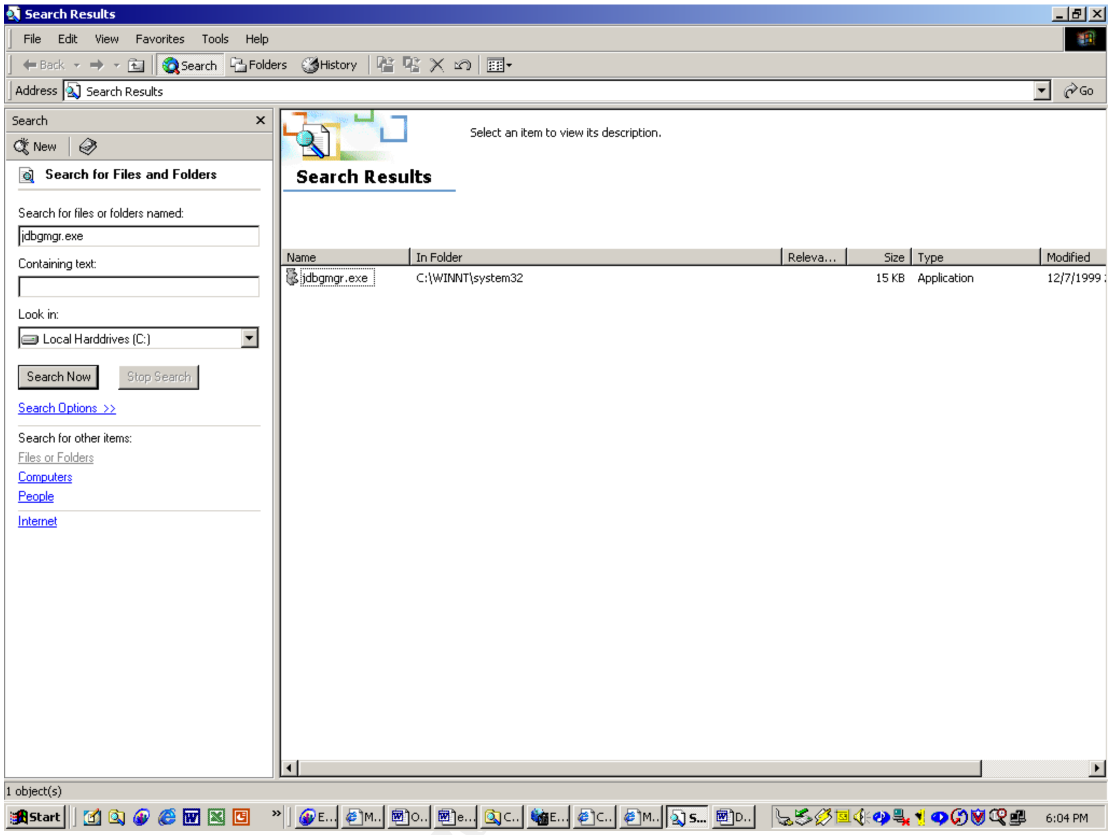
Variants: JDBGMGR.exe (hoax)

There are no known variants of Bugbear. Curiously, there is a hoax version named "JDBGMGR.exe hoax". The hoax tricks victims into deleting a file containing a bear icon that is similar to the bear icon found in the email attachment of the hoax. The icon contained in the hoax email is as follows:

Like Bugbear, the hoax is distributed via email. However, the email message is a hoax and not a virus or worm. The actual text of the email is as follows:

> "I found the little bear in my machine because of that I am sending this message in order for you to find it in your machine. The procedure is very simple:
>
> The objective of this e-mail is to warn all Hotmail users about a new virus that is spreading by MSN Messenger. The name of this virus is jdbgmgr.exe and it is sent automatically by the Messenger and by the address book too. McAfee or Norton does not detect the virus and it stays quiet for 14 days before damaging the system.
>
> The virus can be cleaned before it deletes the files from your system. In order to eliminate it, it is just necessary to do the following steps:
>
> 1. Go to Start, click "Search"
> 2.- In the "Files or Folders option" write the name jdbgmgr.exe
> 3.- Be sure that you are searching in the drive "C"
> 4.- Click "find now"
> 5.- If the virus is there (it has a little bear-like icon with the name of jdbgmgr.exe DO NOT OPEN IT FOR ANY REASON
> 6.- Right click and delete it (it will go to the Recycle bin)
> 7.- Go to the recycle bin and delete it or empty the recycle bin.
>
> IF YOU FIND THE VIRUS IN ALL OF YOUR SYSTEMS SEND THIS MESSAGE TO ALL OF YOUR CONTACTS LOCATED IN YOUR ADDRESS BOOK BEFORE IT CAN CAUSE ANY DAMAGE."

When a user follows the instruction above, the file JDBGMGR.EXE is deleted. This file serves as the Microsoft Debugger Registrar for Java. The search window for the JDBGMGR.EXE is as follows:



Search for files or folders named: jdbgmgr.exe

Containing text:

Look in: Local Harddrives (C:)

Select an item to view its description.

Name	In Folder	Releva...	Size	Type	Modified
jdbgmgr.exe	C:\WINNT\system32		15 KB	Application	12/7/1999

1 object(s)

Although this is a hoax, many users around the world have been affected. There are published repairs for how to restore the file from backup (http://vil.nai.com/vil/content/v_99436.htm), but the issue of mitigating incidents, including hoaxes, should be part of both an incident handling procedures and training to the user community. In the case of this hoax, no incidents were identified within the environment that I protect. However, all users have been trained not to forward emails or follow instructions that change or alter any of the settings on their computer unless they can confirm that the instructions came from their technical support team.

Bugbear "does not contain a bear icon, but rather a generic icon typically associated with EXE files" (McAfee Security, http://vil.nai.com/vil/content/v_99728.htm). This generic icon looks as follows:

References:

The following URL's provide further information relating to the Bugbear worm and have been used within this analysis as reference.

- http://vil.nai.com/vil/content/v_99728.htm
- http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html
- http://www.mcafee.com/anti-virus/viruses/bugbear/
- http://www.sophos.com/virusinfo/analyses/w32bugbeara.html
- http://www.f-secure.com/bugbear/
- http://www.theregister.co.uk/content/56/27380.html
- http://vil.nai.com/vil/content/v_99436.htm
- http://news.com.com/2100-1001-960365.html?tag=lh

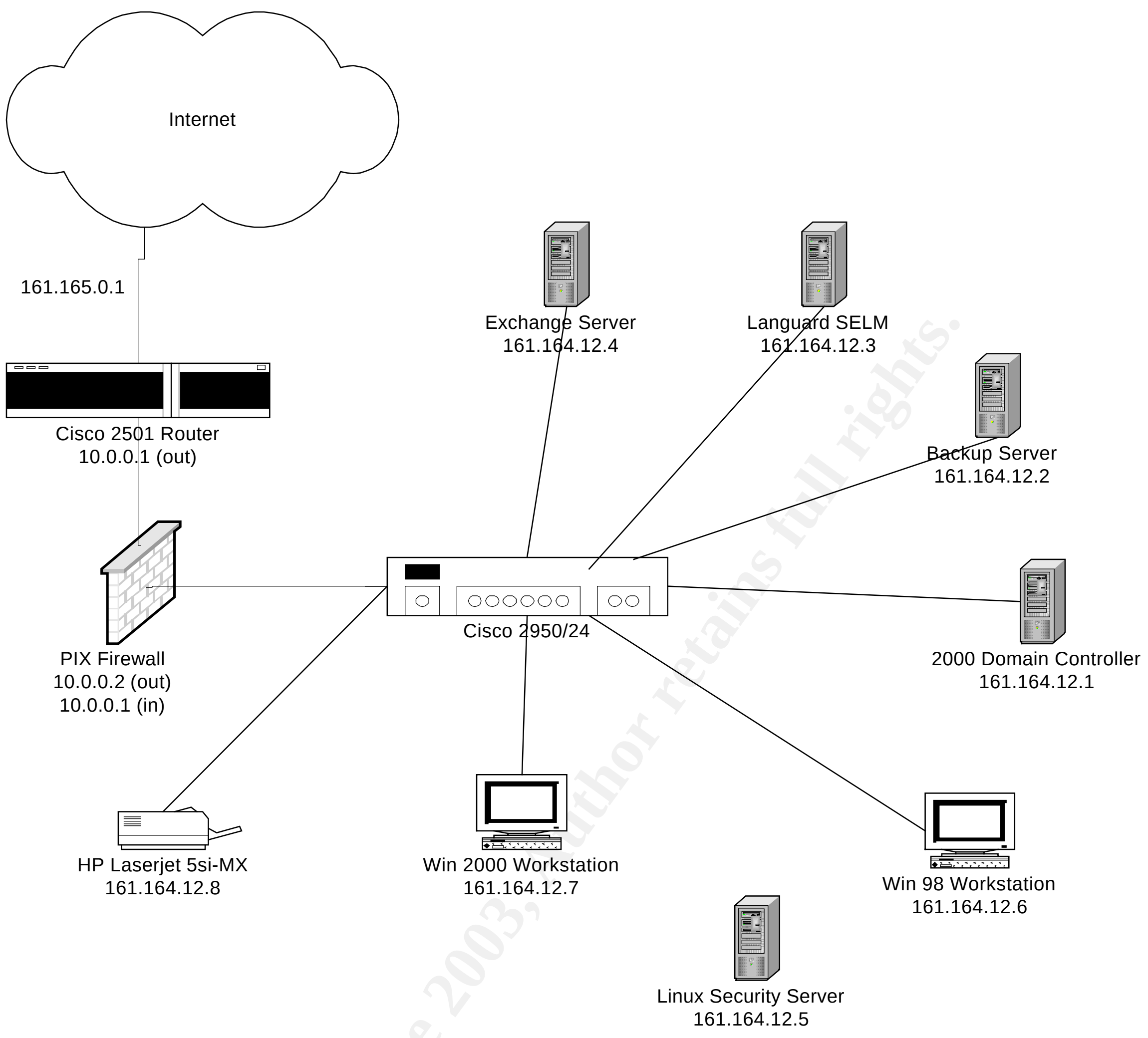
The Attack

Network:

Many times, a vulnerability or incident occurs due to a weakness in the technical controls in an environment. These weaknesses sometimes reside within the network architecture. In this particular incident, the incident occurred due to a weakness of enforcing security policy. For purposes of describing the weakness that allowed the incident to occur, a detail explanation of the whole network architecture is not provided. Instead a simplified version of the network is depicted. This explanation allows the reader to obtain a good understanding of the overall network and business environment as required to understand the incident. Key components that should be assumed are as follows:

- A demilitarized zone (DMZ) that resides between two firewalls
- An internal LAN protected by one firewall
- The internal LAN is not accessible from the internet
- DHCP is enabled within the internal LAN
- All network servers run NAI Netshield 4.5
- All Exchange servers run NAI Groupshield 4.5
- NAI McAfee Virus Scan 4.5 protects workstation.
- All anti virus products obtain new signature updates daily

The diagram below is a basic simplified version of the network where the incident occurred. The key points mentioned above should be assumed to be part of this network. (Note: The network is more complicated than pictured. However, for explanation purposes, a more complicated network is not needed).



The network contains two thousand users using a mixture of Win 98 and 2000 desktops. Users have access to various applications, shared network drives, and printers as necessitated by business requirement. All users have Internet access for purposes of conducting research. Users are administered using Active Directory Services and authenticate to a 2000 Domain Controller.

Microsoft Outlook is the only email communication medium used in this company. Hotmail, Yahoo, and other public email communication websites are being blocked. The Exchange Server controls all Exchange and Outlook functionality.

The network is primarily a Windows shop; Unix, Linux, and Macintosh systems are non-existent except for a security scanning laptop that runs Linux.

All network components are Cisco based. Cisco components include an edge router, a pix firewall, and one switch. There are plans for transitioning to a dual firewall configuration, but this is not the current case.

Additional components include an HP Laser Jet Printer, a backup server, and an internal distribution, warehousing, and supply-chain application that use SQL Servers 2000.

Network Component Details:

- Cisco 2501 Edge Router: The edge router is a Cisco 2501 with the Inter Operating System (IOS) Version 12.0 (9). It is configured with one serial facing the WAN, and one Ethernet facing the LAN.

- Cisco Pix Firewall: The firewall is a Cisco PIX with the PIX operating system version 6.3. The open ports are, 80 (web traffic) outbound, 443 (SSL) outbound, 25 (SMTP) outbound and inbound, and 21 (FTP) outbound.

- Cisco 2950/24 switch: The switch is a Cisco 2950/24 with IOS version 12.

- HP Laser Jet 5si-MX: The Network printer is a HP Laser Jet 5si-MX.

System Component Details

- 2000 Domain Controller: This server is a Dell 2600 with 1 Gigabyte of memory, a 100+ gig hard drive and two Xeon 2 Gigabyte processors. It is a Windows 2000 Advanced Server version 5, service pack 2. This system is the Active Directory Services Domain controller.

- Languard SELM: This server is a Dell 2600 with 1 Gigabyte of memory, a 100+ gig hard drive and two Xeon 2 Gigabyte processors. It is a Windows 2000 Advanced Server version 5, service pack 2, and Microsoft SQL Server 2000 loaded. This server is the Languard SELM storage system. Languard SELM is a tool used to collect the event log files other computers inside a network. Languard SELM provides for central management of all Microsoft System event logs.

- Microsoft Exchange Compa1 Proliant Server: This server is a Dell 2600 with 1 Gigabyte of memory, a 100+ gig hard drive and two Xeon 2 Gigabyte processors. It is a Windows 2000 Advanced Server version 5, service pack 2. This server is the Microsoft Exchange 5.5 server.

- Backup Server: This server is a Dell 2600 with 1 Gigabyte of memory, a 100+ gig hard drive and two Xeon 2 Gigabyte processors. It is a Windows

2000 Advanced Server version 5, service pack 2. Veritas Backup Exec version 8.60 Rev 3808 is used for backups.

- Linux Security Server: The Linux notebook is a Toshiba Tecra 8100 with 128 megabytes of memory, a 10-gigabyte hard drive and a P 3 800 Mhz processor. It is running Linux 7.1 as the operating system. It also has Nmap, and Nessus installed as the vulnerability assessment software.

- Workstations: Workstations specs vary. Each department has been responsible for their own purchases and has determined the specs needed on each workstation based on business need. However, all workstations are running either Windows 98 or Windows 2000.

Business Description:
It is important to not only understand the technical network environment, but also the business environment. The network supports a variety of business usage and teams. As part of this large organization, there are several divisions each having their own management and overall purpose. In addition, each division is allowed to hire their own people including different contractors as needed. For this reason, the environment contains several types of users including company employees and a number of different contractor companies. This creates a problem in that although the network supports all these users, it is sometimes difficult to enforce common standards and policies across all divisions due to political and business strategy differences.

Protocol Weakness Description:

Simple Mail Transfer Protocol (SMTP)

SMTP is protocol for sending and receiving email messages between servers. It is normally used in conjunction with either the POP3 protocol or the Internet Message Access Protocol (IMAP). SMTP is used for sending email. Either IMAP or POP is utilized for receiving messages by a client. The Bugbear worm utilizes its own SMTP engine to email copies of itself to email addresses found on the infected computer.

Multipurpose Internet Mail Extensions (MIME)
MIME is an Internet standard for encoding binary files as email attachments. If an email contains a binary attachment, the email is required to contain a mail extension that identifies the type of file attachment. Internet Explorer processes the MIME extensions and executes the attachment based on the rules associated with that extension's MIME heading. A weakness exists in that certain unusual MIME headers bypass the processing controls by Internet Explorer.

The Bugbear worm bypasses the Internet Explorer MIME processing controls by changing the MIME header of an executable to an unusual MIME header. Upon review of the faulty MIME header by Internet Explorer, the executable would run when the email carrying the executable was reviewed. A correct MIME header would require the user to double click on the executable in order to run it.

Internet Explorer IFRAME
Internet Explorer oversees the restrictions imposed on what actions a web site can take on a visiting computer. When software on the web server requests that particular action take place on the visiting computer, Internet Explorer allows the action to proceed only if the action has been configured and is permitted. However, normal security checks are not present when the requests originate within an IFRAME.

IFRAME, or floating frame, is a sub-window of the main browser window and can be anywhere within a standard HTML document. An IFRAME is its own window and can operate independently of the window that contains it. In unpatched versions of Internet Explorer, normal security checks are not present within IFRAMEs.

A combination of IFRAME tag and a mismatched MIME attachment leads to the automatic execution of malicious code upon review of a message on unpatched versions of Internet explorer.

Network Basic Input/Output System (NetBIOS)
NETBIOS allows applications on different systems to communicate within a LAN. It also allows a host machine to share files or folders across a network. The underlying mechanism for this sharing is provided by the Server Message Block (SMB) and the Common Internet File System (CIFS) protocols.

Misconfigured or unprotected network shares expose files and allow for the potential of an attacker to obtain full control of the host. The Bugbear worm uses unprotected network shares as a means of propagation.

How The Exploit Works:

Bugbear is a very complicated worm with very powerful features. In order to provide a logical explanation of its operation, the explanation of how it works has been organized into two categories: Propagation and Means of Infection; Infection Execution.

Note: In order to provide an adequate technical explanation of Bugbear, the author has relied on personal experience and reference information. Like many others who have researched a worm's capability, writing about it requires the use of a variety of sources. The sources used have been listed above within the section "References". Still, the author would like to thank the Symantic

Corporation as their description of Bugbear was extremely helpful. Some of the concepts and worm details in the Symantic report appear within the sub-sections "Propagation and Means of Infection" and "Infection Execution". The Symantic Bugbear report can be found at http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

Propagation and Means of Infection:
Although there are various ways for Bugbear to infect an environment, the primary method of infection is through email. Thus, Bugbear is known as a mass mailing worm and arrives as an email with an attached .scr, .pif, or .exe worm file.

As part of the mass-mailing portion of the Bugbear worm, the worm retrieves the user's email address and SMTP server from the registry key:

HKEY_CURRENT_USER_\SOFTWARE\Microsoft\internet account Manager\Accounts.

The worm then uses its own SMTP engine to send itself to all email addresses found in the victims computer. The worm also uses existing email messages and performs "reply's", "forwards", and "From's" to distribute and disguise its infected messages. The result is variety of new infected emails from both spoofed and victim's email addresses. The subject of emails are variable and have been discovered as email attachments in subject lines such as "bad news", "Membership Conformation", Market Update Report, and "Your Gift". Because the worm takes advantage of Windows systems containing un-patched versions of the Internet Explorer's IFRAME and Incorrect MIME Headers vulnerabilities, a target user is not required to open the infected attachment. Instead, the vulnerability provides the capability of infecting an un-patched computer by allowing Outlook to automatically execute the attachment as soon as an infected email is previewed or read.

A secondary means of distribution is through unprotected network shares. The worm will search for open file shares in the local network. When an open file share is found, the worm attempts to copy itself to the Startup folder of the remote computer. The result is an infection of the computer upon restart of the system. The worm then continues to distribute either through mass emails or through further unprotected network shares. The worm also attempts to copy itself via shared printers. This causes the printer to print the virus binary code, leading to waste of paper and time spent on troubleshooting affected printers. Troubleshooting printers is more of a nuisance, but depending on the type of organization being protected, this in itself can be considered a minor vulnerability due to the notion of the 'availability' of these printers.

Infection Execution:

Upon infection of a computer, the worm copies itself as an .exe to the System and Startup folders. Depending on Windows Operating System running on the victim's machine, the actual path to get to the System and Startup folders may be different. However, the end result is an .exe in the System and Startup folders with a name chosen by the worm.

The worm also creates three encrypted .dll files in the system folder and two encrypted .dat files in the windir folder. The names of these .dll and .dat files vary. One of the .dll files is used by the worm to activate a key logger named "PWS-Hooker.Trojan". This essentially acts as a keystroke recorder by performing a sort of 'man in the middle attack'; intercepting keyboard messages and forwarding it to the next hook in the chain. The worm occasionally sends the file captured by the keystroke recorder to a few designated emails that are stored in encrypted form in the worm's code. This leak of information can lead to future breaches of security as the potential of sending confidential information such as passwords or proprietary data exists. An intruder can then, at a later date, use this information to infiltrate an environment.

Another one of the created files contains a password that can be used by an attacker to establish a future backdoor connection with the victim's computer. This backdoor provides the potential for the virus author to use a web browser to access the infected computer. Through this interface, the intruder would be able to browse, copy, alter or delete local files, and execute programs on the infected machine. In addition, the intercepted keystroke file created by "PWS-Hooker.Trojan" would be available to the intruder. As part of the backdoor capabilities, the worm opens port 36794 and listens for commands from the intruder.

In addition to the worm's creation of the .dll and .dat files, the worm also modifies or obtains information from a number of registry keys including reading the SMTP information from the registry key as described in the previous section:

HKEY_CURRENT_USER_\SOFTWARE\Microsoft\internet account Manager\Accounts.

The first registry that is modified by Bugbear is:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnc
e

The value name inserted in this key varies according to what random letters the worm chooses and the worm's file name. However, it does follow the format <random letters> <worm's file name>. Although the infected computers operating system will remove the values from this key upon shutdown or restart, the worm will recreate the value each time Windows is launched.

Once active, Bugbear will attempt to shutdown several of the security controls present in the victim's machine. This includes personal firewalls and virus engines. The targeted processes include:

Zonealarm.exe, Wfindv32.exe, Webscanx.exe, Vsstat.exe, Vshwin32.exe, Vsecomr.exe, Vscan40.exe, Vettray.exe, Vet95.exe, Tds2-Nt.exe, Tds2-98.exe, Tca.exe, Tbscan.exe, Sweep95.exe, Sphinx.exe, Smc.exe, Serv95.exe, Scrscan.exe, Scanpm.exe, Scan95.exe Scan32.exe, Safeweb.exe, Rescue.exe, Rav7win.exe, Rav7.exe, Persfw.exe, Pcfwallicon.exe, Pccwin98.exe, Pavw.exe, Pavsched.exe, Pavcl.exe, Padmin.exe, Outpost.exe, Nvc95.exe, Nupgrade.exe, Normist.exe, Nmain.exe, Nisum.exe Navwnt.exe, Navw32.exe, Navnt.exe, Navlu32.exe, Navapw32.exe, N32scanw.exe, Mpftray.exe, Moolive.exe, Luall.exe, Lookout.exe, Lockdown2000.exe, Jedi.exe, Iomon98.exe, Iface.exe, Icsuppnt.exe, Icsupp95.exe, Icmon.exe, Icloadnt.exe, Icload95.exe, Ibmavsp.exe, Ibmasn.exe, Iamserv.exe, Iamapp.exe, Frw.exe, Fprot.exe, Fp-Win.exe, Findviru.exe, F-Stopw.exe, F-Prot95.exe, F-Prot.exe, -Agnt95.exe, Espwatch.exe, Esafe.exe, Ecengine.exe, Dvp95_0.exe, Dvp95.exe, Cleaner3.exe, Cleaner.exe, Claw95cf.exe, Claw95.exe, Cfinet32.exe, Cfinet.exe, Cfiaudit.exe, Cfiadmin.exe, Blackice.exe, Blackd.exe, Avwupd32.exe, Avwin95.exe, Avsched32.exe, Avpupd.exe, Avptc32.exe, Avpm.exe, Avpdos32.exe, Avpcc.exe, Avp32.exe, Avp.exe, Avnt.exe, Avkserv.exe, Avgctrl.exe, Ave32.exe, Avconsol.exe, Autodown.exe, Apvxdwin.exe, Anti-Trojan.exe, Ackwin32.exe, _Avpm.exe, _Avpcc.exe, _Avp32.exe

The worm also reads the contents of Personal value in the registry key:

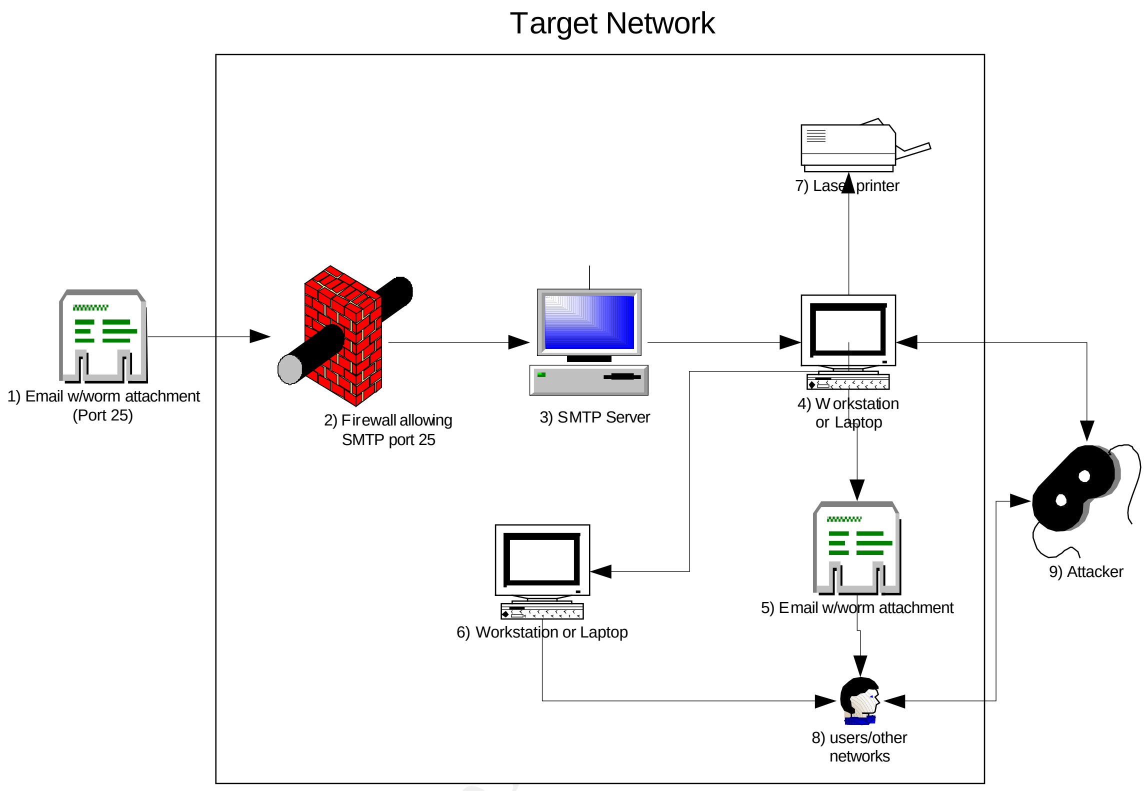
SOFTWARE\Microsoft\Windows\CurrentVersion\Explore\Shell Folder

The contents of that location are then listed and the file names are used to name the worm's file name for future propagation. In addition to the names created from the value of this key, the following name have also been used by the Bugbear worm:

- readme
- Setup
- Card
- Docs
- news
- image
- images
- pics
- resume
- photo
- video
- music
- song
- data

Diagram and Description of the Attack

Below is a basic theoretical step-by-step process of what components need to be present for Bugbear to infect a network and how Bugbear exploits these.



1. Bugbear arrives either through email or a shared network drive. In this scenario, the infection occurs via email through SMTP port 25.
2. The firewall protecting the network must have SMTP port 25 open.
3. The SMTP (Email server) does not block common virus/worm attachments such as .exe, .scr, .pif, .bat, .vbs and the virus engine is not catching the infected message. The server forwards the email to the recipient.
4. The recipient receives the message. Internet Explorer of version 5.01 or 5.5 and is vulnerable to Microsoft Security Bulleting (MS01-020): “Incorrect MIME Header Can Cause IE to Execute E-mail Attachment” and its virus engine does not catch the worm. Outlook requires the Internet Explorer to render the message. Upon the review of the message, the Bugbear worm begins infection as its MIME heading allows its auto execution. The worm creates its .dll and .dat files and begins altering key components of the workstations. The propagation process begins.
5. The worm finds the users address book and uses its own SMTP engine to create new infected messages. These messages can be in

the form of a reply or forward. The messages can also be spoofed by inserting one of the stolen email names in the "From" field.

6. Unprotected NETBIOS Network Shares allows the worm to spread throughout the network. The worm continues to search for network shares.
7. As part of Bugbear's attempt to spread through network shares, printers are affected by causing them to print out the worm's binary code. This is due to an error in the worm's code.
8. Propagation continues and Bugbear continues to infect unprotected and vulnerable users.
9. As part of Bugbear's payload, the worm creates a backdoor by opening port 36794. If the port is opened at the firewall, the victim system will be awaiting orders from the attacker. In addition, Bugbear will attempt to send emails every 20 seconds to a number of email addressees. These emails contain data captured by the key logger. This creates a bi-directional flow of data between the victim system and the attacker.

Bugbear's attack on this system came as a surprise as various technical controls protecting the network from Bugbear were in place. For example, all known desktops and servers were running McAffee with the latest virus update and had been patched. The SMTP server was blocking common virus/worm attachments. The firewall had been configured with strong security rules allowing only specific traffic through certain ports, thus all traffic going through port 36794 was denied. Nevertheless, the infection bypassed all these controls as it arrived via a personal laptop that had been infected with Bugbear. For further details on the attack and how it was identified and contained, see the section "Incident Handling"

Signatures

One of the advantages to an administrator when searching for a possible infection by Bugbear is its numerous signatures. These are mainly due to its extensive functionality and alterations to the systems it infects. The signatures that Bugbear creates are listed below and have been prioritized based on the author's opinion.

1. Random Network Printer Activity: An infected machine will attempt to propagate itself through network shares. When Bugbear reaches a shared network, it will begin printing its binary code. Although this type of activity may indicate infection by other culprits, such activity indicates that an infection has occurred and that further investigation is needed.
2. Port 36794: Once an infection has occurred, Bugbear will create a backdoor via port 36794. The infected system will then be in listening mode awaiting directions from an attacker via port 36794.

3. New ***.exe files: Upon infection, Bugbear creates random .exe files in the Startup and System folders.
4. Modification of HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce: Bugbear modifies this key and inserts a value with the format of <random letters> <worm's file name>.
5. Email attachment file of 50,688 bytes size with a variable name and double extension ending in .exe, .scr, or .pif.
6. .dll files: Three *.dll files in the system folder.
7. .dat files: Two *.dat files in the windir folder.
8. Shutdown of security processes as listed in the "How the Exploit Works"

How to protect against Bugbear

A good patch updating and installing process is a major part of any good security methodology. The process should include a set schedule for periodic reviews and installation of patches as well as a strategy for testing and installing crucial patches. The "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment" vulnerability has been around since March of 2001. Therefore, the most important step for protection from Bugbear is patching all affected systems (http://www.microsoft.com/windows/ie/downloads/critical/q290108/default.asp). Systems that are vulnerable use unpatched versions of Internet Explorer 5.01 or 5.5. IE version 6.0 is not vulnerable and upgrading to it is also a solution.

Another major component of a good security methodology is the timely upload of new virus signatures. In our environment, we use McAfee anti virus and new signature files are updated daily. All major virus protection services have a strategy in place for how to deal with new viruses or worms. By keeping all systems up to date with new virus files, the time between the discovery of a new worm or virus and when the system is protected is minimized. This decreases the time period in which a system can be infected.

A good firewall policy is another way for protecting from many attacks. A good firewall policy is that any ports that are not needed are closed and traffic not allowed. In the case of Bugbear, the firewall rules may be strong, but it is still possible for an infection to come through via the port used for emails. Although closing this port is not a long-term solution, it is a way to prevent further infection from the outside upon discovery of the Bugbear infection. In addition, if the firewall rules are good, they will prohibit the attacker from communication to the infected system via port 36794.

A secondary way of protecting from the Bugbear worm is through the use of a mail server that block common virus/worm attachments such as .exe, .scr, .pif, .bat, .vbs

Incident Handling Process

The driving purpose of security is to protect an asset whether that asset is data, information, etc. An attack on an asset is an attempt to by-pass the security controls protecting that asset. As part of these controls, a process needs to be in place that will help identify an attack and outline the necessary steps for recovering from this attack and improving the security controls protecting the asset.

As mentioned earlier, I am part of a team supporting the network that is made up of contractors. Luckily, we are all from the same company. However, there are several other divisions who have different contractors supporting different applications. The client/contractor relationship is that my company provides support for the infrastructure, while other companies provide application support for systems that they have developed under separate contracts.

My company's teams are divided up into small 4-5 member teams. These teams include a Server team, Security Team, Network Team, Systems Team, and Exchange Team. In addition, there is a help desk team that acts as the main tier 1 customer services team for the client.

I am part of the Security team and we are responsible for providing security operations and security oversight to teams supporting the network. Overall, the Security team has a good combination of technical and procedural strengths. As part of our security responsibilities, we were responsible for creating an Incident Handling Process and Incident Handling Team. Due to separation of duties between our sister teams, the Incident Handling Team is composed of 1-2 members of each system team and two members of the security team. Including within this team is a member of upper management and the lead for the Help Desk Team. The Security Team maintains operational authority during any incident and behaves as the captain during an incident. A key weakness in this team organization is that no members of other divisions (different companies) are part of the Incident Handling Team. Instead, we report directly to our clients and rely on them to fill in the hole as needed.

During the Bugbear incident, the Incident Handling team was called into action and our Incident Handling process was followed. Below are the steps that we have taken as part of our Incident Handling Process.

Preparation

Security from an operational perspective requires various tasks to be performed. The majority of tasks that the Security Team performs are part of the Preparation phase of the Incident Handling process.

The primary and most important operational task is to perform vulnerability assessments. The Security team is responsible for conducting vulnerability assessments on all components for each of the network support teams. We do this on a recurring monthly schedule. This schedule has been published and each of the peer team leads know the exact dates for when their components will be tested months in advanced. A published schedule is very important as it prevents any pushback from the team as they have been given an adequate amount of time for planning and allows them to incorporate this activity into their normal operations. In addition, we require that each team provide a resource to work through any vulnerability that may be found. This resource serves as the point of contact for any problems encountered and is responsible for fixing any holes found.

Prioritization of vulnerability fixing is also important. Vulnerabilities that are found should all be classified as critical, high, medium, and low. For each of the classifications, an agreed upon mitigation schedule should be in place. In our environments, critical vulnerabilities are fixed immediately. High vulnerabilities are fixed within 24 hours. Medium vulnerabilities are fixed within 48 hours. Low vulnerabilities are fixed within a week. In addition, a living document is kept for each team describing each vulnerability and how it was mitigated or fixed. This is important is allows for historical information to be kept. This document also addresses the numerous false positives that are part of any vulnerability tests. By documenting which vulnerabilities were false positives and how these were confirmed, time is saved in future assessment as a list is available for comparison and instruction.

From a technical perspective, it is important that two different tools be used if possible. We use Nessus and Whisker. All of these tools are updated each time a test is scheduled. Also, both Nessus and Whisker are free-ware tools. For political reasons that do not encourage use of free-ware tools and for reasons of obtaining technical support, our client is thinking of purchasing ISS Systems Scanner. This is a commercial vulnerability scanner. In large organizations where stockholders can be impacted by negative security activity, having a commercial tool may be a good way of protecting the team and doing due diligence. However, if your company cannot or will not purchase a commercial tool, performing the vulnerability tests with a free tool like Nessus works just fine.

The second most important task is to perform patch updates. Some security folks believe that patching take place immediately after a patch is available. However, due to the resource intensive nature of patch testing and installing, in our environment patching is officially done quarterly. However, we do have a process in place for notification to teams when critical patches are necessary. These critical patches do not follow the quarterly schedule. Instead, the first step is to determine if the patch is fixing a vulnerability that is already mitigated. If the vulnerability is mitigated, the second step is to determine if there are simple steps that can be done to mitigate or fix the vulnerability without installing the patch. If no simple fix is available, the third step is to determine the criticality of the component in question. If the component is critical, then a testing period of three days is provided to test the patch. If the component is not critical and can be easily repaired from backup, the patch is installed immediately. This process may appear a little odd as the most critical component that is vulnerable is left vulnerable during the testing period. However, many patches in the past have caused critical systems to crash and we prefer to have more control for how a patch gets installed. Also, both the Security Team and the team responsible for the component keep a close eye on the system to ensure safe and normal performance on that system.

The third most important task is to ensure that all components are protected by Anti-virus software running the latest signature file. In our environment, we use the McAfee 4.5 for workstations, NAI Group Shield 4.5 for Exchange Servers, and NAI Net Shield 4.5 for all other servers. The responsibility for making sure that all components have this software and are running the latest signature file actually falls within another peer team. However, we work closely with them in researching vulnerabilities that may be exploited by a virus or worm. The team responsible for the Anti-virus Software sends out an email each time a new signature file is available to all members in the environment. Each recipient is then reminded to verify that they are running the latest signature file by a specified date. The actual updating of the new signature file is actually automated by the anti-virus update service. However, providing a secondary check by the user is a good way to identify if the automatic update service failed. If there are discrepancies between the actual .dat file and the expected one, the user is advised to contact the Help Desk to request technician help.

The fourth most important task is to monitor system log files and IDS logs. At the time of the Bugbear infection, an IDS system was not in place. However, we are now operating with a Network IDS system in place that oversees network traffic at critical junctions within this environment. For confidentiality purpose, the name of the IDS system is not disclosed. Operational requirements for any IDS system should include a schedule and process for maintaining the latest signature files. Here again, in our environment, we have separation of duties. Although the Security team is ultimately responsible for the IDS system, the responsibility for updating the latest signature files is held by a peer team. Another key operational requirement is reviews of the IDS logs. We have an automated process that sends alerts to the Security team for alarms that have been triggered. These are classified as critical, high, medium, and lows. The team reviews all critical and highs as they are generated and reviews mediums and lows on a daily basis. Also, a spreadsheet of all alert and actions taken are kept. This provides historical data and allows for tracking of false positives.

A tool called Languard SELM does the monitoring of system log files. This tools acts as the central collector of all server event logs. Similar to the IDS system, Languard SELM organizes event logs into critical, high, medium, and lows. When a threshold is exceeded, Languard SELM will send alert to the administrators. The security team reviews these logs on a daily basis and also keeps a spreadsheet of all logs examined. In addition, daily, weekly, and monthly reports are generated to determine trends and normal activity in the environment.

Another key component is a strong backup process. We use Veritas Backup Exec Version 8.6 Rev 3808. Like in the anti virus and IDS system, there is separation of duties and another team is responsible for the execution of the backups. We provide oversight as needed. All servers are backed up in a daily, weekly, monthly schedule. Therefore, a system could always go back to either to the last day, the last week, or the last month. All monthly tapes are kept offsite in a secure location. The media used is DTL 40 Gigabyte tape.

Additional security tasks that fall within the preparation phase are as follows:

- Monthly Baselines: Baselines include a review of open ports and services on each computer, a list of patches updated on each computer, and a list of file and registry directory. Tools used for this include Languard Network Scanner, NETSTAT –A command, Sysdiff (using Sysdiff /Snap and Sysdiff /Diff), and Hot Fix Checker.
- Quarterly Password Audits: Each computer and user's network password is audited each quarter to ensure that it is meeting the password policy (12 character, alpha, numeric, special character) and cannot be broken. We use L0pthcrack to audit our passwords. A communication of cracked passwords is provided to the client for review and follow up with end user.
- Subscription to Security Alerts: These include subscription to SANS, FedCIRC, and Microsoft security alerts. In addition, each security team member is expected to remain up to date with new vulnerabilities and worms that are in the wild. To do this, each Security team member subscribes to alert services mentioned and is allowed one hour a week for vulnerability research.

Identification

A call was received from the client security manager that a share printer in another division was printing unusual text. He suspected that it was some time of infection and asked us to investigate. Luckily, the division affected was made up of other client personnel and not other contractors, as this would have increased the likelihood of pushback from the affected division.

Initially, the security team did not believe it was an infection of Bugbear as several technical controls were in place to prevent an attack from Bugbear. However, the client believed this was an incident and the Incident Handling team was called into action.

Further investigation revealed the identity of the system generating the print jobs. The system was a personal laptop that a client user periodically brought to work. Upon review of the system, further Bugbear signatures were found. These included:

- A service listening on port 36794
- New mrg.exe file in the Startup folder
- Modification of HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- No anti virus software running

Upon review of our Active Directory Services, we determined that the personal laptop was not a member of the domain. This in itself was a breach of security as a policy was in place requiring computers to be part of the domain. Thankfully, the user was cooperative and allowed us to further examine his computer. Additional analysis and interview with the user produced the following:

- Computer was running IE 5.01 SP1
- Computer was running Windows 98
- Computer had never been added to the domain
- User was logging into the computer using a local account on the computer
- User did not keep up with any of the security hot fixes on his computer
- User used hotmail, yahoo, and other public internet email services at home
- User periodically used this computer at other client locations and received outlook email at these locations
- The user had brought the computer earlier in the morning and had plugged it into the network
- DHCP broadcast had provided the computer with an ip address upon connection to the network
- The user had used his network credentials to authenticate to shares he needed. This credential had been cached and was not requested of him each time he wanted to connect to the same share.

The above finding scared the Incident Handling team as it reminded us that the policy of not operating a computer that is not part of the domain was not enforced technically. We wondered how many more machines could be following the same scenario. In addition, we understood that if a machine were not part of the domain, then we would have no idea that the machine existed. Because DHCP broadcasts ipaddresses without requiring authentication, it was not possible to track which computers were in the environment unless they were part of the domain. Once a computer was part of the domain, SMS would recognize it and begin pushing critical components such as corporate anti virus software. In addition, we would have knowledge of the system and could include it in security tests as needed.

At this point, the Incident Handling team was very sure that the Bugbear worm had infected the system and a journal of the information collected for the incident was created. The incident moved into the Containment phase.

Containment

The Help Desk team lead was immediately contacted and was advised of the events happening in case further reports were noticed. The bigger problem was the potential that other divisions had not kept up with proper security protection and that they were also infected. Since we were representing the client and were acting as an extension of them, the Incident Handling team decided to obtain approval from the client to contact all system administrators in other divisions to advise them of the events. The Incident Handling team determined that the two most obvious signatures for further infection now that an initial infection had been confirmed was as follows:

1. Random worm binary code printing
2. Email messages with 50,688 bytes attachments.

A conference call with the client security manager and all system administrators in the environment was arranged ASAP. The conversation was recorded for historical purposes. The Incident Handling team explained the events occurring and made the recommendation that all system administrators be on the lookout for a Bugbear infection that included specifically the two signatures listed above.

The Incident Handling team analyzed all of the information. Because we were confident that Bugear had infected the computer, a backup of the infected computer was immediately taken and the infected computer was disconnected from the network. The Incident Handling team also worked with the use to change all of his passwords used by the computer as a precaution.

Luckily, the Incident Handling team was already familiar with Bugbear's payload. Also, because we had technical controls in place such as blocking port 36794, we were comfortable in knowing that the most crucial payload was mitigated. In addition, all Exchange servers had been updated with the latest Group Shield anti virus signature file. Because our environment required each Outlook email to be sent to the Message Transfer Agent of the user's Exchange Server each time either an internal or external email was sent, we were confident that the anti virus software would catch any attempt by the worm to propagate. Therefore, we could concentrate on further payloads of the worm such as continued infection of users and other trusted networks/divisions through unprotected NetBIOS shares.

Three components of the incident had now been dealt with. First, the known infected computer had been removed from the network. Second, port 36794 was confirmed closed, thereby preventing any unknown infected systems from being controlled through a backdoor. Third, it was confirmed that the email servers were protected by an up to date anti virus signature file. The last payload that needed to be addressed was the propagation of Bugbear via NetBIOS shares. Although we had already alerted other divisions of what was going on, the Incident Handling team still needed to determine if further machines in the subnet affected were infected. If another machine were infected, it would attempt to propagate through NetBIOS shares. Therefore, we installed a sniffer on the subnet where the infection occurred.

Review of the sniffed traffic showed only normal activity and no increase of NetBIOS ports traffic was seen. This indicated that no further nodes were infected. Therefore, the probability that additional systems were infected was minimal.

As a further check, we contacted the Help Desk and asked if they had received any calls that may suggest infection of the worm. They had not. We also contacted the other division system administrators and were informed that they had confirmed that their machines were not infected.

All evidence collected indicated that this was an isolated infection and that no further infection had occurred. Still, we kept the sniffer operational and the mail server remained offline. The incident now entered the Eradication phase.

Eradication

After making a backup of the infected system and disconnecting from the network, the computer was shutdown normally. Because the infected system was never a part of the domain and because it was a workstation, a restoration from previous backup was not an option. The Incident Handling team did not think this was a problem as the worm could be eradicated manually by following the instruction of various Anti Virus companies. After worm eradication on the system, the system was patched and added to the domain. Adding the machine to the domain allowed the computer to obtain the standard workstation anti virus software with the latest signature file.

Because at the time of the attack, the Bugbear worm had been widely known, our environment had the luxury of knowing that our major critical components were protected. Further review of the email server showed no signs of infection and no additional patching was required.

Recovery

Although the infected system had been patched and all major critical components had been protected, we still needed a way to inventory all machines that may be using the network, but were not a member of the domain. If a machine was not part of the domain, there was a probability that the machine would not be running standard tools and that it could be vulnerable to various vulnerabilities. One solution would be to run a port scan on all subnets to determine live hosts. This list would then have to be compared to a list of domain members and manually sorted and organized to determine hosts that are not members. This is not a good solution as it would take a great deal of time and there is room for several errors to occur. One of these errors stems from the fact that this would be one snapshot of the current environment. By the time the comparisons were completed, machines may have been added, removed, or have obtained new ip address. Another solution would be to disable DHCP and assign static ip addresses. However, this would require huge overhead and resource allocation.

Based on the possible errors and time constraints, the Incident Handling team and the client managers determined that the solution would be to remind users of the policy and ask for their cooperation. An email would be sent to all users describing the policy mandating all computers to be part of the domain and requested that all users running computers that were not part of the domain contact the help desk immediately. An explanation of what was considered a member would also be included. Each division head was also asked to help out by actually sending the email to all of his employees and acting more like a driver of the process. Due to the possibility of overwhelming the help desk with calls associated with this email, a decision was made to send the emails to each division on an agreed upon schedule. This meant that not all users would be receiving the email at the same time and more importantly immediately after our discovery of the Bugbear infection.

Over the course of the next two weeks, a project was created to send out the policy email. Each division was given a schedule and date for when they needed to be compliant with the policy. Emails were sent out and users took action. The result was the discovery of several workstations and servers that had not been part of the domain. Both workstations and servers were added to the domain and were secured. There were some cases, specifically with servers, where some administrators did not want to add these to the domain. Situations like these were handled on a case-by-case basis by the administrator and client management. If a decision was made not to add the computer to the domain, the division or administrator responsible for that server agreed to take full responsibility of that computer and all of its actions. Although this was not a good solution from a security perspective, it did allow us to determine which machines would not follow the policy and track them with a tighter monitoring.

Lessons Learned

The Bugbear incident occurred due to a lack of enforcement of security policy. Although strong and up to date technical controls were in place, they were not sufficient to ensure 100% security protection. The main lesson learned is that technical controls are only a portion of complete security strategy. A complete strategy includes good security policy that is enforced both technically and procedurally. In this case, a policy was not being enforced technically or procedurally. Such a gap created the necessary environment for an infection by Bugbear to occur.

After the incident was mitigated, the Incident Handling team met to discuss our findings and lessons learned. It was agreed that the overall process was good, but that changes needed to occur. These changes included:

- Official membership by a senior client manager in particular the client Security Manager.
- Extension of the Incident Handling Process team to incorporate other divisions' personnel. For this to occur, the Incident Handling Process would have to be owned by the client security team. This would remove use from the position of driving the process during an incident that reached into other divisions. It was also agreed that current team could be kept functional, but that expectations and requirements would be documented and only reach current network support teams. As soon as an incident was believed to extend past the teams' immediate responsibilities, the client incident handling team would be called into action.

As far as changes from an operational perspective, several were learned and have been implemented. These include:

- Vulnerability testing of all workstations including laptops belonging to all members of the network support team. A recommendation has been given to the client that they work with other division to make sure that this process is also being done.
- Vulnerability testing of the whole subnet where network support teams services is provided. In the past, testing was done only on servers and was done by ip address. A recommendation has been given to the client that they work with other division to make sure that this process is also being done.
- Creation of a process requiring all servers and workstations to be tested from a security perspective before adding to the domain. This process is not only extended for servers and workstation that we administer, but also for computers that may be administered by others who need to join the domain.

Extras

The problem of not being able to technically force computers to become members of the domain is still one that we are trying to resolve. We continue looking for an economical technical solution. The workaround in place relies on the cooperation of users and the buy in from key personnel including other administrators and managers. The client has put the responsibility of ensuring that all computers are part of the domain on the division chiefs. They in turn have given this responsibility to their administrators. The result is fragile balance that could be broken by any user that may not want to adhere to this policy.

Summary

Every month there are new vulnerabilities discovered. As these are discovered, administrators race to mitigate and patch vulnerabilities hoping that an attacker does not beat them to the punch. In this rush, many times only technical controls are given sufficient attention and priority.

While it is crucial to be proactive and conduct technical assessments of security controls to prevent future incidents, it is just as important to consider which procedural controls are in place to ensure that that both policy and technical controls are being followed. An environment may have several technical controls in place, but if these do not have a strong process surrounding them, then the likelihood that these controls are evaded increases.

Another problem exists from the fact that many users are required or feel the need to work from home. If adequate resources are not provided to these users, they may be forced to take matters into their own hand. Sometimes a user who needs to work from home, feels the need to use his or her own computer. Unknowingly, by doing so, these users expose their organization's data to probable unsecured locations and situations. In addition, many of the security controls in place in an organization are not in place at a user's home. This increases the likelihood of an incident occurring due to discrepancies between home and corporate security.

References

- http://vil.nai.com/vil/content/v_99728.htm
- http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html
- http://www.mcafee.com/anti-virus/viruses/bugbear/
- http://www.sophos.com/virusinfo/analyses/w32bugbeara.html
- http://www.f-secure.com/bugbear/
- http://www.theregister.co.uk/content/56/27380.html
- http://vil.nai.com/vil/content/v_99436.htm
- http://news.com.com/2100-1001-960365.html?tag=lh