Global Information Assurance Certification Paper

SANS - GCIH V.2.1

Practical Examination: Option 1
Exploit in action

Responding to BugBear Worm

Russell Cluett CISSP
January 2003

Table of Contents:

Abstract

This practical assignment has been written as another step forward in the quest for GIAC Certified Incident Handler Certification (CGIH). Described herein are real events that I participated in during a BugBear virus outbreak in a global organization that I will refer to as “BigCompany”. Some information has been modified to protect the confidentiality of the organization but will not affect the content of the paper from an incident handling perspective. In an effort to remain vendor-neutral I have intentionally not disclosed any vendors names or the names of their products. Although not mentioned in the content of this paper, all testing described herein was performed in an isolated lab that was separate from the production environment (air gapped) and all changes that were made to the environment followed standard testing and change management procedures.

Part One – The Exploit

Background

On September 29th, 2002 BigCompany started to be impacted by the BugBear virus. Although the virus payload seemed relatively non-destructive it attempted to terminate Anti Virus applications, consumed network resources and with it’s backdoor and keystroke logging capabilities it had the potential to compromise security. To an end user there were no visible signs that their workstation was infected. Identification of infected nodes, containment and eradication proved to be complex and this challenge was compounded by the intricacies of trust relationships between the numerous domains. This paper will discuss how the Canadian Domain of BigCompany was affected and the measures that were put in place to identify, contain, eradicate and recover from infected nodes globally.

Name

Anti Virus software vendors have difficulty in developing or following a standard naming convention resulting in a variety of names for the same virus. For this reason there is a resource available to search for virus and their names at http://www.virusbtn.com/resources/vgrep/index.xml

The results of a Vgrep search for BugBear turned up these varied results.

AV Vendor	Virus Name
Trend	WORM_BUGBEAR.A
Sophos	W32/Bugbear-A
Symantec	W32.Bugbear@mm
Computer Associates	Win32/Bugbear.Worm
McAfee	W32/Bugbear@MM

This virus is also know by a variety of other names such as NATOSTA.A, and Tanatos,

For the purposes of this paper I will simply refer to this malware as BugBear.

The vulnerability that BugBear exploits is known as "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment" and is listed at Common Vulnerabilities and Exposures as CVE-2001-0154
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154

Operating Systems

The machines that are capable of being infected are Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows ME

Machines that are not affected are Macintosh.

Linux print servers were affected in that Windows machines were sending Bugbear generated print jobs to Linux based print queues and these were causing printers to print huge documents of garbled text with sizes equal to the size of the worm.

I cannot say for certain if Unix print servers would be affected or not, but it follows to reason that if they have print queues available for Windows clients then they too would be affected.

Protocols/Applications

The Protocols used to execute the exploit are: Simple Mail Transfer Protocol (SMTP); Server Message Block (SMB) and Netbios.

The Applications Exploited are: Microsoft Outlook, Microsoft Outlook Express, and Internet Explorer.

Brief Description

BugBear is a mass-mailing worm that is written in Visual C++6 and is compressed with the Ultimate Packer for eXecutables (UPX). It targets Windows computers and uses it's own SMTP engine as well as unprotected network shares to propagate. It also attempts to terminate Anti Virus scanners, has a remote backdoor and a key logger that is believed to be used to harvest usernames/password combinations.

A side effect of the method it uses to propagate via network shares is that it copies itself in raw binary data to print queues and floods network printers.

As per MessageLabs, BugBear was the third most active virus for the year 2002, only surpassed by Klez and Yaha
See: MessagesLabs – Press Release 16 Dec 2002
http://www.messagelabs.com/viewNewsPR.asp?id=113&cmd=PR

Variants

There are no known variants of the BugBear worm however snippets of code have been recycled from BugBear and used in new virii, and the offer of supposed BugBear "solutions" has been used as a social engineering ploy to distribute other virii such as WORM_HOBBIT and VBS_QUOCUS.

References

Some helpful websites for information regarding this virus can be found at:

Virus Bulletin:
http://www.virusbtn.com/resources/viruses/bugbear.xml

Symantec
http://www.sarc.com/avcenter/venc/data/w32.bugbear@mm.html

Trend Micro
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BUGBEAR.A

MessageLabs:
http://www.messagelabs.com/viruseye/

Network Associates:
http://vil.nai.com/vil/content/v_99728.htm

Sophos:
http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

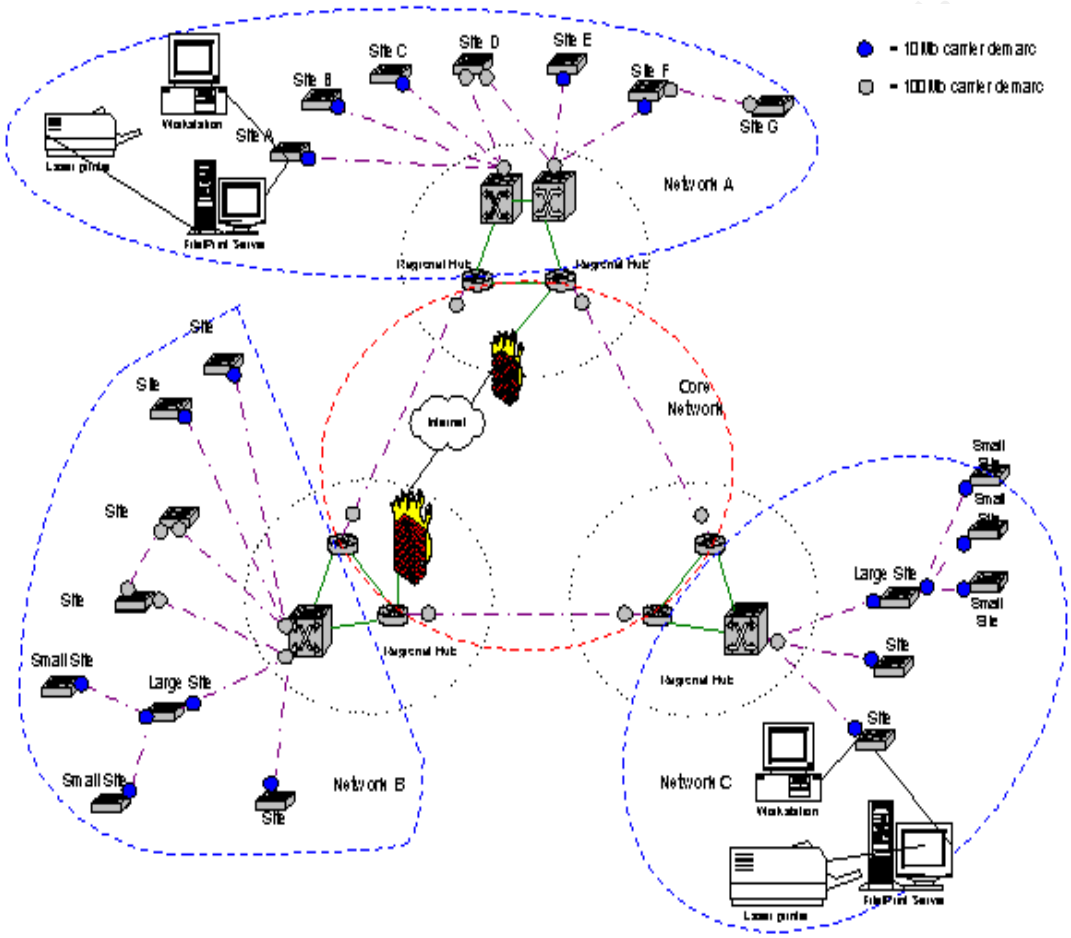
The vulnerability that is exploited by the email propagation method:
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0154

The Microsoft security bulletin for the vulnerability exploited is at:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

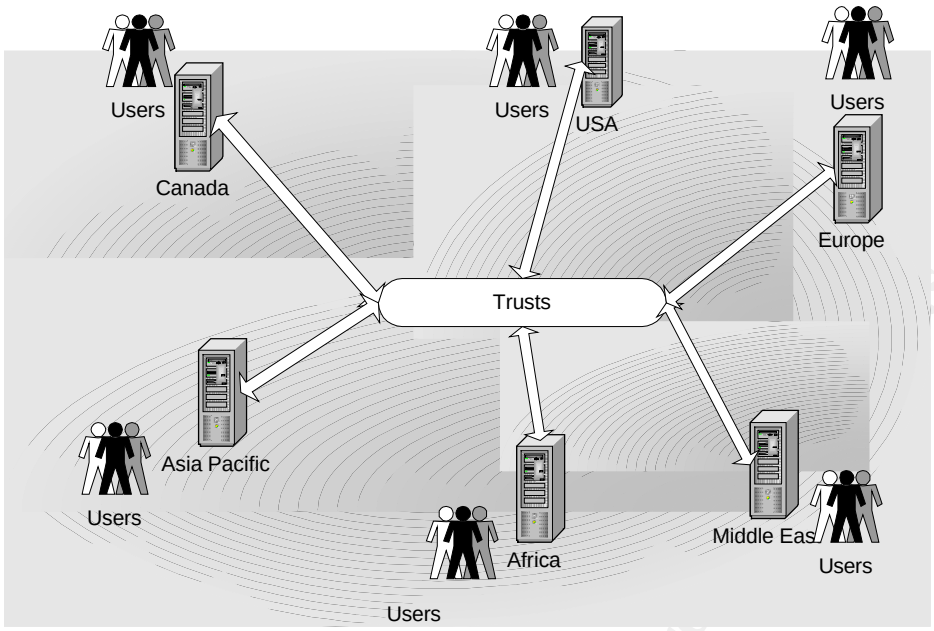
Part Two – The Attack

Description and Diagram of Network

Because of the numerous complexities involved I have only included a basic overview in the diagrams below. The BigCompany network itself spans the Globe and is made up of a variety of large, medium and small offices as well as campus sites and remote users.

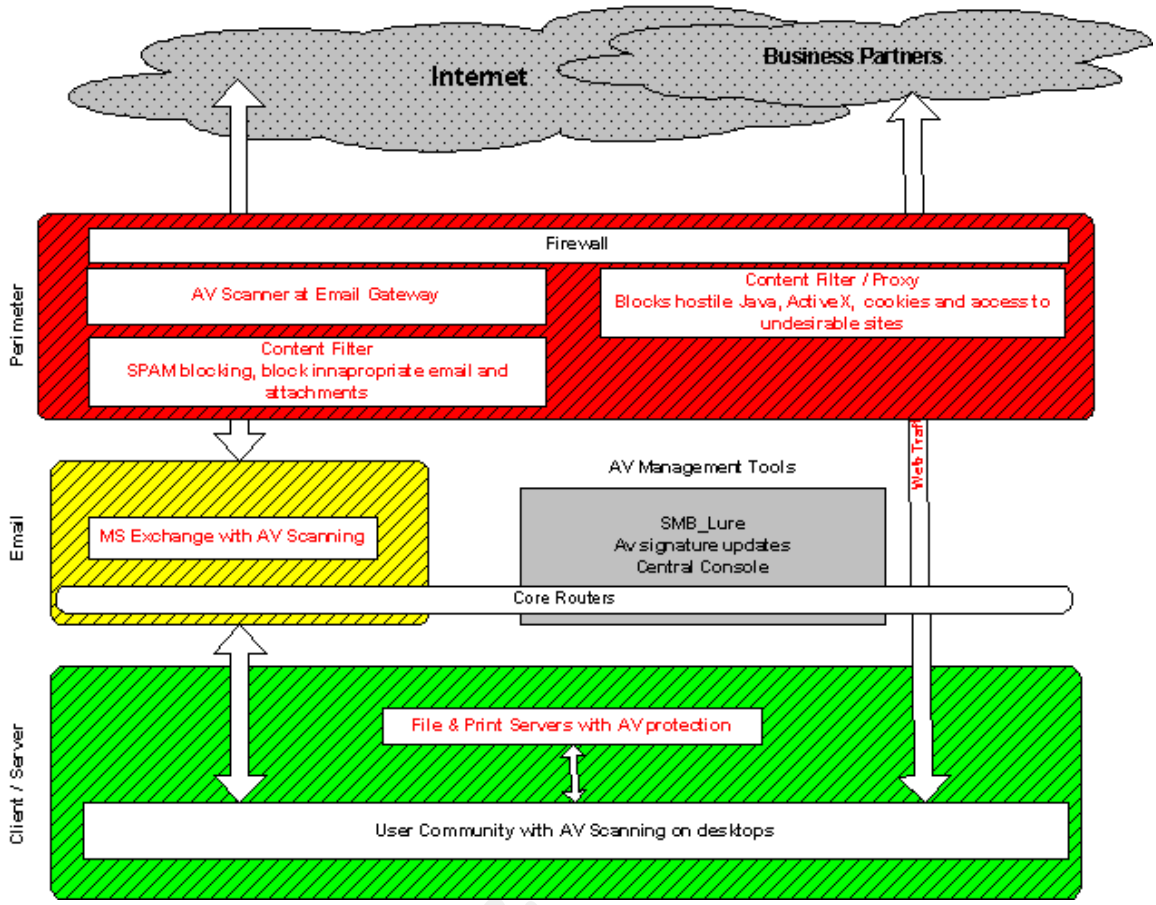


BigCompany is a global organization using Microsoft Windows NT and Windows 2000 Servers in a Multiple Master Domain model with a large number of trusts in place as shown in the simplistic graphic below. Network Resources reside in each domain, as do user accounts. Permissions are granted using the AGLP model, user Accounts are added to Global Groups that are in turn added to Local groups, which then are given Permissions to resources.



The user community consists of over 100,000 users in a Windows environment with mainly Windows NT and 2000 desktops, but also a large contingency of legacy Windows 9x machines. Peer to peer networking is not permitted, nor is it enabled on the default corporate workstation builds. There are however some rogue workstations that are occasionally discovered and disconnected from BigCompanies network. Due to some variance in versions of Internet Explorer at the time (since remedied) it was difficult to be certain of patch levels.

This enterprise incorporates people, process and technology that support a layered approach to comprehensive Virus Defence that is graphically depicted as follows:



Protocol Description

The protocols that the BugBear virus uses to propagate are Simple Mail Transfer Protocol (SMTP), Server Message Block (SMB), and NetBios.

SMTP:
The main purpose of SMTP is as a mail delivery protocol that will relay mail between hosts on different transport systems. It is used for sending email from a mail client to a mail server, and between mail servers.

SMB:
A protocol used for sharing resources between computers. It is a request – response protocol, thus once a connection is established (over TCP, NetBEUI or IPX/SPX) clients can send requests to servers to access/modify files, use resources, etc. This is the protocol used by the SMB_Lure, which we will get into more detail about later.

NetBios:
This is the basis for Microsoft Networking that is used over TCP or UDP for name management, session management and data transfer. It is based on broadcast

traffic to register and resolve host names as well as enumerating network resources.

How the Exploit Works

When Bugbear is initially executed it will first copy itself as %system%\????.exe, where %system% is the location of the operating system files (i.e. C:\Windows or C:\Winnt) and ? represents variable letters that are chosen by the worm.

It then drops files containing a password for the backdoor component and the keystroke logger in three encrypted .dll files in the %system% folder and two encrypted dat files in the windir% folder.

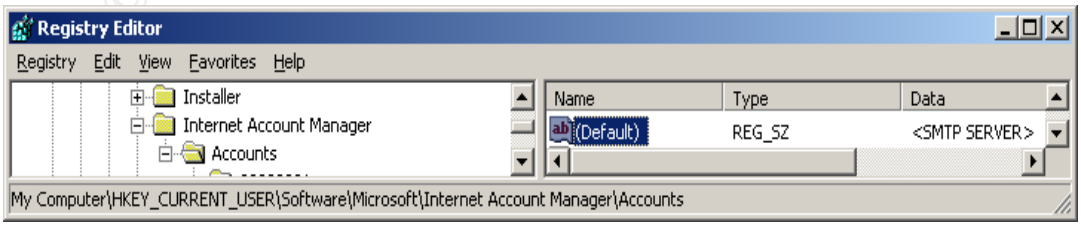
It then creates an entry in the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce equal to it's own value so that it starts each time that you start Windows

It then creates four threads, the first of which will use different routines based on the type of operating system to attempt every thirty minutes to stop a variety of processes that are specific to most major vendors Anti Virus products and Personal Firewalls.

The next (second) thread searches for email addresses in the current mailbox as well as addresses from cached email messages and address books on the victims' machine that it harvests from files with the following sub strings:

- ODS Microsoft Outlook Express Mailbox
- Inbox
- MMF Microsoft Mail File
- NCH Outlook Express Folder File
- MBX Mailbox Message File (Outlook v1-4 or Eudora and others)
- EML Microsoft Outlook Express Electronic Mail
- TBB The Bat! E-mail Hives
- DBX Microsoft Outlook Express E-mail Folder

BugBear then reads the local registry key depicted below



to identify if there is an SMTP server it can use to propagate. If there is no value set the worm will still propagate via two of three SMTP engines that the worm itself carries.

One engine will send an encoded version of the worm in a plain email message with a content type of application/x-msdownload.

The other engine will send the email with a content of audio/x-MIDI and will format the message to be HTML and contain the code that will exploit the Incorrect MIME header vulnerability (CVE-2001-0154). This exploit is possible because HTML Emails are treated by Outlook as web pages, allowing Internet Explorer to render them and open attachments according to their MIME types. If an HTML email contains an executable with an incorrectly given MIME type (of which there are several with flaws) then Internet Explorer will cause the attachment to be executed without warning when a user opens or previews the email in an unpatched version of Internet Explorer.

To avoid detection BugBear will not send itself to the current user of an infected system. The method used to accomplish this is it checks the following registry key for the address it should not send to.

> HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Account Manager\
> Accounts\%Default Mail Account%
> "SMTP Email Address"

The mass mailing routine of this worm will send itself with the viral attachment in emails to 170 of the addresses it finds. This is as a reply to or forward of an existing email on the system or as a new email with no message body and a subject line picked randomly from a variety of possible subjects as detailed below from Trend Micro at http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BUGBEAR.A

- $150 FREE Bonus!
- 25 merchants and rising
- Announcement
- bad news
- CALL FOR INFORMATION!
- click on this!
- Confirmation of Recipes…
- Correction of errors
- Daily Email Reminder
- empty account
- fantastic
- free shipping!

- Get 8 FREE issues - no risk!
- Get a FREE gift!
- Greets!
- hello!
- history screen
- hmm..
- I need help about script!!!
- Interesting...
- Introduction
- its easy
- Just a reminder
- Lost & Found
- Market Update Report
- Membership Confirmation
- My eBay ads
- New bonus in your cash account
- New Contests
- new reading
- Payment notices
- Please Help...
- Report
- SCAM alert!!!
- Sponsors needed
- Stats
- Today Only
- Tools For Your Online Business
- update
- various
- Warning!
- Your Gift
- Your News Alert

BugBear will read the contents of files in the Personal folders and may use file names retrieved from there to compose a file name for the viral attachment to the email, but will not use *.ini files as possible attachment names. BugBear may also create a filename consisting of one of the following words:

- Readme
- Setup,
- Card
- Docs
- News
- Image
- Images
- Pics
- Resume
- Photo

- Video
- Music
- Song
- Data

BugBear will coose an extension for the attachment from .scr, .pif or .exe and can append extensions SCR, or PIF which results in attachments with double extensions.

The mass mailing routine of Bugbear will 'spoof' the return address field in the "From" email address in the emails it sends out. This makes it difficult to trace the original sender (infected node) and possibly tricks the recipient into thinking it is someone they know and trust. The spoofed "From" address is displayed as being from the actual user being logged on to the machine at the time. This is taken from the registry key

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Account Manager\Accounts\%Default Mail Account%
"SMTP Display Name"

The From address is thus displayed as being the actual user of the infected system but the actual underlying address used in the From field is taken from the addresses that the worm has found by searching the local system for email addresses.

The third thread BugBear creates will set up a password protected backdoor on the infected machine that listens on port 36794 for commands from an attacker that will allow him/her to remotely connect to it and access, display download and execute files; list, start and stop processes; access detailed system and network information; and even start an http server that allows the attacker to remotely browse through the infected computer with a web browser. Because of the complexities of sending instructions to the backdoor server it is believed that a client program also exists for this worm

If an attacker attempts to connect to the backdoor on port 36794 two temporary files will be created in the Windows temporary folder:

- ~PHGGUM.TMP
- ~EAYLNLF.TMP

The temporary file, ~PHGGUM.TMP, contains a 20-character string that is required to be used as a session ID to communicate with a client.

Cached passwords, the machine name and the currently logged on UserID are accessed through the Mpr.dll file on Windows 95/98 and ME machines through

system calls and sent out by email. The subject line of the email is the domain name from the “SMTP Default Address” that has been taken from the registry. The sender and receiver of the email are the same and are possibly any of the following:

- boxhill@teach.com
- brdlhow@ml1.net
- c.willoughby@myrealbox.com
- erisillen@canada.com
- gili_zbl@yahoo.com
- jacopo58@excite.com
- jwwatson@excite.com
- langobaden@excite.com
- mannchris@gala.net
- mshaw@hispostbox.com
- rvre2736@fairesuivre.com
- rwilson@singmail.com
- sc4579@excite.com
- sctanner@myrealbox.com
- sdsdfsf@callme.as
- sergio52@mac.com
- sm2001@mail.gerant.com
- stevechurchis@excite.com
- stickly@login.pe.kr
- t435556@email.it
- vique@aggies.org
- zr376q@yahoo.com

The three DLL files and the two DAT files that were installed when the worm first ran are then used to capture keystrokes that are stored and could be later retrieved via the backdoor component. This Keystroke logger Bugbear worm installs is identified by most AV vendors as PWS.Hooker.Trojan.

The fourth thread scans for shared network resources. When it finds them it attempts to copy itself to the remote nodes %Startup% folder using a random filename with a .exe extension.
Because it copies itself indiscriminately to any available network resource it floods network printer queues and has printers trying to print out copies of the worms code. Interestingly enough, although this worm targets Microsoft systems this time Linux print servers were also affected.
This behaviour turned out to be instrumental in identifying infected network nodes.

Description and diagram of the attack

The first copy of this mass-mailing worm was identified in the wild on September 29, 2002 from Malaysia. There are no significant symptoms visible on an infected node and it spread around the world very quickly, becoming one of the most prevalent virii of 2002. The rapid speed at which BugBear spread is attested to in this quote from ZDNet News, - *Bugbear to set new virus record, By Robert Lemos and Matthew Broersma - October 8, 2002, 5:12 AM PT*
http://zdnet.com.com/2100-1105-961130.html

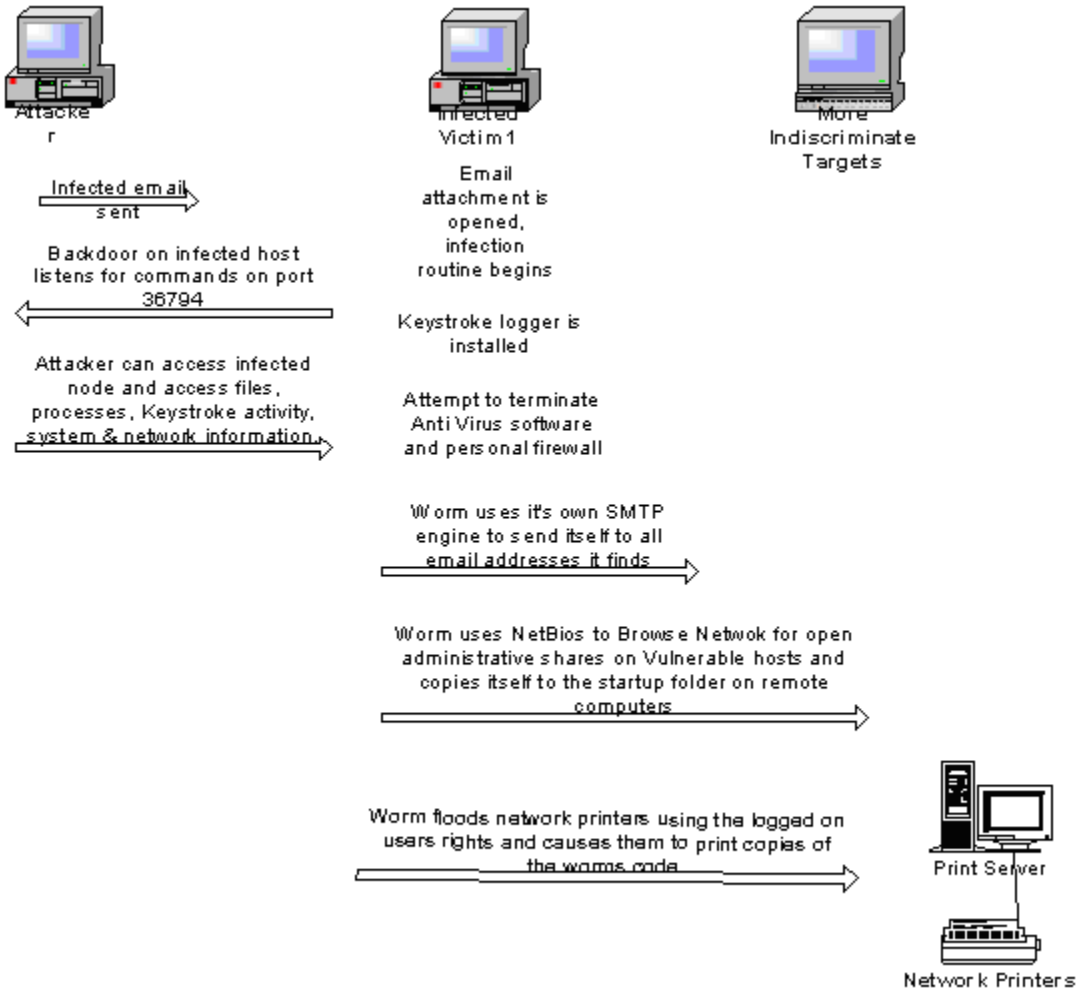
"Last week, e-mail service provider MessageLabs intercepted 320,000 missives containing the Bugbear attachment, more than the Klez.h virus managed in its first week in April. Klez.h has created the most-ever Internet traffic so far."

BugBear entered BigCompany and gained a foothold in the environment prior to Anti Virus signature files being available to defend against it. Although the initial entry point is unknown we do know that it did not enter through SMTP relays, and that it's entry point had bypassed the perimeter defence that would have otherwise prevented it. It is believed to have entered via an Australian user who circumvented perimeter defence and accessed web based email through an Internet Service Provider. Once inside the network this virus quickly and actively searched for and found some vulnerable rogue computers with open administrative shares that it was successful in infecting.

Its payload then began attempting to deliver the following five threads:

- Mass mailing
- Network aware infector
- Backdoor server
- Terminating Anti Virus and Personal Firewall software
- Keystroke logger emailing out passwords to specific email addresses.

The diagram below shows at a high level how the infection/propagation routine takes place.



Propagation

In BigCompany this worm attempted propagation in two ways.

- Mass emailing an attachment compressed with a hacked version of UPX.
- Enumerating network resources (Netbios / SMB) and copying itself to the startup folder of remote computers with open administrative shares.

Signature of the attack

On the infected node:

- Port 36794 TCP open
- Existence of 50,688 or 50,684 bytes <randomName>.exe files in the %WinDir%\System\ directory
- three encrypted .dll files in the %system% directory
- two encrypted .dat files in the %windir% directory.

- A value equal to the worms filename added to the registry key HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
- If an attacker has attempted to connect to the backdoor there will be two temporary files in the Windows temporary folder:

    - ~PHGGUM.TMP
    - ~EAYLNLF.TMP

On the network:

- Network printers flooded with print jobs of approximately 500 pages beginning with some garbled characters followed by the text "=!This program cannot be run in DOS mode".
- Network traffic enumerating shares trying to find **"$C"**
- SMTP email traffic with 50,688 byte attachments with double extensions ending in .exe; .scr; or .pif.

How to protect against it

Although I will discuss the use of defence in depth further in the section of this paper entitled "Preparation" some specific items that would protect against BugBear are:

User Training:

- Educate users not to open unsolicited email attachments.
- Ongoing communications with the user community about new threats and safe email practices.

On hosts:

- User rights: lock down workstation builds so users cannot install software or modify system files.
- System configuration: Do not allow systems that have unprotected Network shares or world write able drives.
- Keep security patches up to date. In this case Internet Explorer 5.01 Service Pack 1 or Internet Explorer 5.5 Service Pack 1 would require the Microsoft patch q290108 to prevent the viral attachment from executing in a preview pane.
- Maintain current and properly configured Anti Virus software.

On the network:

- Block all unneeded traffic at the perimeter, in this case specifically TCP and UDP ports 137, 139, 25 and 36794
- Configure your mail server to strip off executable code.

- Up to date and properly configured Anti Virus software on SMTP Relay nodes, Email, File & Print Servers.

N.B. In this case protection dependant on up to date Anti Virus signature files would not have been effective. Due to the reactive nature of Antivirus software, protection against this newly discovered virus was not included in the vendors latest signature files.

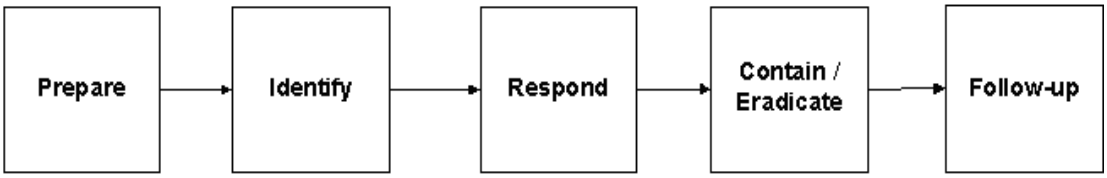
Part Three- The Incident Handling Process

Preparation

BigCompany executives take security very seriously. BigCompany has clearly defined the business and privacy requirements and with those in mind developed and published security policies and supporting standards that are well communicated. This is implemented and managed through a security program that considers the people, process and technology in all aspects of the organization including system lifecycle. This is complemented through a validation process to ensure compliance. I will discuss the people, process and technology of some of the layers of defence that BigCompany uses to prepare for, and protect against attacks.

BigCompany has a 7x24 Incident response capability made up of a central team with global representation from regional groups. This team monitors new threats through a variety of internal and external means. Once notified of a threat or an event there is a verification process and an initial assessment is performed by the CIRT analysis team of the situation before determining that an incident is taking place. If there is an incident then this team will identify the assets impacted, and the severity, urgency and status of that impact. Next, asset protection priorities are identified, as are contacts and resources. At this time the rest of the CIRT team members are called together for an initial meeting where details of the incident are discussed; mitigation, containment, eradication and recovery strategies are determined; tasks and responsibilities are distributed; and communication timeframes are established.
There is also an escalation process that takes place for public relations, media, legal, and investigation and security teams. This may include law enforcement as required.
At the conclusion of incidents there is a close-down process in which a report is created detailing all aspects and analysis of the incident, that is reviewed by the CIRT team and management to identify any what further countermeasures or safeguards that should be implemented to prevent recurrence.

This process is depicted in the following graphic:



BigCompany has ongoing mandatory security awareness education and regular communications to the user community sensitizing them to the virus threat.

Perimeter:
Firewall and Proxy servers provide IP and URL filtration for known malicious and undesirable websites. Firewalls prevent any traffic from entering or leaving BigCompanies network that has not been approved through a comprehensive change request process including risk mitigation and analysis.

Internet email Gateway defence consists of Anti-Virus software on the internet relay nodes with real time scanning in both directions (incoming and outgoing), Anti-virus scanning of e-mails and attachments; file type filtering stripping common file types known to execute malicious code; content filtering for active content and SPAM filters.

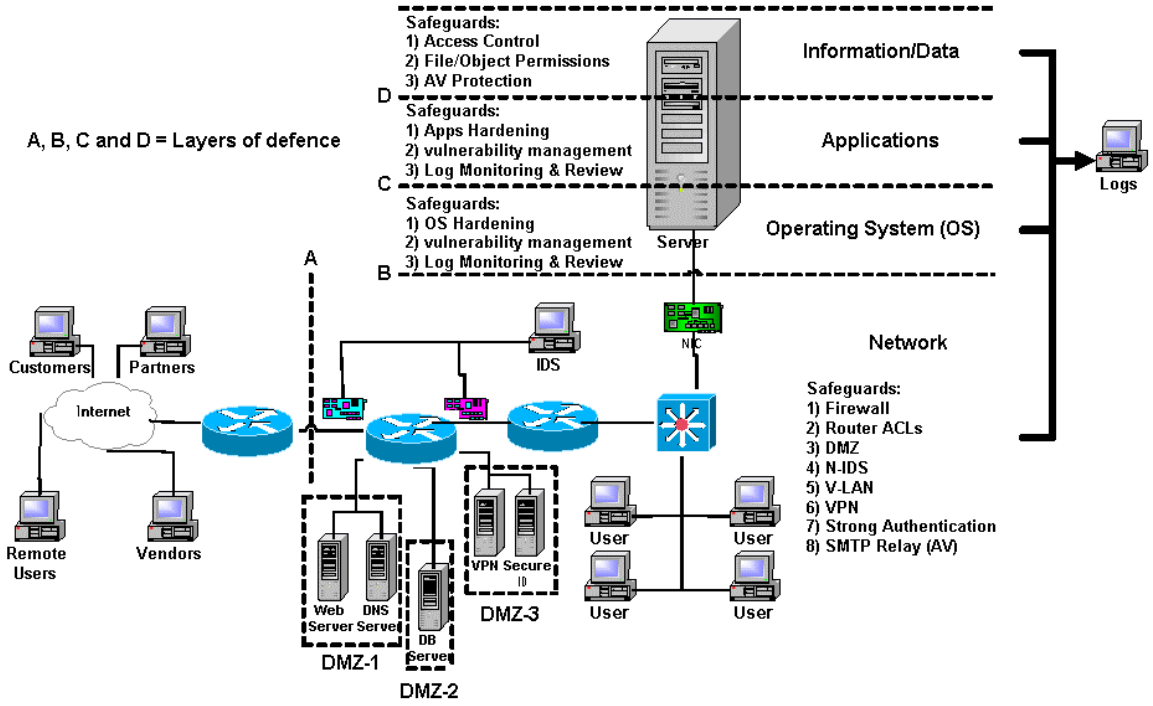
Email:
The Microsoft Exchange email servers are protected with anti-virus software with automated signature file updates and real time scanning in both directions, Anti-virus scanning of internal e-mails and attachments, and attachment and/or subject line filtering capability. BigCompany has the ability for message size restrictions and mailbox or server lockdown. The Global Address List (GAL) first entries are dummies configured with pager notification should they be sent to in the event a mass mailer uses it for distribution purposes.

Realtime AV Scanning protects internal systems on File & Print servers and desktops. Signature files are kept up to date with automated signature deployment and central management.

BigCompany has a vulnerability management tracking system to ensure vulnerability alert distribution to the people responsible for systems and provide security patch/fix implementation accountability. This provides real time patch/fix implementation status and ensures escalation when reasonable implementation times and compliance are not met. It also allows BigCompany to be able to tell how much exposure to risk exists relative to any unpatched systems.

BigCompany has restricted and minimized access rights and eliminated unnecessary services. Although the use of shared drives and peer-to-peer networking on workstations is prohibited, it is occasionally discovered on “rogue” workstations through the validation process.

A graphic depiction of the different layers of defence can be seen in this diagram:



Identification

The BugBear virus symptoms were first identified internally on September 30, 2002 at 11:01 PM when a user called in to the BigCompany help desk in Canada reporting that someone has sent a print job to a server that has several printers on it. The user said that he cannot stop the jobs on the printers at his location and the help desk analyst could not either. The print job was all garbage and was emptying the print tray due to its size. The NT server operator at this location stopped 13 jobs on 13 printers. This was initially misdiagnosed as a print job that was not configured properly.

Forty minutes later it was realized that this matter is much bigger than the one site. Printers all over Canada are printing the same document. The Canadian Support team identified that this was a symptom of the newly discovered W32/BugBear@mm virus. Identifying infected users was a matter of matching the UserIDs from the submitted print jobs to the users; they were identified as being from Australia, Germany and Italy, who were starting their workday as most Canadians has ended theirs. The Global Incident Response Team was engaged.

Containment

Several containment measures (some attempts were successful and others were not) took place simultaneously around the globe and the order in which they are listed should not be interpreted as their order of importance. For clarity purposes I will group containment measures based on the different vectors of the virus behaviour; 1) propagation by email, 2) propagation by network, 3) spawning print job payload and 4) the backdoor/key logging payload.

1. Propagation by email

At the Internet email gateways several steps were taken to prevent the virus from further propagation. The messaging team verified that in fact the extension blocking was effective to prevent further instances of the virus from entering the environment. New signature files were deployed on the SMTP Relay nodes (inbound and outbound email) and the Exchange servers (internal email distribution). This proved to be effective and there were no instances of the virus propagating through the email system,

There was still the risk that it could be imported through a user accessing webmail, infecting him/herself and allowing it's further attempts at propagation through the network. In an effort to reduce the number of users that might fall prey to the social engineering aspects of this malware a user communication was sent by email to all employees as follows:

> Subject: Bugbear Virus Alert
>
> A medium risk virus has been detected within the BigCompany Network causing large volumes of undecipherable print on network printers. We are working to get the latest signature files deployed and to ensure that infected users are disconnected from the network. If you suspect you have been infected, contact the help desk @ <telephone_number_sanitized>. Please stand by for more information.

As a side note, the gateway scanners were previously configured to send a notice to the sender of an email when it was identified as viral. In one case the scanner replied to the spoofed "From" addresses and this was a group of people (distribution list) rather than an individual. This was identified this as a Bad Thing ™ and this feature was disabled.

At the desktop level the Microsoft vulnerabilitie that was being exploited by BugBear was the "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment" (MS01-020), which affects clients using Microsoft Internet Explorer 5.01 & Microsoft Internet Explorer 5.5 and allows an email attachment to be executed by opening and email or viewing it in the

preview pane. Although there had recently been a directive for all users to upgrade the Internet Explorer web browsers to a recent and fully patched release, data collected on connected PC's showed a 50% compliance rate of upgrades to IE 6.0. Another communication was sent out directing users that they were required to upgrade to Internet Explorer 6.0. and included the instructions for connecting to the distribution server nearest to their location and downloading the installation package.

New Anti-Virus signature files that detected this virus were tested and distributed to desktops promptly as available.

2. Propagation by Network

BugBear attempts to copy itself across the network by enumerating network shares and writing itself to the startup folder of those it finds. Because it doesn't do this properly it will also write itself indiscriminately to any network resource available (i.e. printers). In order to contain the spread of this virus we had to identify and isolate the infected nodes that were exhibiting this behaviour. The tool used to do so is referred to as an SMB_Lure, a Linux server running Samba in debug mode to facilitate extensive logging. John Morris (from AVIEN) pioneered this concept and more details are available at http://morris.dnsalias.com/smb-lure.htm

I had previously built an SMB_Lure and positioned it to represent nine virtual machines emulating Windows computers with a variety of open 0 byte file shares found in their own domain that appeared at the top of the browse list in Network Neighbourhood, making them appear as the first nodes that would be enumerated. A separate Samba log file was kept for each machine to visit this virus "honeypot" and the log files were parsed hourly. Through communication with other AVIEN members it was determined that searching the Samba logs for the string "couldn't find service $c" would effectively identify nodes that were trying to propagate BugBear. I had a script for parsing the Samba logs, originally written by Paul Schmehl (http://www.utdallas.edu/~pauls/checklogs.html) that I modified for this purpose as below:

```
sambalogs=/usr/local/samba/logs/*.log
alerts=/home/alert.log
touch $alerts

for log in $sambalogs;
do
  chmod 770 $log
  if [ -f $log ]; then
    counter=0

    bugbear=`cat $log | grep -ci "couldn't find service $c"`
    if [ $? == 0 ]; then
      echo "BugBear hits = $bugbear." >> $alerts
      counter=`expr $counter + 1`
```

```
fi

      counter=`expr $counter + 1`
    fi
    if [ $counter -gt 0 ]; then
      logname=$log
      echo `basename $logname` >> $alerts
      hostname=`basename $logname .log`
      echo $hostname >> $alerts
      IP=`cat $log | grep -e "$hostname " | cut -d'(' -f2 | cut -d')' -
f1 |
sort -u`
      echo $IP >> $alerts
      user=`cat $log | grep "sesssetupX:name=" | cut -d'[' -f2 | cut -
d']'
-f1 | tail -n1`
      echo $user >> $alerts
      echo "" >> $alerts
    fi
  fi
done

# mail the alert.log if there's anything in it and
# move the samba logs to the backup directory

if [ -s $alerts ]; then
  mail -s "SMB Lure Logs" Russell.Cluett@BigCompany.com < $alerts
  for oldlogs in $sambalogs
  do
    mv -f $oldlogs /usr/local/samba/logs/backup/
  done
fi

# do some "maintenance"
 rm -f /usr/local/samba/logs/*.log

chmod 770 /usr/local/samba/logs/backup/*
rm -f $alerts
```

This string was added to the log parsing script that harvests the IP address, hostname and the UserID for the user logged on at the time as displayed in the following log sample that was abbreviated for clarity:

Allowed connection from **(<IPAddress>**)

[2002/10/01 12:40:20, 3] smbd/password.c:authorise_login(854)

authorise_login: ACCEPTED: guest account and guest ok (nobody)

Domain=[**MachineNAme**] NativeOS=[Windows 2000 2195]
NativeLanMan=[Windows 2000 5.0]
[2003/01/15 12:40:20, 3] smbd/reply.c:reply_sesssetup_and_X(868)

sesssetupX:name=[**UserID**]
[2002/10/01 12:40:20, 2] smbd/reply.c:reply_sesssetup_and_X(985)

Defaulting to Lanman password for [**UserID**]

[2002/10/01 12:40:20, 3] smbd/reply.c:reply_sesssetup_and_X(1042)

Registered username nobody for guest access
[2002/10/01 12:40:20, 3] smbd/process.c:chain_reply(1023)

Chained message

[2002/10/01 12:40:20, 3] smbd/process.c:switch_message(685)

switch message SMBtconX (pid 4769)

[2002/10/01 12:40:20, 3] smbd/sec_ctx.c:set_sec_ctx(328)

setting sec ctx (0, 0) - sec_ctx_stack_ndx = 0

[2002/10/01 12:40:20, 3] smbd/service.c:find_service(140)

checking for home directory $c gave (NULL)

[2002/10/01 12:40:20, 3] smbd/service.c:find_service(209)

find_service() **failed to find service $c**

[2002/10/01 12:40:20, 0] smbd/service.c:make_connection(251)

[MachineName] (130.175.158.60) couldn't find service $c

[2002/10/01 12:40:20, 3] smbd/error.c:error_packet(94)

error string = No such file or directory

[2002/10/01 12:40:20, 3] smbd/error.c:error_packet(109)

error packet at smbd/reply.c(164) cmd=117 (SMBtconX)
NT_STATUS_BAD_NETWORK_NAME

This information was sent to me in the format below and copied to the Global Response Team for dissemination to regional teams to remove the infected nodes from the network.

```
-----Original Message-----
```

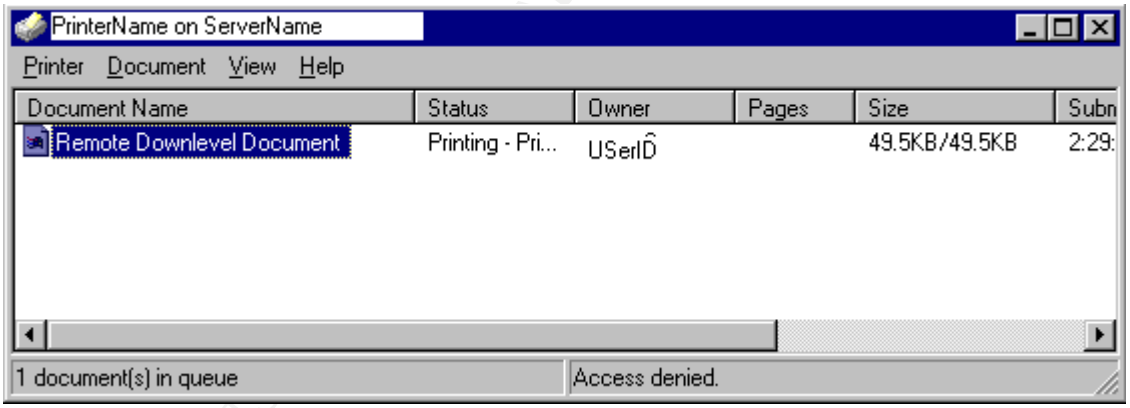
```
From: root [mailto:root@SMBLure.BigCompany.com]
Sent: October 01, 2002 1:30 PM
To: Russell.Cluett@BigCompany.com
Subject: SMB Lure Logs


BugBear hits = X+1 (incremental)
MachineName.log  (Name of Samba log file for infected machine)
MachineName      (Computer name of infected machine)
192.168.0.1      (IP Address of infected machine)
JoeUser          (UserID of infected user)
```

This technique has proved useful in identifying nodes around the globe that are infected with all network aware worms that enumerate network resources.

3. Spawning Print jobs.

There were some infected users around the globe who were unknowingly sending huge print jobs and jamming up network printers. The UserIDs from the print job submissions were harvested as in this bitmap below and communicated to the Global response team for dissemination to regional teams to remove the infected nodes from the network. Until such a time as the users were removed from the network the user accounts were disabled.



Because these print jobs were coming from around the globe it was suggested that BigCompany might be able to stop the print jobs inbound to Canada by deploying simple ACL's on the core routers which link Canada to the rest of the BigCompany by blocking the LPD service (udp/tcp 512) and possibly the Network Print Protocol (tcp/udp 92) traffic.

npp	92/tcp	Network Printing Protocol
npp	92/udp	Network Printing Protocol
printer	515/tcp	spooler
printer	515/udp	spooler

This was determined not to be an effective option as the print jobs were being sent between the client and the server over NetBios (udp/tcp 137-9) and from the server to the printer on ports 92 or 515 and for business reasons the NetBios traffic could not blocked.

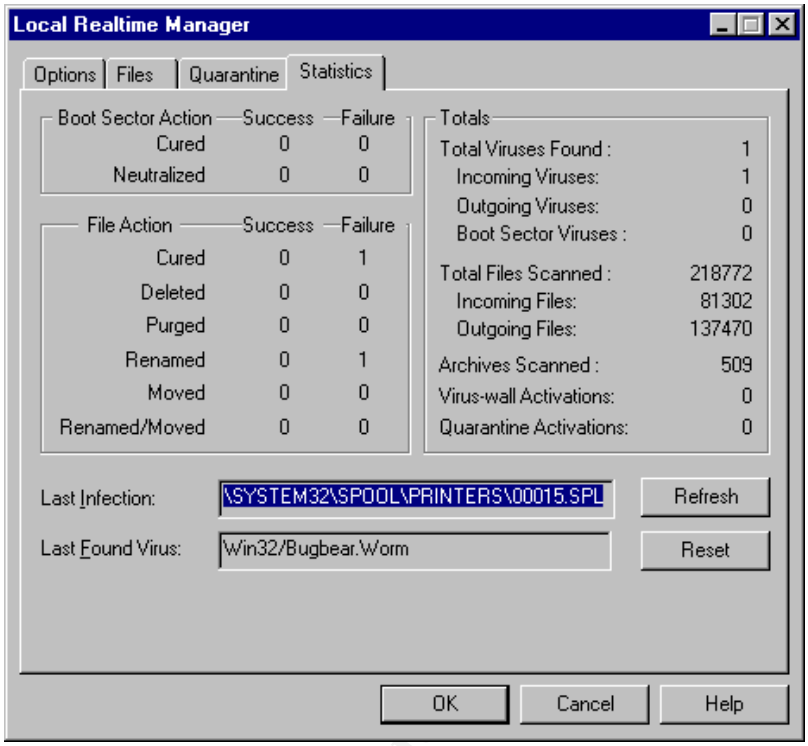
Updated signature files were distributed to the file & print servers but this did not prevent the print jobs from being processed by the spoolers on the print servers, although the same signatures were effective elsewhere. Originally the real-time monitor on print servers had only been configured to scan “incoming” files for performance reasons. It was verified that the Real-Time monitors were configured to scan all incoming files and the print queues were not in the exception list.

This didn’t make sense to me so in an isolated lab I set up a print queue for a dummy printer on an NT workstation and shared it out. The workstation had a standard build of AntiVirus installed on it, including scanning for incoming and outgoing files.

When BugBear was sent to the print queue the AV log files showed the following:

The Win32/Bugbear.Worm virus was detected in C:\WINNT\SYSTEM32\SPOOL\PRINTERS\00015.SPL. Machine: TestBox. User: TestUser, Action: Cure Failed.

The virus was detected in incoming files so the print servers should have been protected.



I still didn't understand why the print servers in Canada weren't detecting BugBear but after considerable amount of research it was identified that an “Incoming” scan is only performed after a file is closed or flushed to disk and an Outgoing scan is performed before a file is opened or executed (such as a print job). The outgoing scan provides the anti-virus protection that will prevent an infected file from being opened or executed on a local machine or when connected though the network. Thus the Anti Virus software from our vendor would only identify BugBear if “Outgoing” scanning were configured for the Real-Time monitor. This still didn’t seem logical to me but in light of this discovery, the print servers were configured to scan in both directions and were effective at eliminating the bugbear print jobs from being processed.

4. Backdoor/key logging payload

BugBear installs a keylogger and a backdoor component that listens on port 36794. Although this port is not open on our firewalls it was thought to be prudent to identify any internal network traffic that was probing for devices listening on that port as they could be an attacker trying to gain a foothold (or advance one) in the Enterprise. To this ends we deployed Roger Thompson's WormCatcher (see http://www.wormwatch.org/about/) to listen on port 36794 at various strategic locations throughout the Enterprise, all reporting back to the Global response Team. No such traffic was identified. The updated Anti Virus software

on desktops would also serve to identify any infected nodes that might have this backdoor present.

To prevent the password stealing thread from sending out any cached passwords that this worm may have stolen the known recipients of the emails that BugBear sent these to (as described earlier) were blocked at the SMTP relays.

Eradication

Once we had identified the virus, the infected nodes, and re-enforced our perimeter protection, eradication was quite straightforward.

1. For the user community; signature files and software updates were deployed and users were made aware of the threat and to prevent further propagation. The Virus Response process worked very well with users calling the regional Help Desks who in turn escalated to their regional incident response team, who in turn escalated to the global incident response team for coordination and communication. Thus if someone in Canada identifies (through a print queue, SMB_Lure, or otherwise) an infected user in Italy it will be escalated to the Global Team who contacts the team in Europe to have the infected node in Italy removed from the network by the local support team.
2. File and Print Servers; New signature files were deployed and full system scans were done. None of the servers were actually infected with BugBear although some instances of BugBear infected files were identified in users home directories and quarantined.
3. Mail Servers and relays were never impacted by the BugBear virus but received signature file updates and full scans none the less.

Recovery

Workstations: Before any infected nodes are returned to the network they are rebuilt in an isolated environment by local support teams from known good media, brought up to date with current software, patches and Anti Virus, and any pre-existing data is scanned for virii. This process is simplified by having image disks available for system partitions and the default images store user data on separate logical partitions, thus allowing for quick formatting and rebuilding of system partitions without transferring data to other media. Prior to data being returned to the production environment it is scanned for malicious code. Although this seems like a labour-intensive approach it is deemed essential to safe computing.

File & Print Servers were not affected outside of the symptom of a Denial of Service attack while these huge print jobs were being processed. They did not require recycling the print spoolers but System administrators were disadvantaged by having to spend time deleting the errant print jobs from the print queues.

Printers were affected and often could not get through the huge print jobs so had to be powered off for a few minutes to clear the buffer.
Email systems were not affected and required no recovery.

The SMB_Lure still identifies the occasional rogue user who has become recently infected with BugBear. The alerts generated are distributed as per the virus response process and the user is disconnected from the network and the computer is rebuilt to standard prior to being returned to the network.

Lessons Learned

The original entry point of BugBear is still not known, but is believed to be by a user importing it through web based email access through an ISP. The cause of the infestation was a combination of a new Virus being released into the wild that no signature files were available to defend against coupled with some users having unpatched systems and/or non standard workstation with peer to peer networking or shares that were vulnerable to the exploit. Malicious code through Webmail access is a problem that needs to be revisited.

The method of assigning user rights to printers in Canada left something to be desired. In this case authenticated users in trusted domains had rights to submit print jobs, and BugBear submitted itself using the users access rights to numerous network printers in Canada, causing them to spew out huge amounts of paper (thankfully softwood *is* one of Canada's natural resources☺) and effectively cause a denial of service attack on print servers.

Anti Virus software on Print Servers in Canada was incorrectly configured to only scan Incoming files, which would not prevent an infected file from being opened or executed, only Outgoing scanning would have prevented that. Transactions that this would include would be such things as opening (on the server) any file containing executable code.

Because most actively spreading viruses these days spoof the "from" portion of the addresses, alerts from Gateway scanners are not necessarily getting to the true sender of viral emails and this can cause problems and confusion for the recipients. Not all gateway scanners have the ability to differentiate between viri that does or does not spoof the senders address and use this information to determine if an alert should be sent to the sender of the virus.

The SMB_Lure is a valuable tool in identifying network aware file infectors, particularly when infected hosts show no obvious symptoms.

Extras

References:

Symantec:
http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

Virus Bulletin – VGREP
http://www.virusbtn.com/resources/vgrep/index.xml

VUNet.Com - Bugbear virus on the loose By Iain Thomson [01-10-2002]
http://www.vnunet.com/News/1135543

VUNet.com - Bugbear side effect hits printers By Iain Thomson [07-10-2002]
http://www.vnunet.com/News/1135719

ZDNet News,- Bugbear to set new virus record, By Robert Lemos and Matthew Broersma - October 8, 2002, 5:12 AM PT
http://zdnet.com.com/2100-1105-961130.html

MessagesLabs – Press Release 16 Dec 2002

http://www.messagelabs.com/viewNewsPR.asp?id=113&cmd=PR

Virus Bulletin:
http://www.virusbtn.com/resources/viruses/bugbear.xml

Symantec
http://www.sarc.com/avcenter/venc/data/w32.bugbear@mm.html

Trend Micro
http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_BUGBEAR.A

MessageLabs:
http://www.messagelabs.com/viruseye/

Network Associates:
http://vil.nai.com/vil/content/v_99728.htm

Sophos:
http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

Sophos - Are your printers possessed by the Bugbear?

http://www.sophos.com/virusinfo/articles/bugbear2.html

Virus Bulletin – Vgrep
http://www.virusbtn.com/resources/vgrep/index.xml

Microsoft Security Bulletin (MS01-020)
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

Common Vulnerabilities and Exposures
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154

RFC 821 - SIMPLE MAIL TRANSFER PROTOCOL
http://www.ietf.org/rfc/rfc0821.txt

Just what is SMB? V1.2 Richard Sharpe 8-Oct-2002
http://samba.anu.edu.au/cifs/docs/what-is-smb.html

The File Extension Source
http://filext.com/d.htm

Anti Virus Information Exchange Network (AVIEN)
www.avien.org

The SMB-Lure, File-Share Worm Detector , by John K. Morris
http://morris.dnsalias.com/smb-lure.htm

Paul Schmehl – Checklogs Script
http://www.utdallas.edu/~pauls/checklogs.html

Roger Thompson's WormCatcher
http://www.wormwatch.org/about/

Last Updated: May 27th, 2017

Upcoming Training

SANS Madrid 2017	Madrid, Spain	May 29, 2017 - Jun 03, 2017	Live Event
SANS Atlanta 2017	Atlanta, GA	May 30, 2017 - Jun 04, 2017	Live Event
Community SANS Virginia Beach SEC504*	Virginia Beach, VA	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Mentor Session - SEC504	Reston, VA	Jun 13, 2017 - Aug 01, 2017	Mentor
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS ICS & Energy-Houston 2017	Houston, TX	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Seattle SEC504	Seattle, WA	Jul 10, 2017 - Jul 15, 2017	Community SANS
Community SANS Sacramento SEC504	Sacramento, CA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Ottawa SEC504	Ottawa, ON	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Community SANS Annapolis SEC504	Annapolis, MD	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Phoenix SEC504	Phoenix, AZ	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Des Moines SEC504	Des Moines, IA	Jul 24, 2017 - Jul 29, 2017	Community SANS
Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
Community SANS Raleigh SEC504	Raleigh, NC	Aug 07, 2017 - Aug 12, 2017	Community SANS
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event