# Global Information Assurance Certification Paper

# Bugbear worms its way to the Top

## An Analysis of a Bugbear Infection

## GCIH Practical Assignment Version 2.1, Option 1



**Bas Debbink**

**January 16, 2003**

# Table of Contents

## Introduction

This paper will describe the way in which the Bugbear virus operates and proliferates. It will also look at an infection, its symptoms, and the way in which this incident was handled. Hopefully the examples of what went right and what went wrong, as well as the listing of best practices, will assist others in dealing with virus and worm outbreaks.

The Bugbear virus was first detected on September 30, 2002. I immediately received an e-mail notification from our Anti-Virus vendor, indicating that the Alert Status for this new virus had been set to Medium:

> Greetings,
>
> The 4226 DATs will be available in response to W32/Bugbear@mm which is now at Medium risk, it could go High.
>
> -----Original Message-----
> Sent: Monday, September 30, 2002 11:54 AM
> Subject: W32/Bugbear@mm Enterprise Proactive
> Importance: High
>
> Hello,
>
> W32/Bugbear is currently at 'Medium' risk.
>
> McAfee AVERT has raised W32/Bugbear@mm to 'Medium' risk status.
>
> Virus: W32/Bugbear@mm
> Risk: Medium
> Information: http://vil.nai.com/vil/content/v_99728.htm
>
> EXTRA.DAT files are posted on the VIL page (see the above URL).
>
> The 4226 DATs will be posted in response to this threat. The updated DAT files will be available in approximately 2 hours, at the following locations:
>
> http://www.nai.com/naicommon/download/dats/find.asp
> ftp://ftp.nai.com/virusdefs/4.x/
>
> Best Regards,
> (Name and phone-number removed for sanitation purposes)
> McAfee Certified Intranet Defense Specialist
> Network Associates Inc.
> Your Network, Our Business!

The Director of our department had organized a miniature golf outing for our entire team, so there I was, lining up for the perfect putt, and watched the ball barely miss the Helpdesk Manager's head as my pager went off the second my club connected with aforementioned ball. I returned to my desk to find about a dozen e-mail messages and four voice mail messages waiting for me, alerting me to this new virus. Fortunately none of them were reports on actual infections.

Four days later, I received the following e-mail, indicating that the Alert Status for the Bugbear virus had been raised to High:

> This is a HIGH Virus Alert Upgrade for W32/Bugbear@MM. This virus has had it's (sic) Risk Assessment upgraded to High. Information about the W32/Bugbear@MM can be found on VIL at: http://vil.nai.com/vil/content/v_99728.htm.
>
> All DAT packages were made available 9/30/02.
>
> If you have received W32/Bugbear@MM or suspect you might have received W32/Bugbear@MM, please submit a sample to http://www.webimmune.net.
>
> For further information on Risk Assessments please see: http://www.mcafeeb2b.com/naicommon/avert/virus-alerts/avert-risk-assessment.asp
>
> Regards,
>
> AVERT
>
> ---
> McAfee AVERT - Analysis, Research, and Outbreak Management
> Visit www.avertlabs.com

An hour later, another notice was sent out, indicating that a stand-alone tool had also been released since one of the symptoms of this virus is that it disables popular Anti-Virus software:

> Greetings,
>
> With BugBear going to high risk we've decided to release AVERT Ultimate Stand-alone utility call Stinger.
>
> The release of Stinger is somewhat premature and not actually needed in this case as the products can detect and completely remove BugBear.

Stinger is design to be used when the products can't completely remove/repair a virus from a users system.

While the VirusScan product can detect and remove Bugbear Stinger can be used by ANYONE with or without our products.

This version also highlights some new engine technology that will be featured in the upcoming 4.2.40 engine release scheduled to hit the streets soon. Stinger uses Digitally Signed DATs and has process scanning to kill viruses in memory such as BubBear (*sic*).

It is located here http://vil.nai.com/vil/stinger/. Please read the page as there are details we want to make sure everyone is familiar with, especially the feedback mechanism.

More on Stinger as we go forward with its future.

Best Regards,

(Name and phone-number removed for sanitation purposes)
McAfee Certified Intranet Defense Specialist
Network Associates Inc.
Your Network, Our Business!

We would see several instances of the Bugbear virus over the next three weeks. It could have been much worse, however, if it weren't for the speed with which the Operations organization implemented the new Dat files. In the next few weeks, Bugbear overtook the Klez worm as the most prolific worm in Internet history, a doubtful honor.

# Part 1 – The Exploit

## 1.1 Brief Description

The Bugbear virus is based on a vulnerability that exists within Microsoft Internet Explorer 5.0 and 5.5 that exploits in-line frames embedded within e-mail messages, causing attachments to run without a user having to actually open it. The worm arrives in an e-mail attachment that has a double extension, and is an executable file of the type .exe, .scr, or .pif, and can also send itself out using its own SMTP engine, placing the double extension attachment to the e-mail message. E-mail addresses are gathered from an infected user's workstation, and can be combined into new e-mail addresses, which are then used to spoof the From field.
Bugbear also spreads through network shares, scanning for open file shares on the infected computer, and copying itself into those shares. A flaw in the code results in networked printers being seen as file shares, causing large print jobs to spool to those printers, and hundreds of pages of garbage to be printed as the worm tries to write its code.
The Backdoor Trojan portion of Bugbear allows an attacker to connect to the infected computer, giving him or her access to the computer's resources including passwords, logged keystrokes, and files.

The actual exploit, the iFrame vulnerability, on which the Bugbear virus is based, works as follows:

> Because HTML e-mails are simply web pages, IE can render them and open binary attachments in a way that is appropriate to their MIME types. However, a flaw exists in the type of processing that is specified for certain unusual MIME types. If an attacker had created an HTML e-mail containing an executable attachment and then modified the MIME header information to specify that the attachment was one of the unusual MIME types that IE handles incorrectly, IE would launch the attachment automatically when it rendered the e-mail.
>
> An attacker could use this vulnerability in either of two scenarios. She could host an affected HTML e-mail on a web site and try to persuade another user to visit it, at which point script on a web page could open the mail and initiate the executable. Alternatively, she could send the HTML mail directly to the user. In either case, the executable attachment, if run, would be limited only by user's permissions on the system.[1]

[1] Microsoft description at
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

## 1.2 CVE/CERT Assignments

Even though there is no individual Common Vulnerabilities and Exposures (CVE) number for the Bugbear virus, the exploit on which it is based has CVE number 2001-0154
The CERT number VU980499 was also identified with the vulnerability on which the Bugbear virus is based.

## 1.3 Operating System/Protocols/Services/Applications

The Bugbear virus affects the Microsoft Windows Operating Systems:

- Windows 9x (all versions, releases, and service packs)
- Windows NT (all versions, releases, and service packs)
- Windows 2000 (all versions, releases, and service packs)
- Windows XP (all versions, releases, and service packs)
- Windows ME (all versions, releases, and service packs)

The vulnerable applications are Internet Explorer 5.01 and 5.5. Since Microsoft Outlook and Outlook Express use Internet Explorer to render HTML pages (used in e-mail), these applications can execute the exploit. Installing the proper Microsoft patch, however, will prevent Internet Explorer from executing code automatically. Details on this patch can be found here:
http://www.microsoft.com/windows/ie/downloads/critical/q290108/default.asp
Another solution is to upgrade to Internet Explorer 6. Customers running Windows NT 4.0, Windows 2000, or Windows XP can choose any type of installation for IE 6, fully eliminating the vulnerability. Windows 9x and ME have to choose either a Typical or Full Install 6 (Typical Install is the default setting), since choosing either of the other installations options, Minimal Install or Custom Install, will not protect against the vulnerability.
Since the vulnerability lies in the application, any version and release of the Windows Operating Systems that can run Internet Explorer 5.01 and 5.5 are vulnerable to this exploit. Even though Windows XP is vulnerable to this attack, since it is delivered standard with IE 6, it is much less susceptible to the exploit.

The protocols that the Bugbear virus uses for propagation and operation are:

- TCP/IP (Transfer Control Protocol/Internet Protocol)
- UDP (User Datagram Protocol)
- SMTP (Simple Mail Transfer Protocol)
- SMB (Server Message Block)
- NetBIOS (Network Basic Input Output System)
- HTTP (Hyper Text Transfer Protocol)
- MIME (Multipurpose Internet Mail Extensions)

A brief description discussing these protocols can be found in section 2.2

The virus also tries to disable several Anti Virus and Personal Firewall applications, including McAfee's VirusScan and Zonelab's ZoneAlarm (See Chapter 2 for more information).

## 1.4 Variants

Even though there are no known variants of the Bugbear virus, it has the following aliases:
W32.Bugbear@mm, W32/Bugbear-A, W32/Bugbear.A@mm, W32/Bugbear.worm, Win32Bugbear, Worm/Tanatos, WORM_NATOSTA.A, Tanatos, W32/Bugbear, Tanat, W32/Tanat, I-Worm.Tanatos, W32/Bugbear.A@mm
Additionally, some virus scan software will detect this virus as Exploit-MIME.gen. or Exploit-MIME.gen.exe
There are several viruses other than Bugbear that exploit the same iFrame vulnerability, such as:
W32.Klez@mm (all variants)
W32.Frethem@mm (most variants)
W32.Yaha@mm (most variants)
W32.Nimda@mm (all variants)
W32.Lirva.a@mm

Other exploits exist that attack the iFrame vulnerability, such as reading local files, reading cookie files, and window spoofing. A list of these can be found at:
http://209.100.212.5/cgi-bin/search/search.cgi?searchvalue=iFrame&type=archives

## 1.5 References

More information on this virus can be found at the following locations:

http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html
http://vil.nai.com/vil/content/v_99728.htm
http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

Microsoft Security Bulletin describing MIME header vulnerabilities:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp

Other viruses and worms that use the iFrame exploit are described here:
http://vil.nai.com/vil/content/v_99237.htm - Klez

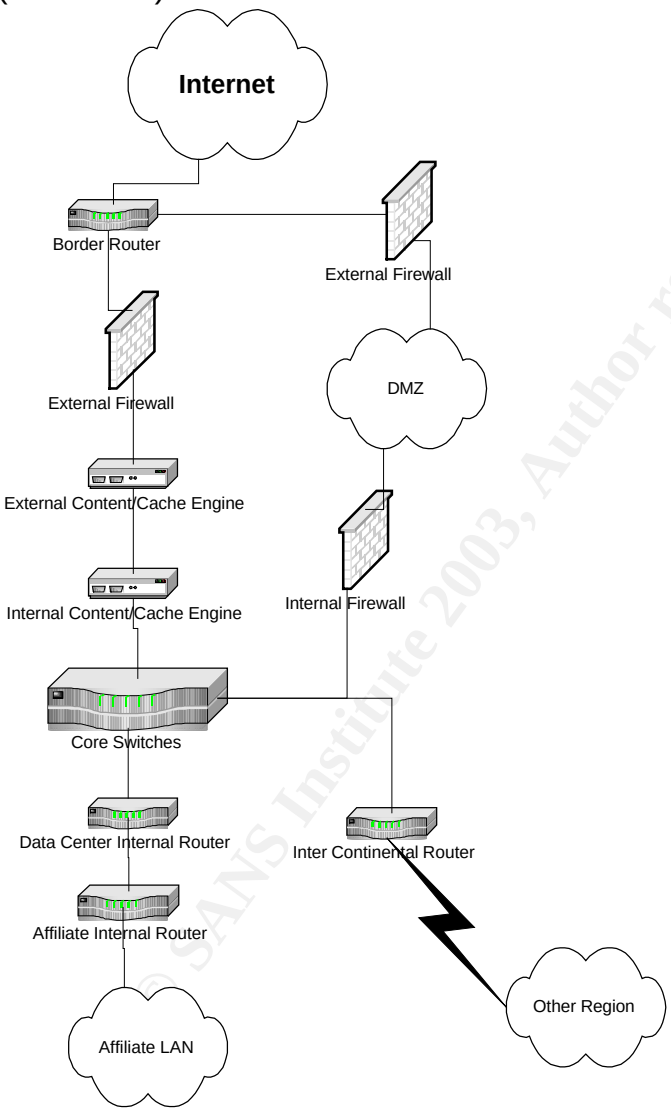
http://securityresponse.symantec.com/avcenter/venc/data/w32.klez.a@mm.html - Klez
http://vil.nai.com/vil/content/v_99949.htm - Lirva
http://securityresponse.symantec.com/avcenter/venc/data/w32.lirva.a@mm.html - Lirva
http://vil.nai.com/vil/content/v_99209.htm - Nimda
http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html - Nimda
http://vil.nai.com/vil/content/v_99528.htm - Yaha
http://securityresponse.symantec.com/avcenter/venc/data/w32.yaha.f@mm.html - Yaha
http://vil.nai.com/vil/content/v_99569.htm - Frethem
http://securityresponse.symantec.com/avcenter/venc/data/w32.frethem.b@mm.html - Frethem

iFrame Exploits:
www.packetstormsecurity.com

# Part 2 – The Attack

## 2.1 Description of the Network

Due to the size of the network, an accurate representation of the entire network is rather difficult. The following shows the basic layout of Internal Affiliate connections, connectivity to the Internet, as well as a connection to other regions (continents):




The external and internal Firewalls are Cisco PIX 525s, the cache/proxy engines are Cisco 500 series, and the Core Switches are Cisco 6500 series running IOS version 6.3(3a) with installed MSFC modules running Cisco IOS 12.x. Where possible, the Intrusion Detection probes are integrated into the Catalyst 6500 switches. The firewalls will be configured to allow all traffic in the outbound

direction and only limited (return) traffic in the inbound direction. PAT will be performed for this traffic. Internal routers are generally Cisco 7500 series, and the Intercontinental routers are Cisco 7200 series, both are running IOS versions 12.0 and 12.1(13). Outbound Web traffic is cached by the Content Engines. The software on the networking devices is patched frequently to stay up to date with the latest vulnerabilities. The internal infrastructure calls for two types of firewalls to be utilized. The first is a stateful inspection firewall that provides high speed and security. The second type is also a stateful inspection firewall that provides application proxy support and app/web management services. All Company supported firewalls are to be "appliance type" firewalls at a minimum rather than applications that run on an OS-based server such as NT or Solaris. The reason for this is that firewalls based on operating systems tend to have higher management costs and more widely know security exploits than "black-box" style firewalls.

At the heart of the network are the core routers at a Data Center; each region has its own Data Center (North America, Latin America, EMEA – Europe – Middle East – Africa, and Aspac – Asia Pacific). Each connection from any of the affiliates to the Internet is routed through the regional Data Center. Connectivity to the Data Center ranges from 64K Frame Relay links to full T3s, depending on the size of the affiliate. HTTP and HTTPs traffic to the Internet is routed through internal and external Content Engines, functioning as web page caching engines for an improved browsing experience, as well as a proxy server. All other Internet-bound traffic bypasses the content engines.

There are several DMZs on the network, all of them adequately protected with Firewalls on either end. The Border routers route incoming traffic to the appropriate segment, e.g. B2B, VPN pool, RAS pool, and filter some protocols such as Telnet and internal IP addresses (to prevent spoofing).

Links between the continents' Data Centers are DS3 lines, and are routed through Inter Continental routers.

Access to the Internet works as follows:

A Company user uses a Web Browser to open a web page. The browser will be configured to use the Cache/Content Engine as a proxy. If the page or object is in the cache storage, the object will be returned immediately. If the object is not in the cache, the CE will retrieve the requested content from the Internet. If the local Internet Access Point is unavailable, an alternate egress will be used. The alternate site will be one of the Primary or Secondary Network Centers. The Content Engine at that site will then access the Web content and return the results to the user.

This is a sample of the Ruleset on the external PIX Firewalls (hostnames and IP Addresses changed, and not all rules are shown):

```
PIX Version 6.2(2)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 DMZ-Slot:2 security10
nameif ethernet3 Statefull:3 security20
```

```
nameif ethernet4 unused1 security35
nameif ethernet5 unused2 security40
hostname BDFW01
domain-name int.bas.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
no names
access-list BD-acl-Statefull:3 deny ip any any
access-list BD-acl-outside permit ip host 111.222.213.6 host 11.22.0.121
access-list BD-acl-outside permit tcp host 111.222.123.6 host 22.11.44.205 eq
tacacs
access-list BD-acl-outside permit tcp host 111.222.123.6 host 22.11.33.206 eq
tacacs
access-list BD-acl-outside permit tcp host 111.222.123.31 host 22.11.33.206 eq
tacacs
access-list BD-acl-outside permit icmp host 11.22.0.247 111.222.213.0
255.255.255.0 echo-reply
access-list BD-acl-outside permit udp host 11.22.0.247 111.222.213.0
255.255.255.0 eq snmptrap
access-list BD-acl-outside permit udp host 11.22.0.247 111.222.213.0
255.255.255.0 eq syslog
access-list BD-acl-outside permit tcp host 11.22.0.247 111.222.213.0
255.255.255.0 eq 65
access-list BD-acl-outside permit udp host 11.22.0.247 111.222.213.0
255.255.255.0 eq tftp
access-list BD-acl-outside permit icmp host 11.22.0.247 150.160.170.0
255.255.255.0 echo-reply
access-list BD-acl-outside permit udp host 11.22.0.247 150.160.170.0
255.255.255.0 eq snmptrap
access-list BD-acl-outside permit udp host 11.22.0.247 150.160.170.0
255.255.255.0 eq syslog
access-list BD-acl-outside permit tcp host 11.22.0.247 150.160.170.0
255.255.255.0 eq 65
access-list BD-acl-outside permit udp host 11.22.0.247 150.160.170.0
255.255.255.0 eq tftp
.
.
```

.
access-list BD-acl-outside permit tcp 11.22.0.130 255.255.255.254 host 10.28.6.4 eq bgp
access-list BD-acl-outside permit tcp 11.22.0.130 255.255.255.254 host 10.28.6.3 eq bgp
access-list BD-acl-outside permit icmp 11.22.0.130 255.255.255.254 host 10.28.6.4 echo-reply
access-list BD-acl-outside permit icmp 11.22.0.130 255.255.255.254 host 10.28.6.4 echo
access-list BD-acl-outside permit icmp 11.22.0.130 255.255.255.254 host 10.28.6.3 echo-reply
access-list BD-acl-outside permit icmp 11.22.0.130 255.255.255.254 host 10.28.6.3 echo
access-list BD-acl-outside permit tcp 11.22.0.0 255.255.255.128 host 22.11.44.205 eq tacacs
access-list BD-acl-outside permit tcp 11.22.0.130 255.255.255.254 host 22.11.44.205 eq tacacs
access-list BD-acl-outside permit tcp host 11.22.0.167 host 22.11.44.205 eq tacacs
.
.
.
access-list BD-nonat-acl-inside permit ip host 111.222.213.6 host 11.22.0.121
access-list BD-nonat-acl-inside permit ip host 111.222.213.6 host 11.22.0.119
access-list BD-crypto-acl-DMZ-Slot:2-0 permit ip host 111.222.213.6 host 11.22.0.121
access-list BD-crypto-acl-outside-0 permit ip host 111.222.213.6 host 11.22.0.121
access-list BD-acl-inside-V1 permit ip host 111.222.213.6 host 11.22.0.121
access-list BD-acl-inside-V1 permit ip host 111.222.213.6 host 11.22.0.119
access-list BD-acl-inside-V1 permit tcp 172.26.4.214 255.255.255.254 host 11.22.0.121 eq ssh
access-list BD-acl-inside-V1 permit tcp host 172.26.4.253 host 11.22.0.121 eq ssh
access-list BD-acl-inside-V1 permit tcp host 172.26.5.4 host 11.22.0.121 eq ssh
access-list BD-acl-inside-V1 permit tcp host 172.26.4.253 host 11.22.0.119 eq
access-list BD-acl-inside-V1 permit tcp host 172.26.5.4 host 11.22.0.119 eq telnet
access-list BD-acl-inside-V1 permit tcp host 172.26.6.242 host 11.22.0.119 eq telnet
access-list BD-acl-inside-V1 permit tcp host 172.26.6.241 host 11.22.0.119 eq telnet
access-list BD-acl-inside-V1 permit icmp 111.222.213.0 255.255.255.0 host 11.22.0.121
access-list BD-acl-inside-V1 permit icmp 150.160.170.0 255.255.255.0 host 11.22.0.121

```
access-list BD-acl-inside-V1 permit icmp 111.222.213.0 255.255.255.0 host
11.22.0.119
access-list BD-acl-inside-V1 permit icmp 111.222.213.0 255.255.255.0 11.22.0.0
255.255.255.128 echo
access-list BD-acl-inside-V1 permit udp 111.222.213.0 255.255.255.0 11.22.0.0
255.255.255.128 eq snmp
access-list BD-acl-inside-V1 permit tcp 111.222.213.0 255.255.255.0 11.22.0.0
255.255.255.128 eq telnet
access-list BD-acl-inside-V1 permit icmp 111.222.213.0 255.255.255.0
11.22.0.130 255.255.255.254 echo
access-list BD-acl-inside-V1 permit udp 111.222.213.0 255.255.255.0
11.22.0.130
.
.
.
access-list BD-acl-inside-V1 deny icmp any any
access-list BD-acl-inside-V1 deny tcp any any eq imap4
access-list BD-acl-inside-V1 deny tcp any any eq 88
access-list BD-acl-inside-V1 deny tcp any any eq 135
access-list BD-acl-inside-V1 deny udp any any eq netbios-dgm
access-list BD-acl-inside-V1 deny tcp any any eq 137
access-list BD-acl-inside-V1 deny udp any any eq netbios-ns
access-list BD-acl-inside-V1 deny tcp any any eq netbios-ssn
access-list BD-acl-inside-V1 deny tcp any any eq pop2
access-list BD-acl-inside-V1 deny tcp any any eq pop3
access-list BD-acl-inside-V1 deny tcp any any eq smtp
access-list BD-acl-inside-V1 deny tcp any any eq 445
access-list BD-acl-inside-V1 deny udp any any eq 135
access-list BD-acl-inside-V1 deny udp any any eq 445
access-list BD-acl-inside-V1 permit ip 140.141.1.128 255.255.255.128 any
access-list BD-acl-inside-V1 permit ip 10.0.0.0 255.0.0.0 any
access-list BD-acl-inside-V1 permit tcp 140.151.8.194 255.255.255.254 host
140.141.3.3 eq www
access-list BD-acl-inside-V1 permit tcp 140.151.8.194 255.255.255.254 host
140.141.3.3 eq https
access-list BD-acl-inside-V1 permit tcp host 140.151.8.196 host 140.141.3.3 eq
www
access-list BD-acl-inside-V1 permit tcp host 140.151.8.196 host 140.141.3.3 eq
https
access-list BD-acl-inside-V1 deny ip any any
access-list BD-acl-DMZ-Slot:2-V1 permit tcp host 111.222.123.6 host
22.11.44.205 eq tacacs
access-list BD-acl-DMZ-Slot:2-V1 permit tcp host 111.222.123.6 host
22.11.33.206 eq tacacs
access-list BD-acl-DMZ-Slot:2-V1 deny ip any any
```

```
access-list BD-crypto-acl-outside-1 permit ip host 111.222.213.6 host
11.22.0.119
pager lines 24
logging on
logging timestamp
logging standby
logging buffered warnings
logging trap notifications
logging facility 23
logging queue 2048
logging host DMZ-Slot:2 111.222.123.20
no logging message 304001
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
interface ethernet4 100full
interface ethernet5 100full
icmp permit any unreachable outside
icmp deny any outside
icmp permit 111.222.213.0 255.255.255.0 inside
icmp permit 150.160.170.0 255.255.255.0 inside
icmp permit any unreachable inside
icmp deny any inside
icmp permit any unreachable DMZ-Slot:2
icmp deny any DMZ-Slot:2
icmp permit any unreachable Statefull:3
icmp deny any Statefull:3
mtu outside 1500
mtu inside 1500
mtu DMZ-Slot:2 1500
mtu Statefull:3 1500
mtu unused1 1500
mtu unused2 1500
ip address outside 11.22.0.21 255.255.255.128
ip address inside 140.141.1.201 255.255.255.128
ip address DMZ-Slot:2 111.222.123.4 255.255.255.0
ip address Statefull:3 192.168.0.5 255.255.255.252
ip address unused1 1.1.1.1 255.255.255.0
ip address unused2 2.2.2.1 255.255.255.0
ip verify reverse-path interface outside
ip audit info action alarm
ip audit attack action alarm
failover
failover timeout 0:00:00
failover poll 10
```

```
failover ip address outside 11.22.0.22
failover ip address inside 140.141.1.202
failover ip address DMZ-Slot:2 111.222.123.5
failover ip address Statefull:3 192.168.0.6
failover ip address unused1 1.1.1.2
failover ip address unused2 2.2.2.2
failover link Statefull:3
pdm history enable
arp timeout 14400
global (outside) 1 11.22.0.100 netmask 255.255.255.255
global (outside) 1 11.22.0.101 netmask 255.255.255.255
nat (inside) 0 access-list BD-nonat-acl-inside
nat (inside) 1 195.52.0.0 255.255.0.0 0 0
nat (inside) 1 177.48.0.0 255.255.0.0 0 0
nat (inside) 1 137.84.0.0 255.240.0.0 0 0
nat (inside) 1 10.0.0.0 255.0.0.0 0 0
```

The Firewalls specifically allow certain protocols to certain hosts, disallowing any other protocols and applications that are not excluded. The rules include standard practices such as blocking incoming NetBIOS, POP (Post Office Protocol), and SMTP traffic. The hosts that are allowed protocols such as Telnet and ICMP are generally management hosts. The Firewalls also provide Network Address Translation (NAT) services. The Firewall OS code is in the process of being upgraded, and Intrusion Detection will be enabled in this new release.

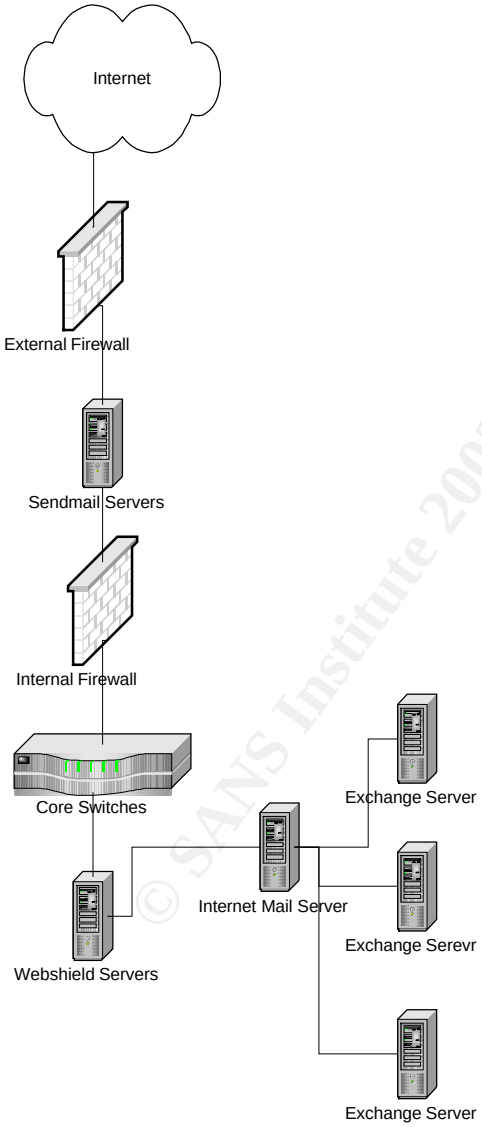
The border router is the first device that traffic coming from the Internet will pass. A typical border router ACL looks like this:

```
Extended IP access list 42 (Compiled)
    deny ip host 0.0.0.0 any
    deny ip 10.0.0.0 0.255.255.255 any
    deny ip 127.0.0.0 0.255.255.255 any
    deny ip 172.16.0.0 0.15.255.255 any
    deny ip 192.168.0.0 0.0.255.255 any
    deny ip 224.0.0.0 31.255.255.255 any
    deny ip 42.105.0.0 0.0.7.255 any
    deny ip 11.224.15.0 0.0.0.255 any
    deny ip 192.33.56.0 0.0.0.255 any
    permit esp any 42.105.0.128 0.0.0.127
    permit udp any 42.105.0.128 0.0.0.127 eq isakmp
    permit ahp any 42.105.0.128 0.0.0.127
    deny ip any 42.105.0.128 0.0.0.127
    permit icmp 11.224.168.0 0.0.0.255 host 11.224.208.174 echo
    permit icmp host 11.224.208.173 host 11.224.208.174 echo
    permit icmp any 42.105.0.0 0.0.0.127 echo
    permit icmp any any echo-reply
    permit icmp any 42.105.0.0 0.0.0.127 ttl-exceeded
```

deny icmp any any
permit ip any any

This ACL prevents IP Spoofing by blocking any incoming traffic that has an IP address that matches that of the internal network as well as public IP address ranges, and drops all traffic not specifically allowed. The border routers do not currently have any Intrusion Detection Features.
Affiliate routers generally do not have an ACL, or a very limited one allowing SNMP traffic to a specific network management segments and denying SNMP traffic everywhere else.

The Electronic Mail Environment looks as follows:




The Internet Mail Servers are located at the Data Centers at each region, and Exchange servers are located at each of the affiliate sites.

The Exchange servers are Compaq Proliant ML530's and 3000's running Microsoft Windows NT4.0 SP6, Microsoft Exchange 5.5 SP 3.0 and McAfee's Groupshield version 4.04. The Groupshield application scans all internal e-mail messages for viruses, deleting or quarantining the virus, and logging an alert. These logs can be used for analysis and trending. The Exchange servers themselves are running McAfee's Netshield, the server virus scanning software.
The Webshield servers are Compaq Proliant DL380 servers, running Microsoft Windows NT4.0 SP6 and McAfee's Webshield version 4.5. The Webshield application scans all Intefrnet email, both inbound and outbound for viruses. It either cleans or quarantines the e-mail and/or attachment. Similar to the Exchange servers, the Webshield servers are running McAfee's Netshield, the server virus scanning software
The Internet Mail Servers (hub servers) are Compaq Proliant 6500 servers running Microsoft Windows NT4.0 SP6 and Microsoft Exchange 5.5 SP 3.0
The Send Mail Gateways are Sunfire 280 servers running Solaris 8.0 and Sendmail Switch.
E-mail coming in from the Internet goes through the Internal Firewall, gets scanned by the Inbound Webshield servers, passes through the Internet Mail Server, and finally gets routed through the user's Exchange server and mailbox. When an Internal Email server has outgoing email pending, it will connect to the SMTP email forwarder located in the Network Services Security Zone. This connection will be via the Back DMZ. The SMTP server will then forward the email to the Internet via the Front DMZ. The Inbound and Outbound Webshield servers are separate in order to increase performance and avoid bottlenecks. Internal e-mail is routed from Exchange Server to Exchange Server and is scanned for viruses by the Groupshield software on the servers themselves. Updates to Webshield and Groupshield are set to occur every night, meaning that if updates are available, they will be implemented within 24 hours. Emergency releases, such as extra.dat files can be pushed out to the mail servers as well on an ad-hoc basis.

## 2.2 Protocols

The Bugbear virus is based on a vulnerability with MIME headers, and propagates over a network using TCP/IP and NetBIOS. Other protocols used by the Bugbear virus include HTTP, for the Remote Access Trojan, SMTP for e-mail propagation, and SMB, to propagate through network shares. The following sections describe these protocols, and identify some of the weaknesses inherent in them.

### TCP/IP

IP, the Internet Protocol, is the Network/Layer 3 protocol on which the Internet is based. It provides simple connectionless packet exchange, and follows a standard format for addresses. Its main job is to allow data to be transmitted across and between networks.

TCP, Transmission Control Protocol, is a Transport/Layer 4 protocol, and is used for error control. It runs over IP to provide error checking, and making sure that all traffic arrives at the destination, resending information if required. TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. TCP connects devices through ports, which act like doors on the computer. For instance, the Bugbear virus opens TCP port 36794 on the victim's computer, meaning that an attacker can connect to that port on the infected computer, allowing him or her to run the Remote Access Trojan piece of the virus. The main security weakness in TCP/IP is the absence of authentication and encryption. This makes it very difficult to determine the source and destination of TCP/IP traffic, since it allows for 'spoofing', i.e. hiding its true identity by assuming a false one. The absence of encryption means that any traffic sent using TCP/IP can be intercepted and read easily, which makes it very easy for intrusion detection systems to operate on the one hand, but also makes it easy for an attacker to intercept traffic. The Bugbear virus takes advantage of these vulnerabilities by faking its origination point, and by allowing remote administration to be made available to the attacker.

### UDP

UDP, User Datagram Protocol, is an unreliable, connectionless datagram protocol. It is a protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). Like TCP, UDP uses the Internet Protocol to actually get a message from one computer to another. Unlike TCP, however, UDP does not provide the service of dividing a message into packets (datagrams) and reassembling it at the other end. Specifically, UDP doesn't provide sequencing of the packets in which the data arrives. Network applications that want to save processing time because they have very small data units to exchange (and therefore very little message reassembling to do) may prefer UDP to TCP. The Trivial File Transfer Protocol (TFTP) uses UDP instead of TCP, and streaming media uses UDP as well since the human eye or ear will make up for lost packets. Bugbear only uses UDP for its network propagation, since UDP is part of how NetBIOS connects to shares. As with TCP, UDP uses ports as well, 137-139 in the case of NetBIOS.

### NetBIOS

NetBIOS, Network Basic Input/Output System is an API (Application Programming Interface) that allows Windows and DOS computers on a network to communicate. Most Windows Operating Systems run NetBIOS over TCP/IP to simplify use of network resources such as file shares and network printers.

NetBIOS uses TCP and UDP Ports 137-139 and 445 to connect to resources. Port 445 was added by Windows 2000 to allow for file and print sharing.
The vulnerability in NetBIOS lies in the fact that it does not offer any type of built-in security features, such as authentication and encryption, allowing for exploits such as SMBRelay and other Man-in-the-Middle attacks.

### SMB

SMB, Server Message Block, is a protocol for sharing, among others, files and printers between computers. SMB is a client/server, request-response protocol, which means that servers make file systems and other resources available to clients on a network. Those clients can then access file shares and printers on those servers.Once they have established a connection, clients can then send commands to the server that allow them to access shares, and open, read and write files. This is how the Bugbear virus is able to look for, and connect to, file shares and network printers.
Since SMB relies on NetBIOS for its communication, it inherits all of NetBIOS's weaknesses and vulnerabilities.

### MIME

MIME stands for Multipurpose Internet Mail Extension. It is a definition of a file type so that a mail reader or web browser can process the file more easily. This applies to every file, image, application, etc. that may be requested by a web browser or mail program. If the extension is not stored within the Mime Type, the server will not recognize it and will not be able to assist the application and that particular file type will not be viewable by the user. The server identifies mime types by the format subtype/type extension, for example video/quicktime and image/jpeg.
Malicious code can exploit the MIME header either by crafting a long string resulting in buffer overflows attacks, or by tricking an application such as Outlook into thinking that an infected file is in fact a valid file type. The Bugbear virus uses this latter exploit.

### HTTP

HTTP, Hyper Text Transfer Protocol, is the primary protocol utilized by the World Wide Web (WWW). HTTP defines how files on the WWW are transferred. HTTP is a "request-response" type protocol that specifies that a client will open a connection to a server then send a request using a very specific format. The server will then respond and close the connection. Bugbear uses HTTP to allow an attacker to connect to its Backdoor Trojan.

### SMTP

The Simple Mail Transfer Protocol was designed to offer an electronic mail store and forward solution. A mail server will receive a message, store it to disk, and then attempt to forward it to another mail server. If the other mail server is unavailable, the source server will hold the message and attempt to resend the message at regular intervals. The Bugbear virus comes with its own SMTP engine, which it uses to mail itself to addresses harvested from an infected host's Personal Address Book.

## 2.3 How Bugbear Works

Bugbear propagates via e-mail, and by using open shares in local networks.
In order to mass-mail itself, the worm will search for e-mail addresses in the following file types:
".ODS", ".MMF", ".NCH", ".MBX", ".EML", ".TBB", ".DBX".
The local mail account details are read from the registry and the worm's built-in SMTP engine takes care of sending the messages. The mail generated by Bugbear uses HTML/iFrame_Exploit - this way, an executable attachment is automatically run on unpatched systems, without asking for user's confirmation, even if the mail is only displayed in Outlook's preview window. The email will have to be written in HTML or have HTML elements embedded in it. This exploit allows the virus to propagate without a user having to actually open an attachment.
The iFrame vulnerability itself allows an attacker to execute script on any page that contains frame or iframe (inline frame) elements, ignoring any protocol or domain restriction implemented by Internet Explorer. By executing script, an attacker could steal cookies from almost any site, access and change content in sites and in most cases also read local files and execute arbitrary programs on the client's machine. Frames, which are essentially sections of the browser display that are separate Web pages, may contain URLs in other domains or protocols, and therefore have strict security rules, which prevent frames in one domain to access content and information in another. It is possible, however, to set the frame's URL. Setting the child frame's URL to "javascript:[code]" will execute the script in the context of the currently loaded URL. [2] If an e-mail message contains an executable attachment with a MIME type that is incorrectly specified, a flaw in Internet Explorer will cause the attachment to be executed without displaying a warning dialogue.
The virus can forge E-mail addresses by taking part of one e-mail address from the Personal Address Book, and another part from a different address. For example, if a user has johndoe@test.com and homersimpson@doh.org in their address book, the From field in the outgoing message can be forged, appearing to come from homsersimpson@test.com . The subjects of mail messages sent

[2] http://www.internetnews.com/dev-news/article.php/1459341

by Bugbear are randomly generated. See Appendix B for a list of possible subjects.
The body and the attachment's file name are also randomly generated. The attachment's file name has a double extension. The first one can be: ".diz", ".txt", ".cpp", ".jpeg", ".jpg", ".gif", ".bmp", etc. The second one is .exe, .pif, or .scr. Bugbear will then attempt to propagate itself by sending a copy itself using its own SMTP engine to every person in the infected user's Address Book.

Bugbear will check a local computer's open shares and attempt to copy infected files into those shares. A flaw in the code identifies networked printers as file shares to the worm, resulting in Bugbear trying to 'write' its contents to the Printer, which in turn results in hundreds of pages of garbage to be printed.

When it is run, Bugbear copies itself using random names in the Windows system folder and in the startup folder. For the copy dropped in the system folder, the file name always starts with 'f' (e.g. "c:\windows\system\fjmt.exe"). For the copy dropped in the startup folder, the file name always starts with 'c' (e.g. "c:\windows\Start Menu\Programs\Startup\cok.exe"). The Bugbear virus is able to find the location of the Startup folder from the following Registry key: HKEY_Current_User\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Startup.
In order to be executed when Windows starts, the worm will add a registry key under "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce", with the path to the executable file that represents the worm itself.

Bugbear drops a key logger dll and three more files with binary data, using random names. Two of the data files have a ".dll" extension (84 bytes and 34 bytes long), and the third has the ".dat" extension and is 2 bytes long. These three files contain only data used in the spreading process. This Key Logger Trojan is known as the PWS-Hooker Trojan, and is also dropped by viruses such as W32/Badtrans. This Trojan is able to capture keystroke information entered by the user, store them in memory or the .dat file, and when an attacker connects to the victim's computer using the remote administration exploit (see next paragraph), he or she can obtain the captured keystroke logs. On Windows 9x/ME systems it also tries to gather cached passwords using WNetEnumCachedPasswords. The keystrokes and cached passwords are sent to an e-mail account chosen at random from this list of addresses that are hard-coded inside the worm:
mshaw@hispostbox.com
mannchris@gala.net
gili_zbl@yahoo.com
c.willoughby@myrealbox.com
brdlhow@ml1.net
sc4579@excite.com
jwwatson@excite.com

stevechurchis@excite.com
langobaden@excite.com
jacopo58@excite.com
sctanner@myrealbox.com
erisillen@canada.com
sergio52@mac.com
rvre2736@fairesuivre.com
zr376q@yahoo.com
t435556@email.it
sdsdfsf@callme.as
boxhill@teach.com
stickly@login.pe.kr
vique@aggies.org
sm2001@mail.gerant.com
rwilson@singmail.com
[3]

As mentioned, Bugbear also works as a Remote Administration Trojan. Bugbear opens port 36794 and listens for commands from a remote machine. The remote user can find, copy, write to, execute, or delete files; upload then execute a file; start, list, or stop processes; retrieve cached passwords; send the process list, intercepted keystrokes, and system info (user, processor type, memory info, drive info, network resources list).

> The commands that are accepted by the backdoor component of the worm require an authentication password, followed by the command and the command parameters, if any are required. For example, the "i" command does not require parameters. The following reply is an example of the "i" command being performed:
>
> Server: '<BDW95>'
> User: 'BD'
> Processor: I586
> Win32 on Windows 95 v4.10 build 1998
> -
> Memory: 127M in use: 50%  Page file: 920M free: 865M
> C:\ - Fixed Sec/Clust: 64 Byts/Sec: 512,  Bytes free: 4786129063/4786129063
> D:\ - CD-ROM
>
> Network:
> unknow cont  (null) (Microsoft Network)
> unknow cont  (null) (Microsoft Family Logon)

[3] http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

The "h" command contains a parameter, which is a port number. This command causes the backdoor component to open that port and listen to the commands transferred in the form of HTTP Get requests. The requests are parsed, processed, and the results are returned in the form of the formatted HTML pages. This gives a hacker a convenient way to browse the compromised computer's resources[4].

The attacker can upload files to the compromised computer. If a text file is clicked, its contents appear in the browser windows. Otherwise, the browser will offer to download the file and open it using an associated application.

The following images show examples of the backdoor interface of the Bugbear virus[5]

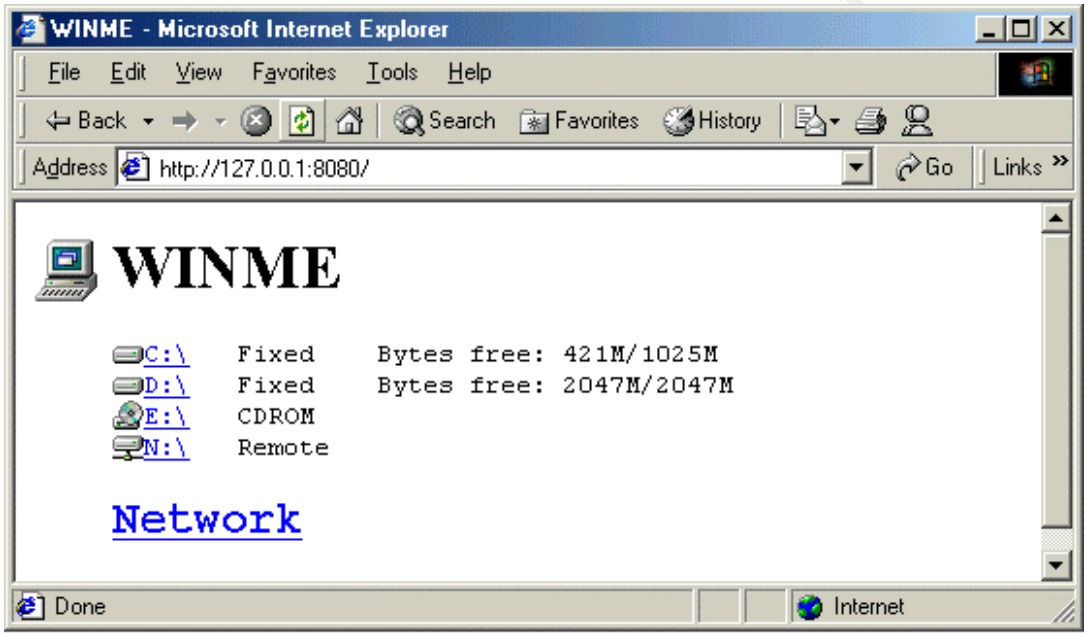
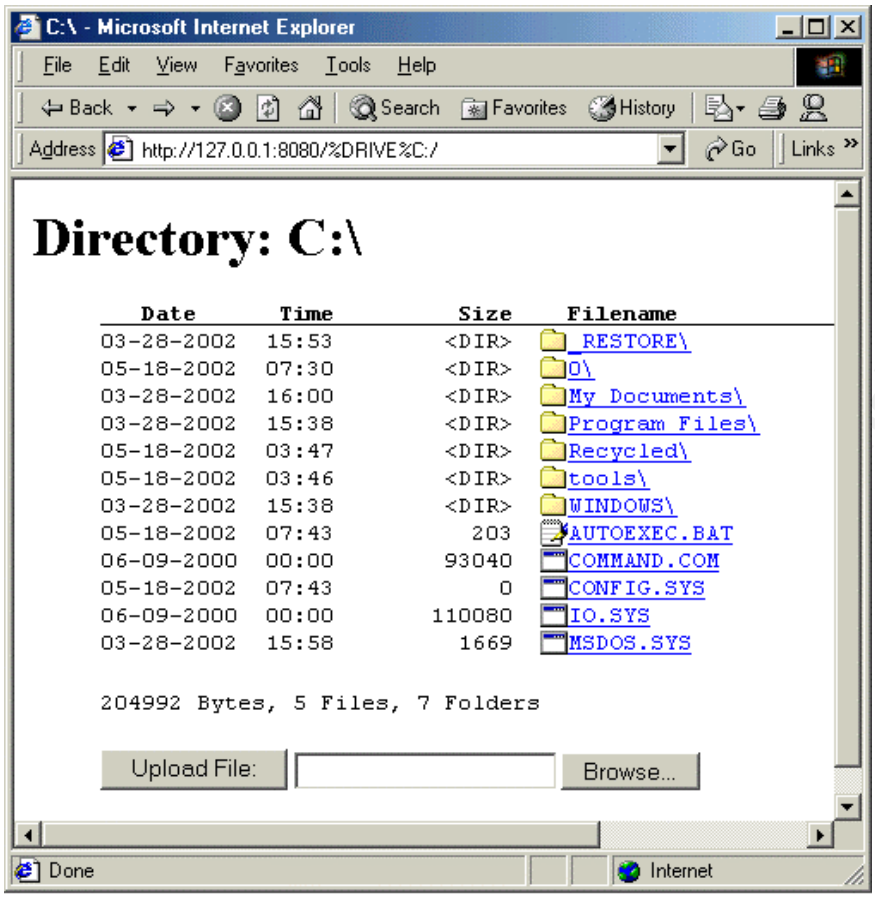



Image Copyright © F-Secure Corporation

[4] http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

[5] Images used with permission from http://www.F-Secure.com





## 2.4 Description and diagram of the attack

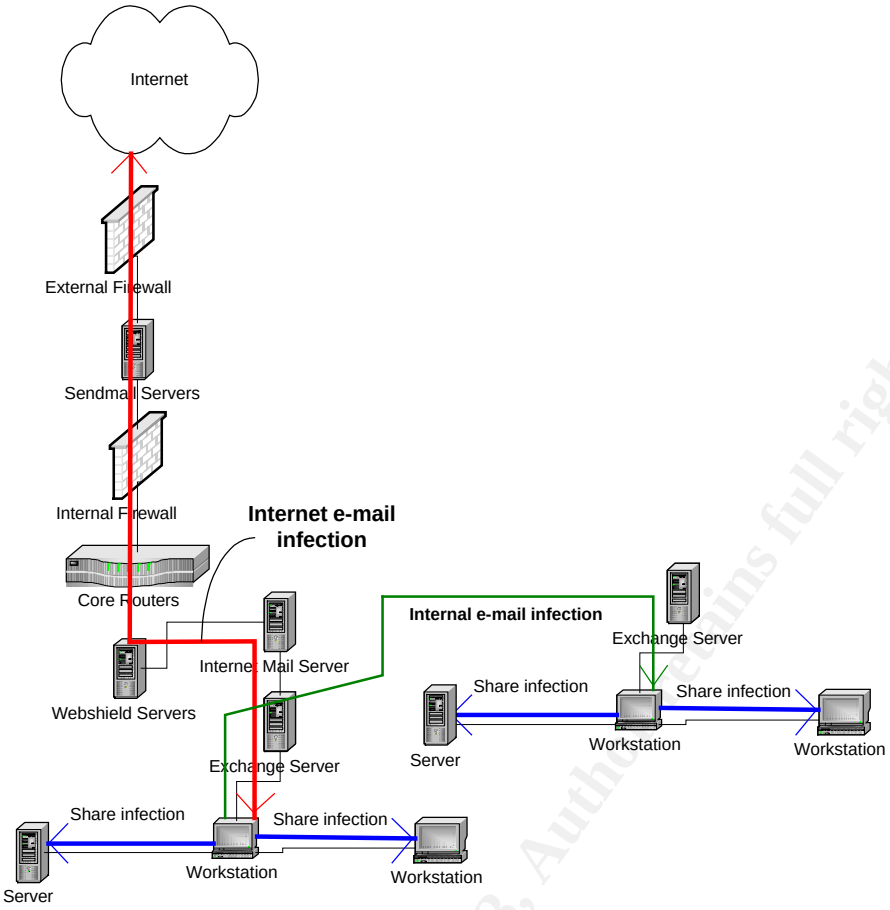
A step-by-step analysis of the attack would be as follows:

- A user downloads an infected file from the Internet. Since the user's Anti-Virus software was out of date or disabled, the Bugbear virus would not be intercepted.
- Another possibility of infection would be an e-mail message with an infected attachment. If the user had an un-patched version of Internet Explorer 5.01 or 5.5 and would look at the message in Outlook or Outlook Express, with a preview pane, the malicious code would run automatically.
- The user can also get infected if he or she has an unprotected share on their hard drive, and someone else on the same network was infected with Bugbear, and had access to that file share. The worm could copy itself into the user's Startup folder, resulting in the device being fully infected at the next reboot.
- The Bugbear virus would copy itself to the "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\R

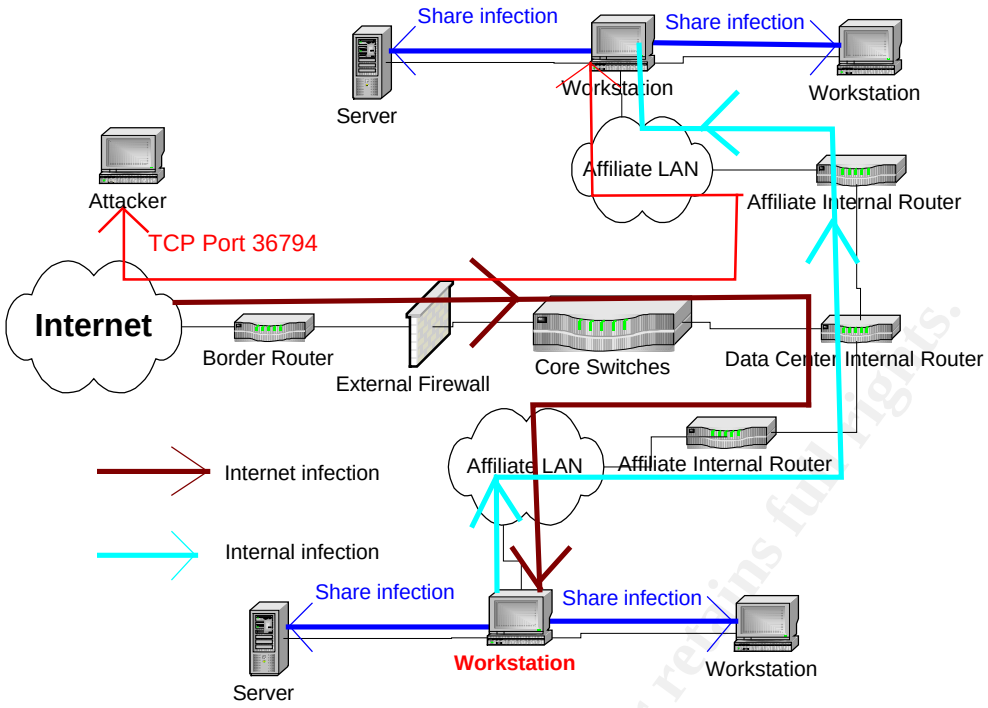
unOnce" registry key so that it would launch itself when the system is rebooted.

- The worm creates three .dll files and a .dat file to the systems Windows System and Startup folders, installing the keylogger Trojan.
- The Trojan part of the worm opens TCP port 37694 to allow the attacker to connect to the Backdoor Trojan.
- The worm enumerates the infected computer's open shares, and tries to propagate itself by copying an infected file into another computer's Startup folder.
- Due to a flaw in the code, networked printers are considered to be file shares, and the worm will attempt to 'write' itself to the printer.
- The Bugbear virus will search through the infected user's Personal Address Book, create fake e-mail addresses combining parts of real e-mail addresses, and spoof the From field with the created e-mail addresses.
- The virus will then send itself to people in the infected user's Personal Address Book using its own SMTP engine, spoofing the From field by inserting the created e-mail address.
- The virus will attempt to stop several running processes, including Anti-Virus products and Personal Firewalls.

This flowchart shows a possible Bugbear scenario, arriving through an e-mail message from the Internet. Note that the Internet e-mail infection can go both ways, since the infected computer can also send out e-mail messages to Internet addresses. It also shows further propagation through network shares to both servers and workstations (any device with an open share). The third propagation method can be e-mail propagation within the network, between two affiliate sites, both of which have their own Exchange servers:




A different scenario is shown in the following diagram, which shows a user getting infected through a download off the Internet, which can result in the virus spreading through local network shares (local as in at the same affiliate), or go across the WAN, infecting devices at other sites. It also shows the Trojan part of Bugbear, allowing an attacker to get access to an infected workstation through TCP Port 36794:




## 2.5 Signature of the Attack

The worm arrives attached to an e-mail message. The file size is 50,688 bytes (UPX packed). This information can be used to block attachments at Mail Gateways that allow file filtering based on file size or file type. If the proper levels of Anti-Virus software are loaded on your Internet Mail Gateways, the log files show entries indicating that an attachment containing the W32/Bugbear@MM worm was blocked by Webshield.

When run on the victim machine, the virus copies itself to %WinDir%\%SysDir% as F***.EXE (where * represents a random character). For example:
Windows 98: C:\WINDOWS\SYSTEM\FYFA.EXE
Windows 2000 Pro: C:\WINNT\SYSTEM32\FVFA.EXE
The executable files are 50,688 bytes long.
The following Registry key is set in order to run at the next system startup:
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce "%random letters%" = %random filename%.EXE (Win9x)
The worm copies itself to the Startup folder on the victim machine as C**.EXE (where * represents a random character), for example:
Windows 98: C:\WINDOWS\Start Menu\Programs\Startup\CUK.EXE
Windows 2000 Pro: C:\Documents and Settings*$user*\Start Menu\Programs\Startup\CYC.EXE

The worm also creates three files with the extension dll in the System directory. Only one is actually a dll library. The other two are data files used by the worm. The real dll name is 7 characters long, for example:

C:\WINDOWS\SYSTEM\zakqlkq.dll
The worm uses this dll for logging keystrokes. This can be used to steal passwords and other sensitive information. The installed dll is 5,632 bytes in size. The other dll files contain binary data and are used by the Trojan for propagation. They are 84 bytes and 34 bytes long and their names are 6 and 7 characters each, for example:
C:\WINDOWS\SYSTEM\eamoim.dll
C:\WINDOWS\SYSTEM\pagurgu.dll. [6]
Looking for files of that byte size that also follow the naming convention used by the Virus is a good way to check for infection. Employing a file integrity program will be helpful in detecting creation of these files as well as the creation of the registry key. More on that in section 2.6

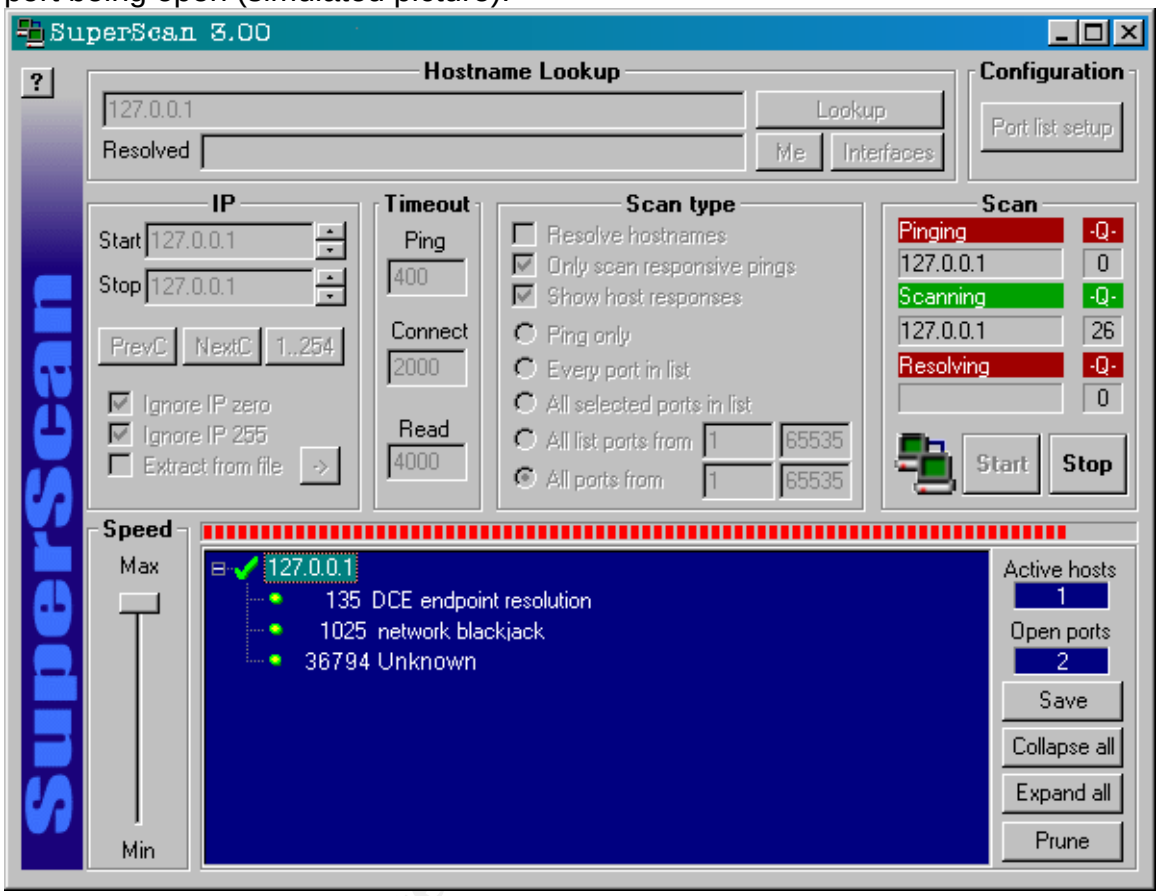
The worm opens a port on the victim machine, TCP 36794, to enable the attacker to connect to the Backdoor Trojan. This remote access server allows an attacker to upload and download files, and run and terminate processes. The opening of port 36794 would have been visible using the netstat –a command from the command line, and, if infected, a PC would have shown something similar to this:
C:\>netstat -a
Active Connections

| Proto | Local Address | Foreign Address | State |
|---|---|---|---|
| TCP | BDW2K:1167 | Exchange2K:1046 | ESTABLISHED |
| TCP | BDW2K:1171 | Exchange2K:1062 | ESTABLISHED |
| TCP | BDW2K:1182 | Exchange2K:1046 | ESTABLISHED |
| TCP | BDW2K:1186 | Exchange2K:1062 | ESTABLISHED |
| TCP | BDW2K:1199 | FPServer:microsoft-ds | ESTABLISHED |
| TCP | BDW2K:1383 | 2Exchange:1081 | ESTABLISHED |
| TCP | BDW2K:1632 | FPServer:microsoft-ds | ESTABLISHED |
| TCP | BDW2K:1841 | Exch55:3002 | ESTABLISHED |
| TCP | BDW2K:1899 | EXPF1:3752 | ESTABLISHED |
| **TCP** | **BDW2K:36794** | **0.0.0.0:0** | **LISTENING** |
| TCP | BDW2K:1911 | WINDC1:kerberos | TIME_WAIT |
| TCP | BDW2K:1913 | WINDC2:microsoft-ds | TIME_WAIT |
| TCP | BDW2K:1915 | WINDC3:microsoft-ds | TIME_WAIT |
| TCP | BDW2K:1917 | WINDC1:kerberos | TIME_WAIT |
| TCP | BDW2K:1918 | RACSRV:microsoft-ds | TIME_WAIT |

The bold entry shows that the Trojan has opened port 36794 on the infected workstation and is now listening on port 36794. If the State had said ESTABLISHED, the situation would have been much worse, since that would

[6] http://www3.ca.com/solutions/collateral.asp?CID=34561&ID=2602&CCT=19520

indicate that someone actually connected to the port and has compromised the computer.
Also, a port scanning application, such as SuperScan, would have noticed this port being open (simulated picture):




Bugbear checks for various running processes, stopping them if found. These processes include many popular AV and personal firewall products, which Bugbear will attempt to stop every 30 seconds; see Appendix A for a full list. If you know you have one of those applications installed on your workstation or server, but you don't see the process related to that application in the Processes list in Task Manager, chances are you got infected.

There are ways to detect the Bugbear virus traveling across the network, which in turn enables use of Intrusion Detection alerts to detect the virus, and consequently, filtering of traffic that matches the signature. The following Snort signature can be used to capture output from the Bugbear virus:

alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg: "NETBIOS BugBear Worm.";
content: "|77 00 69 00 6B 00 2E 00 65 00 78 00 65 00 00 00|"; flags: A+;)

The resulting output would be as follows:

```
TCP TTL:128 TOS:0x0 ID:174 IpLen:20 DgmLen:276 DF
***AP*** Seq: 0x672AFD85  Ack: 0x5A3BD20C  Win: 0x41DB  TcpLen: 20
00 00 00 E8 FF 53 4D 42 A2 00 00 00 00 18 07 C8    .....SMB........
00 00 00 00 00 00 00 00 00 00 00 00 02 08 10 03    ................
01 08 E1 00 18 FF 00 DE DE 00 92 00 16 00 00 00    ................
00 00 00 00 96 01 03 00 00 00 00 00 00 00 00 00    ................
20 00 00 00 00 00 00 00 02 00 00 00 44 00 00 00    ...........D...
02 00 00 00 03 95 00 00 5C 00 44 00 6F 00 63 00    ........\.D.o.c.
75 00 6D 00 65 00 6E 00 74 00 73 00 20 00 61 00    u.m.e.n.t.s. .a.
6E 00 64 00 20 00 53 00 65 00 74 00 74 00 69 00    n.d. .S.e.t.t.i.
6E 00 67 00 73 00 5C 00 41 00 64 00 6D 00 69 00    n.g.s.\.A.d.m.i.
6E 00 69 00 73 00 74 00 72 00 61 00 74 00 6F 00    n.i.s.t.r.a.t.o.
72 00 5C 00 53 00 74 00 61 00 72 00 74 00 20 00    r.\.S.t.a.r.t. .
4D 00 65 00 6E 00 75 00 5C 00 50 00 72 00 6F 00    M.e.n.u.\.P.r.o.
67 00 72 00 61 00 6D 00 73 00 5C 00 53 00 74 00    g.r.a.m.s.\.S.t.
61 00 72 00 74 00 75 00 70 00 5C 00 77 00 69 00    a.r.t.u.p.\.w.i.
6B 00 2E 00 65 00 78 00 65 00 00 00                k...e.x.e...
```
[7]

A Sniffer can capture the Bugbear worm's network propagation as well. Since the worm tries to copy itself to the Startup folder of open network shares on other computers, traffic showing SMB requests to open shares and copy files to other directories would be prevalent, an example of which can be seen in this Ethereal screenshot:

```
0000  00 10 b5 0d ac b9 00 03  47 b7 98 f5 08 00 45 00   ........ G.....E.
0010  00 98 6b 54 40 00 80 06  0a f0 c0 a8 01 64 c0 a8   ..kT@... .....d..
0020  01 67 0a e8 00 8b 1b 80  34 17 00 17 4f a3 50 18   .g...... 4...O.P.
0030  43 87 ce 38 00 00 00 00  00 6c ff 53 4d 42 2d 00   C..8.... .l.SMB-.
0040  00 00 00 18 07 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 01 c8 ff fe 00 00  01 08 0f ff 00 de de 01   ........ ........
0060  00 11 00 16 00 20 00 f4  16 23 3e 12 00 00 00 00   ..... .. .#>.....
0070  00 ff ff ff ff ff 00 00  00 2b 00 5c 77 69 6e 64   ........ .+.\wind
0080  6f 77 73 5c 73 74 61 72  74 6d 7e 31 5c 70 72 6f   ows\star tm~1\pro
0090  67 72 61 6d 73 5c 73 74  61 72 74 75 70 5c 43 59   grams\st artup\CY
00a0  43 2e 65 78 65 00                               C.exe.
```

The accessing of the network shares will also generate a lot of traffic through TCP port 137, which can be picked up by Intrusion Detection Systems.

## 2.6 Protection against Bugbear

Microsoft originally published the iFrame Exploit in March of 2001. An appropriate patch for Internet Explorer 5.01 and 5.5 had been available for well over a year when the Bugbear virus broke out. As mentioned in section 1.3, upgrading to Internet Explorer version 6 will prevent the malicious code from

[7] http://www.e-secure-db.us/dscgi/ds.py/Get/File-12349/RE_Snort-sigs_Yet_another_BugBear_signature.txt

running automatically as well. It is a system administrator's responsibility to keep up with security patches, and to be aware of the risks that are involved with not being adequately protected. This is why it is vital that whenever new devices are put on the network, they have the proper patch level. This is generally not a user's responsibility in a corporation where servers and workstations are deployed following strict guidelines. Often, workstations are delivered to a user with a standard image, taking the guesswork out of their hands. However, we find that very frequently, unauthorized devices find their way on the network, mostly development or test workstations and servers, which do not follow this practice. In addition to Microsoft's Security Patch to fix the actual vulnerability, all the major Anti-Virus vendors released appropriate patches to their Virus Scan products the same day the virus was discovered. As more and more information became available, the vendors also started releasing stand-alone clean-up tools, primarily to circumvent the problem with Bugbear disabling Virus Scan software. Some of the vendors' patch and removal tool links are:
http://www.mcafeeb2b.com/naicommon/download/dats/find.asp
http://vil.nai.com/vil/stinger/
http://securityresponse.symantec.com/avcenter/defs.download.html
http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.removal.tool.html
http://www.kasperskylabs.com/updates.html
http://www.pandasoftware.com/upgrades.asp?idioma=1
http://www.pandasecurity.com/utilities/bugbear.htm

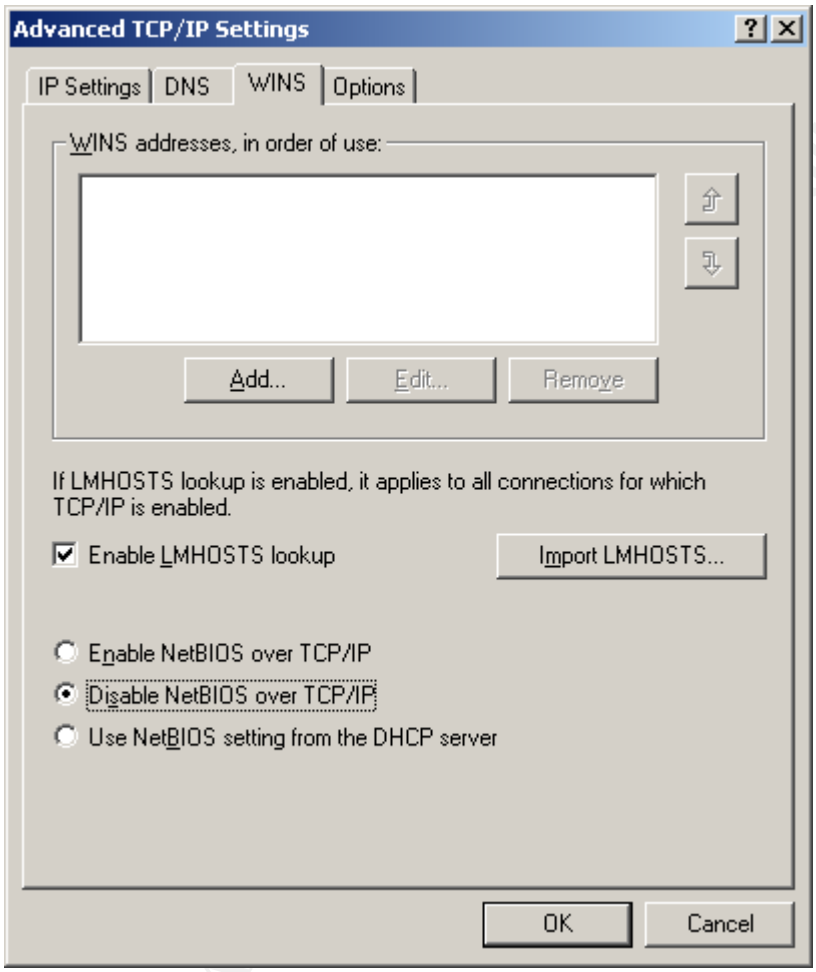
As far as protecting oneself against the virus once it has already entered the network, i.e. preventing further propagation, there are a few steps that can be taken that are always a good idea when strengthening a network, not just when responding to a specific incident.
To counter the e-mail propagation aspect of the Bugbear virus, besides having up to date virus protection software installed on the mail servers, and patched versions of Internet Explorer and Outlook, file attachment filtering should be instituted. Executable files are the favorite means available to a virus writer to ensure infection and propagation. It is a good practice to block any type of executable file at the Mail gateways, including files with extensions such as .exe, .vbs, .bat, and .scr. If there really is a strong need to send these types of files across an e-mail environment, perhaps because you do a lot of work with outside vendors, you could institute a policy that requires executable files to be zipped and password protected, and perhaps even encrypted. That little bit of extra work for the people who need to send these types of files will greatly reduce the chances of virus infections entering a networking environment.
Restricting users' access to the desktop, which can be done with Windows NT, 2000, and XP, would have prevented the DLL files from being written to the system folders and changes to the Registry being made. Suggested permissions to the
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Once
registry keys are:
Administrators (Full Control)
SYSTEM (Full Control)
Creator Owner (Full Owner)
Everyone (Read)
This will allow people with Administrative rights, perhaps helpdesk personnel, to install software, but disallow a regular user to make changes to those registry keys.
Using a file integrity program such as Tripwire will also help prevent, or at least detect, whenever a change is made to the computer. These types programs would have been triggered upon creation of the registry key, as well as the creation of the dll, exe, and dat files by the worm.
Perimeter protection was simplified because the port number that was used to enable the key logger and remote access Trojan, 36794, was known quickly, enabling network administrators to implement rules on their Firewalls and/or routers, blocking traffic through that port. A host-based intrusion detection solution, or personal firewall would have also helped detect and/or prevent traffic leaving or coming in through that port. A rule that could be added to a Firewall would be:

access-list BD-acl deny tcp any any eq 36794

This would block any TCP traffic coming in to and going out of the network through port 36794.
Network intrusion detection systems should be implemented to counter network propagation by the Bugbear virus. Even if those systems are unable to stop the virus, the logs will at least identify which devices are infected. Most Intrusion Detection vendors released updated signatures shortly after the Bugbear virus hit, allowing their products to detect this worm.
Limiting the number of open shares on the network is also good practice. The first step in that, of course, is knowing where your network shares are. Tools such as ShareEnum (http://www.sysinternals.com/ntw2k/source/shareenum.shtml) will list shares on your domain, as well as some of the security settings on those shares. Since the network propagation happens through SMB (NetBIOS over TCP/IP), there are some additional, more radical options. One of the solutions would be to disallow all traffic on TCP and UDP ports 137-139 and 445 (for Windows 2000) within a network, since those are the ports that NetBIOS uses to connect to shares. You can accomplish this by implementing the following rules on the routers:
access-list *access_list_name* deny tcp any any eq 137
access-list *access_list_name* deny tcp any any eq 138
access-list *access_list_name* deny tcp any any eq 139
access-list *access_list_name* deny tcp any any eq 445
access-list *access_list_name* deny udp any any eq 137
access-list *access_list_name* deny udp any any eq 138
access-list *access_list_name* deny udp any any eq 139

access-list *access_list_name* deny udp any any eq 445
(Even though port 138 is generally not identified with NetBIOS, it offers the NetBIOS datagram service)
Another option is to unbind NetBIOS from TCP/IP if you don't use any network shares or network printers. For Windows 2000, this can be done by clicking on Start-Settings-Network and Dial-up Connections. Right-click on your network connection and click Properties. Select Internet Protocol (TCP/IP) and click on Properties. Then click on the Advanced button and select the WINS tab. Next, select the 'Disable NetBIOS over TCP/IP' and click OK:



# Part 3 – The Incident Handling Process

## 3.1 Preparation

As our company's Virus Incident Response Team (VIRT) lead, I was responsible for making sure procedures were in place that dealt with Virus Incidents. I created standard operating procedures for e-mail and network-based viruses based on the ever-popular 6 step incident handling approach. As the lead, I need to be available 24x7, hence I carry a pager, as does my back-up. I also carry a laminated card with contact information for all the members in the Incident Response Team in my wallet.
Our Company Security Policies has a section dedicated to Virus Protection, excerpts of which are shown below:

- Electronic mail must be scanned when it enters and leaves the corporate network (i.e., to/from the Internet or to/from a business partner or vendor).

- The scans must occur without user intervention.
- Exception: On systems where this is not possible, the user is responsible for initiating the scan on desktop systems, laptop systems and single user workstations and is accountable for ensuring that the scans are performed; the system administrator is responsible and accountable for servers.

- Users shall not disable automatic virus scanning.

- The following table contains the responsibility for updating version, engine, and signature files:

| Desktop / Laptop PCs (LAN Connected) | File and Print Servers | Collaboration, Groupware, FTP and E-mail Servers | Stand-alone PCs and Laptops (Never network connected) |
|---|---|---|---|
| Site Administrator | Server Administrator | Server Administrator | User |

- New versions of the virus signature files must be loaded within one (1) week of release by the vendor.
- New releases of the Anti-Virus scanning engine must be installed within four (4) weeks of release by the vendor.
- New features for the software package must be installed within ninety (90) days of release by the vendor.

- Anti-Virus Software Configuration:
- All computer systems processing information or accessing networking resources must have current, approved Anti-Virus software installed, properly configured and running at all times (never disabled).
- The “heuristic” scanning property must be enabled.
- The Anti-Virus software must be configured to automatically clean the infected file (i.e., clean / fix the virus) or to quarantine the infected file if automatic cleaning is not possible.

Additional Company policies include Acceptable Internet/E-mail use. Even though it is better to rely on technical preventive measures, having these policies in place will protect you in case of any legal action. Examples of such policies are:

- Users are prohibited from installing unauthorized software on Company computers.
- Users’ access, and the privileges associated with such access, must be limited to those needed for performing job requirements.
- Access to the Internet must be through a Company approved Firewall.
- Users must be individually identifiable prior to being granted access to the Internet
- Users must ensure that precautions are taken to protect Company networking and computing resources when obtaining software, files and data from the Internet.

Hardening your own network is always challenging, but should be an on-going process. Knowing where your network shares are will allow you to properly secure them, preventing viruses like Bugbear from spreading through network shares. Since the Bugbear virus uses a static high port to enable its key logging and Trojan backdoor access, it is easy to protect yourself against that particular part. Don’t stop there. Check all your ports. Only enable the ones that are required, and block all the other ones. Network Intrusion Detection and Host-based Intrusion Detection applications are essential in keeping your environment secure, and unfortunately, both are missing from our network at this point.
In addition to the Anti-Virus responsibilities, a good relationship with your Anti-Virus vendor is essential as well. The notifications in the Introduction of this paper are an example of how proactively setting up a notification method can help with being up to date on current events. If possible, find out if you can be put on a contact list that the vendor will use when they change their status of a virus. Different levels of severity could have different contact means as well. You could receive e-mail when the status is Low, an e-mail and voicemail if the status went to Medium, and e-mail, voicemail, and a page if the status were to go to High. Our virus Incident Response Team consists of members of the Information Security team, members of the Networking Deployment and Operations groups, the E-mail Deployment and Operations groups, and the Desktop Deployment group. We try to have at least two representatives of every group in order to

provide back up when someone is unavailable. The Global Tier 2 Support Team has the list of office phone numbers, home phone numbers, and pager/cell phone information for all members of the Virus Incident Response Team. Also, we had collected a list of virus contacts at each affiliate company, someone whom we could contact as a central person in case of virus outbreaks, but also to hold responsible for communicating any virus alerts that we release to them.

Another factor that is vital for a Virus Response Team is upper management support. You need to make sure that you can use other managers' resources, you need to have ways to escalate problems up the ladder when you have problems getting in touch with people, and you will need to create metrics to show upper management how important having a proper Incident Response process is. The process is our Response Team's primary tool. Because of the size of the Company, having a process, and having the right people in the right places, are the best tools for tackling security incidents.

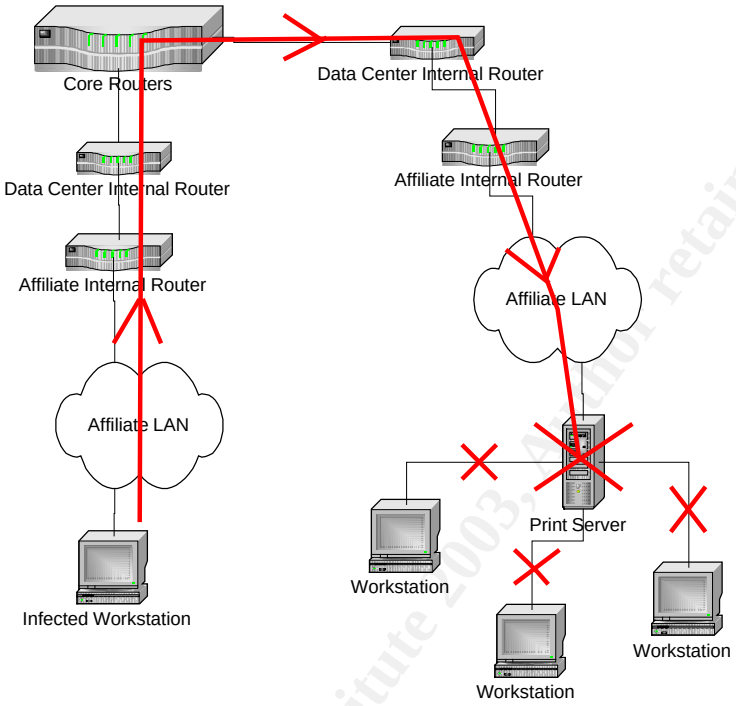
In general, management is of the opinion that containment and eradication is more important than gathering information and prosecuting. Of course, this depends heavily on the type of security incident. Viruses and worms need to be stamped out quickly, and no investigation as to the source is generally required. Spam or abusive e-mail, however, as well as unauthorized access attempts will prompt an immediate investigation to the source. As mentioned, management support is an essential part of escalation. When a user or a helpdesk is unresponsive, taking matters up the ladder will usually generate some response. In our case, we have our VP's support, and we are free to contact him if after 48 hours, no progress is made in eradicating the infection (we can contain from a central location, eradication needs to happen on-site). Management should also be kept informed at regular intervals as to what is going on with the infection. If only a handful of devices are infected, this could be daily, if the infection is more widespread, hourly or two-hourly updates might be required. You should coordinate with your own manager as to what your authorities are when it comes to escalating matters; either go through him or her, or, as a Virus Response Team lead, you escalate up to higher levels of management directly.

In our environment, when a virus infection is detected, a user will generally first contact their helpdesk, who will then escalate to the VIR Team. The VIR Team lead or their delegate remains the primary point of contact during the entire process. Although other groups will be involved in either Containment or Recovery, having that single point of contact will prevent the process from going off-track, and will allow a more streamlined approach.

## 3.2 Identification

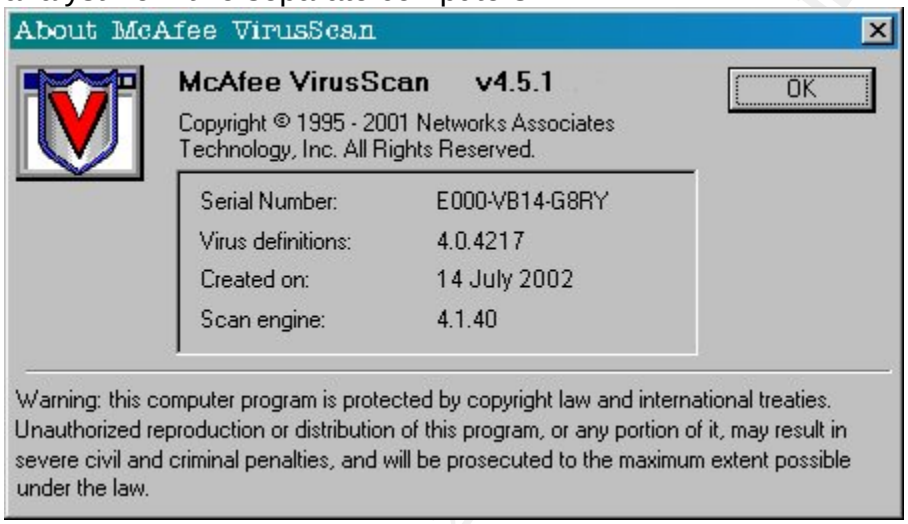
In our situation, a user that did not have the proper level of Anti-Virus protection got infected with the Bugbear virus. The exact cause of this infection was never determined, but it seems highly likely that the user downloaded something off the Internet and got infected that way. The fact that the Webshield and Exchange mail servers had been updated to the latest virus definition files meant that

further infection inside the corporate network through e-mail was cut off at the Exchange servers and infection of systems through e-mail outside the corporate network was blocked at the Webshield servers. It also meant that infection through e-mail was highly unlikely.
The only impact that we experienced from this virus was printers spooling print jobs across states, countries, and even continents. The amount of data contained within the code flooded the print servers' mail queues within seconds, effectively performing a Denial Of Service (DOS) attack against the users at the site, since legitimate users at the affiliate sites were unable to print. This is shown in the following diagram:




The Bugbear virus attempts to print the encrypted contents of the worm itself, resulting in hundreds of pages of garbage being printed. Our affiliate in California saw print jobs coming in from Brazil and Belgium. Our affiliate in Florida saw print jobs originating in Malaysia, and a Pennsylvania operating company had users from Australia printing to their network printers.
In all cases, the local helpdesks were notified immediately, after which the situation was escalated to the Virus Incident Response Team. The Team lead then contacted the Virus contact and Information Security Officer (ISO) at the site that was affected to determine the impact of the infection. The names in the print spool windows showed the names of the users who were attempting to print these documents. These users were quickly identified as not being located at that affiliate or even in the same country. Based on the information that had become available on the Anti-Virus vendor sites, the suspicion that this could be the Bugbear virus became rather strong.

Since we had upgraded all our Webshield and Groupshield servers to the latest Virus Definitions, we were able to determine that the virus had entered our environment in some other fashion. To further verify this, we looked at the Webshield and Groupshield logs and saw many entries indicating that a message was quarantined because it was infected with the Bugbear virus. We knew that our protection at the Mail Gateways was working properly. The only way we could determine how they became infected, was by visiting the users' desks and looking at their computers. The local helpdesk personnel at the involved sites did so, and discovered that the users had either built development machines and not installed Anti Virus software, or had disabled virus scan on their workstation, resulting in outdated, non-existing, or non-operating Anti Virus Software. Here are two screenshots that were taken by the local helpdesk analyst from two separate computers:

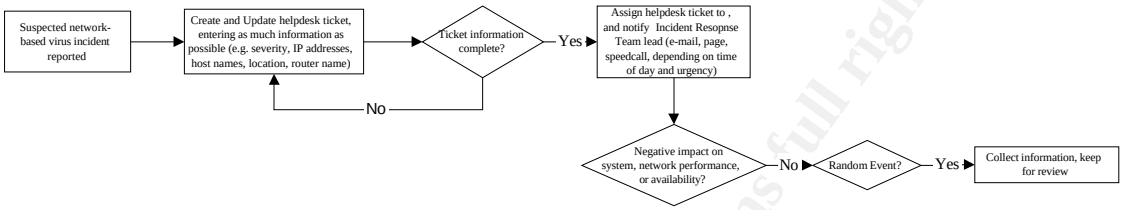



**Outdated Scan engine and Virus definitions.**


**No AV software running at all**

Scanning for the virus was inconclusive, since the virus can disable the virus scan engine, resulting in false positives. Consequently, the Stinger virus tool was run (see section 3.4), confirming the virus to be Bugbear.
For each of the incidents, the helpdesk created a helpdesk tracking ticket, and correlated the ones that showed similar patterns. This helped the Incident Response Team to find patterns in the infections, and also ensured that all critical information was stored in a central location.
Other ways in which we would have been able to identify the Bugbear virus would have been to look for the presence of the files mentioned in section 2.3. If an executable file had been present in Registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Once, that would have been an indication of an infection. Also, doing port scans

or running the netstat command, as described in section 2.5, would have confirmed this virus to be Bugbear if port 36794 had been opened.
As a sanity check, we contacted the sites that had seen print jobs coming from the suspected users' workstations to make sure that we had in fact identified the culprits correctly (we had in the meantime removed their network access). Fortunately, no further print jobs were being spooled to the print servers, verifying that we were on the right track.
The diagram below shows the steps in the Identification phase we have set up for Virus Incidents




## 3.3 Containment

The affiliate's immediate response was to clear out the print queues, so the users at the sites that were affected could print again. This, of course, did not contain the virus at the site where the actual infection was located. The Information Security Officer at the infected site was asked to locate the user's workstation, disconnect them from the network, and do a full system virus scan. Also, to prevent the backdoor Trojan part of the virus from working, both inbound and outbound traffic was blocked on the Firewalls at the Data Center, blocking any tcp traffic attempting to go through port 36794 with the following rule:

access-list BD-acl deny tcp any any eq 36794

One of the sites that was affected was located in California. Two users from Belgium flooded their print server, which serves 1500 users, with print jobs. Their initial response was to remove their site from the Corporate WAN. This impacted their site in that they could not access other affiliates, nor could other affiliates access them. Effective though this solution might have been to resolve the immediate problem, it wasn't the best response. After the initial knee-jerk reaction had passed, they decided to block access to the print server from anywhere outside of that particular site itself. That way, the users inside the affiliate LAN could use their printer again, they were back on the corporate network, and the only thing that was affected is that other sites couldn't print to printers at the California site. Normally, there would rarely be a need to print to network printers at other locations, but in this site's case, this is a business requirement.

The Groupshield and Webshield servers were properly protected by the latest virus engine and dat file versions so there was little else that we could do on that end. If the virus definitions had not been in place, or had proven to be ineffective, we would have had some additional options, such as blocking file attachments with the .exe extension, or files that were 50,688 bytes large. Filtering mail messages by subject line would not have been effective, since the large number of different subjects would impact performance, and the list also includes some standard subject lines that would have blocked legitimate e-mail messages. Having the proper Anti-Virus protection, however, was by far the best solution, since in the next two weeks, over 11,000 e-mail messages coming in from the Internet were found to be infected with the Bugbear virus, all of which were intercepted by the Webshield servers.

Ultimately, the containment process was to locate the infected workstation, remove it from the network, and clean the infection. If a site did not respond to a request within 2 hours, or the device could not be located, the Network group was asked to isolate that workstation from the network by implementing an access list on the router, blocking any traffic to and from the infected device's IP Address by telnetting into the router, applying an access control list, and then applying that access control list to the inbound Ethernet interface on that router:

Simple ACL:

access-list *xx* deny *ip_address* (any traffic from that device)

More complex ACL:

access-list *xx* deny ip *ip_address mask* (any IP traffic from that device or subnet)

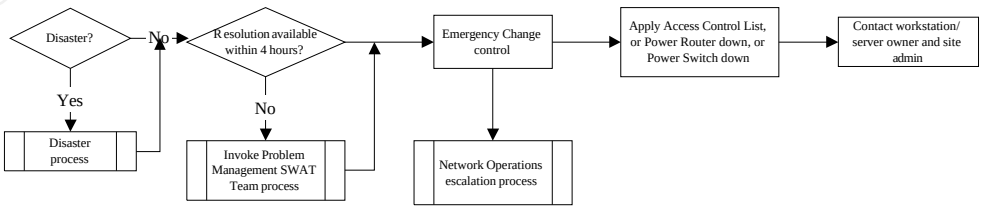
Implementing this ACL on an affiliate router does not contain a worm completely, as it can still spread inside the LAN, but the rest of the Company network can't be compromised. Management approved this approach after Nimda brought down the Company's Headquarters for 2 days and infected a dozen other affiliate sites. Containment is the single-most time sensitive step in our process; if a person's computer can't be isolated within 2 hours, the entire LAN is isolated.

Had management decided that evidence should be preserved in cases of virus outbreaks, back-ups would have been made of the infected computers, using PowerQuest DeployCenter, the same software our Company uses for imaging workstations, or Drive Image Pro from the same company. This is done by booting from a floppy disk with the PQ Drive Image Pro software installed, and then capturing the computer image to an external hard drive, an external CD/RW drive, or a network share. When creating an image to a network share, a network boot floppy disk must be used, of course. This is the procedure to create an image using Power Quest Drive Image Pro:

1. Restart the computer with the PowerQuest Disk 1 in drive A.
2. At the command prompt, remove the boot disk from the drive and replace it with the PowerQuest Disk 2.
3. Type pqdi and then press ENTER. Drive Image Pro loads and initializes the mouse driver, analyzes your hard disk, and then displays the PowerQuest Drive Image Pro 2.0 dialog box.

4. Click Create Image to start the Create an Image wizard. The Create an Image wizard displays the Select Source Partition(s) page, prompting you to select the source partition that you want to copy.
5. Click C: to select the installation that you just configured, and then click Next to continue. The Create an Image wizard displays the Name Image File page, prompting you to specify the name and path for the image file.
6. In the Image File box, type d:\myimage.pqi and then click Next to continue. The Create an Image wizard displays the Compress Image File page, prompting you to select a compression level for the image file.
7. Click High, and then click Next to continue. The Create an Image wizard displays the Ready to Create Image File page, prompting you to review your configuration settings for this imaging operation.
8. Click Finish to create the image file.
Drive Image Pro displays the Creating the Image dialog box, showing the progress of the imaging process. Drive Image Pro displays a message box when the imaging process is complete.
9. Click OK.
Drive Image Pro displays the PowerQuest Drive Image Pro 2.0 dialog box.

Log files would have been collected from the routers, firewalls and content engines as well for further analysis.
Our Virus Incident Response Team does not use a 'jump kit' for virus-related security incidents, because of the way our Company and our network is designed. The function of this team is mainly to have a central team coordinating the incident handling process, and to assist remote systems engineers, helpdesk analysts, and other technical personnel in locating and eradicating an infection. The remote extension of the Response team as it were has access to CDs with images of all desktop and server builds, CDs with the PowerQuest DeployCenter or Image Pro software in order to re-image a computer, as well as equipment to identify infected sources such as network sniffers. Other options would have been to have a clean version of the regedit.exe and netstat.exe file on either a floppy or CD in case there was suspicion of infection or Trojans. An effort is underway to have a complete jump kit for each region, which would include software as mentioned before, as well as hardware such as an extra hard drive, both internal and external, an external CD/RW drive, and some blank media. These jump kits do not exist at this point, but there is obviously a need for these. The diagram below shows the steps in our Containment procedure:

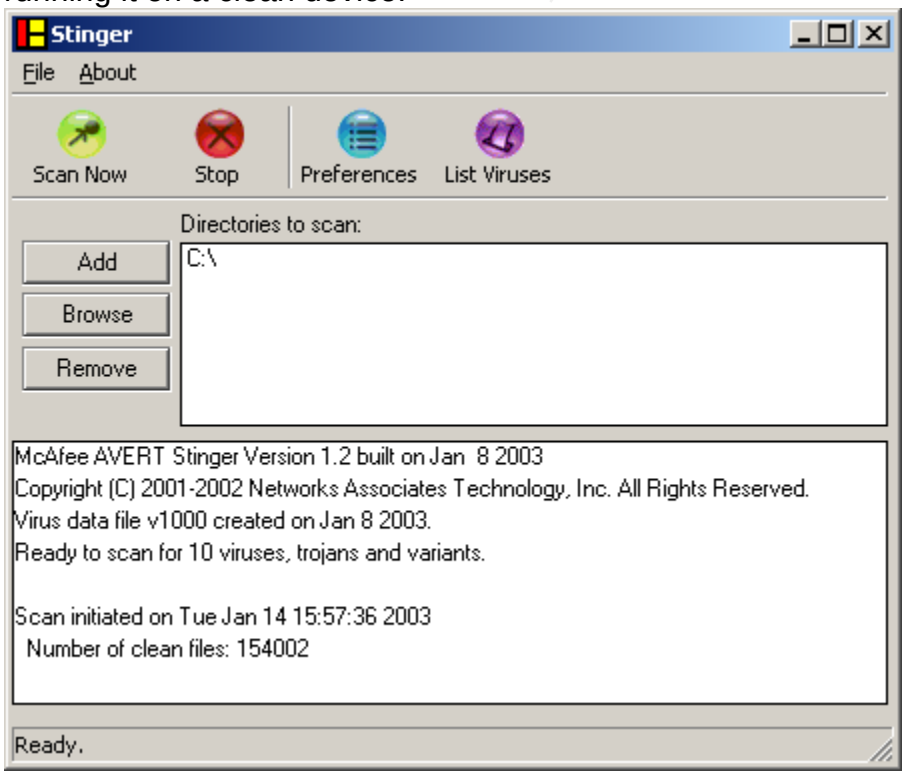



A Problem Management SWAT Team should include managers, ensuring that proper escalation takes place if a solution is not found within a certain timeframe,

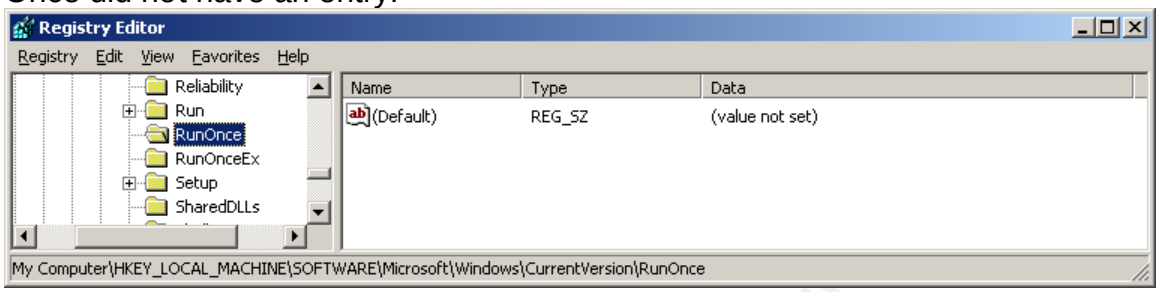
because they have the ability to gather and assign resources if required. This was discussed in detail in the Preparation section. A Disaster is indicated as complete loss of network connectivity due to a DoS attack (network traffic flooding, for example) or hardware failures due to exploits or DoS.

### 3.4 Eradication

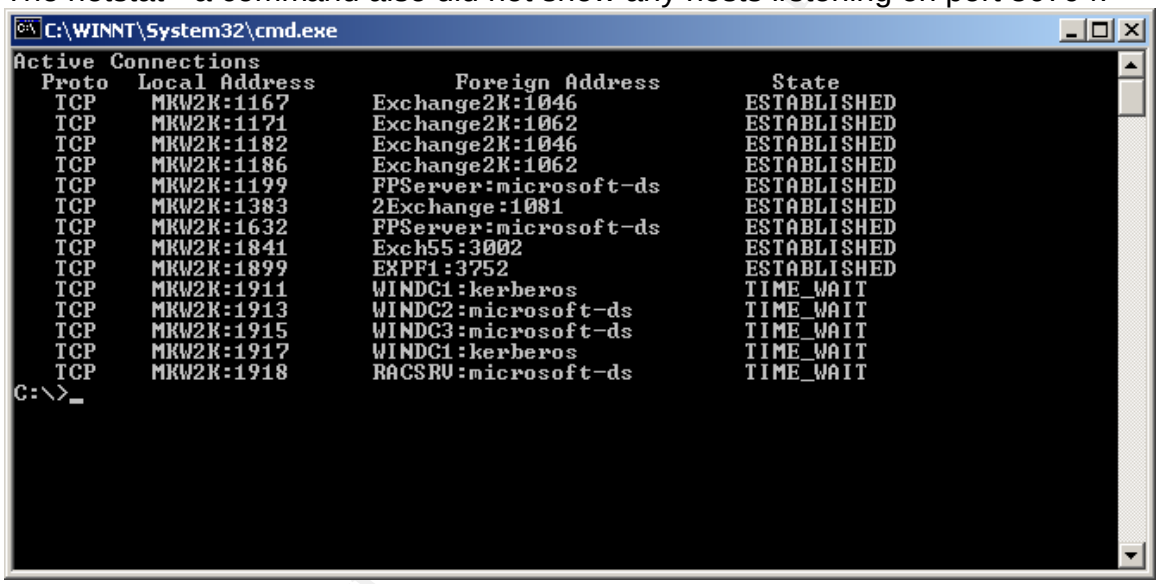
After the workstations/servers had been removed from the network, or otherwise isolated, the clean-up effort could commence. On the majority of the systems infected, it was clear that Anti-Virus software was disabled, not installed (development machines), or otherwise altered to not perform the weekly updates. This is, of course, against company policies, and the users were thus notified. Fortunately, the stand-alone tools released by the Anti-Virus vendors, such as McAfee's Stinger, get around the disabling of the Anti Virus software processes caused by Bugbear. These types of programs do not attempt to clean the infected files using the Anti-Virus software's built-in engine. Instead, they look for the signatures the virus leaves behind and restore the system. In this case, Stinger first stops all the worm's threads, it looks for the dll and dat files that the Trojan dropped, the registry entry that gets created to run upon reboot, and the executable files that are the Bugbear virus files themselves. The tool then deletes all these files. These types of tools are not full-featured virus scanners, but generally only scan for specific viruses. This is a screenshot of Stinger after running it on a clean device:



After the stand-alone tool had run, a full system scan was done again to ensure that the device was clean. Additional signs that the infection had in fact been eradicated were the fact that the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Once did not have an entry:

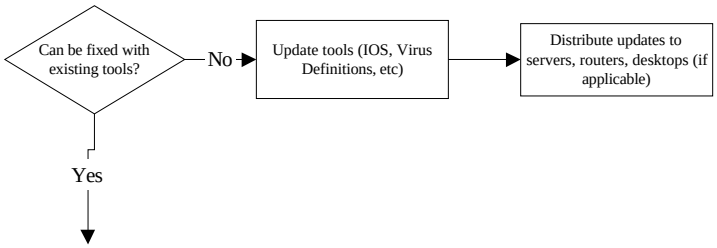



The netstat -a command also did not show any hosts listening on port 36794:




When the assurance that a system was in fact completely clean was less than conclusive, the device was re-imaged, meaning a complete re-partitioning and reformatting of the hard drive, and reinstallation of all applications was performed. The determination whether or not a device was 'conclusively clean' was left up to the helpdesk engineer. If they did not have access to the stand-alone tool, then there was no doubt that the device had to be re-imaged. Even if the stand-alone tool indicated that the device was not infected, a 'better be safe than sorry' type technician could decide to re-image the device anyway. This re-imaging is done by the local affiliate's helpdesk, using either a network share from which the helpdesk technician pulls down an image, or a CD that has a standard image loaded. The program used for this imaging and re-imaging is PowerQuest DeployCenter. For security purposes, the CD was the best solution in this case, since an infected computer connecting to the network to receive a new image could possibly still infect other hosts or become infected before the Anti-Virus software was loaded and brought up to the latest version.

After all these steps were complete, Anti-Virus software was loaded or reloaded, and brought up to date on scan engine and definition file level.
By default, the Exchange servers and Webshield servers are updated with the latest Virus definition files within 24 hours of their release. In this instance it was no different, which greatly reduced the chances of an infection. The virus protection software on the e-mail servers eradicated any viruses that tried to enter the network environment through e-mail.
These are the Steps involved with the Eradication phase:




## 3.5 Recovery

Once the infected systems had been cleaned or re-imaged, and had updated virus software loaded, the environment could be normalized again. This included removing all the access control lists from the routers at affiliate sites where there had been an infected user. All that was required to do this was remove the line:
access-list *xx* deny *ip_address*
from the ACL with the number *xx*.
In theory, we could have removed the rule from the Firewall as well. However, since port 36794 is a non-essential port, the rule could safely be left on the Firewall. It depends on a company's policies whether or not this rule is actually necessary in the first place. A lot of organizations block all ports except the essential ones, a good security practice. If the California site had not needed their print server to be accessible by other sites, blocking TCP traffic to the print server from outside that site would have been a good permanent solution. A company can make the determination that only users within the site can access a print server, and therefore only allow internal traffic to that server. Now, in the California site's case, as mentioned, this solution was not an option, and it's difficult to determine what would have worked. This goes back to the all-or-nothing situation NetBIOS puts a system administrator in mentioned in part 2.2. Since the Bugbear print traffic could have come from a site with which the California affiliate does business, restricting access to the print server by IP address would have needed a solution. Adding the appropriate Access Control List to the router, only allowing certain subnets access to that print server's IP address, would limit the amount of rogue traffic flooding that printer. This ACL would have accomplished that:
access-list 101 permit ip 10.20.30.0 0.0.0.255 host 192.168.1.1
access-list 101 permit ip 11.22.33.0 0.0.0.255 host 192.168.1.1
access-list 101 deny ip any host 192.168.1.1
access-list 101 permit ip any any

Where 10.20.30.0 and 11.22.33.0 would have been the subnets of sites given access to the printer, which has the IP Address 192.168.1.1.
One of the other options could have been to look for print jobs of a specific size (50,688 bytes, the size of the worm) and block those using an IDS.
This virus infection resulted in some virus reports that had not been run in a while, specifically reports on virus engines. McAfee informed users that having the latest virus definition files was not enough to counter the Bugbear virus. The virus engine had to be upgraded as well, to at least 4.1.60. The reports that were run showed a large number of servers that were one version behind on virus engine. Part of the Recovery effort included ensuring that every computing device was up to date on software and configuration files. These types of reports should be an on-going process.
Suggestions were made to the Desktop Development team to test setting the permissions on the Run and RunOnce registry keys as described in section 2.6, and that is currently being investigated.
Once the environment had been returned to normal, all previously infected workstations which had been put back on the network received 'special attention' from all system administrators and helpdesk personnel at the respective sites for the next 2 days.
The following shows the steps in the Recovery Phase:


## 3.6 Lessons Learned

The Bugbear virus incident within our Company was definitely noticeable, but thanks to a strong Anti-Virus policy, the impact was rather minimal. Overall, home users were much more at risk for this particular virus than corporate users. The most important thing to remember, which is reflected in the section size of this chapter, is that Preparation is the single most important part of a virus incident process. You will never be 100% protected, but 99% will save you a lot of time and headaches (and potentially, your job).
Reporting is the only way to catch the 1% that falls outside of Company Policies, and it is much more realistic to expect 0.9% than the full 1%. Users can get very creative to ensure they are not reported on. Use whatever means you have to report on virus protection level, as well as virus incidents. Use scripts, log file parsing tools, or full-blown Anti-Virus Management applications, as long as they give you an indication of how your environment is doing. Address the devices that are not up to the current standard with a heavy hand. Non-compliance is not an option. Use the reports that show virus incidents to indicate trends, but also to impress upon upper management that they need to know the money they spend on Virus Protection solutions is not wasted.
In our case, our reporting capabilities were less than they should have been, resulting in far more devices that were behind on virus software engine or

definition file than we anticipated. All these devices are time bombs waiting to go off. Developers might see Anti-Virus software as an annoyance on their test computers. As long as those devices are restricted to an isolated development environment this does not preset a problem. However, these precautions are not always taken. Therefore, user awareness is part of the Preparation phase as well. It is wise to devise strategies to raise user awareness. Show some details to the regular users, point to articles discussing the fate of less fortunate companies who got hit hard with a virus, whatever works.

The incidents described here were all similar. A user got infected by not following company policies, the Bugbear virus tried to spread through e-mail and failed, through network shares and failed, and then tried to print the contents of its code to network printers, causing massive amounts of paper to be wasted. The key logging/backdoor Trojan was rendered powerless by blocking access to and from port 36794. Had that not been done, company sensitive information could have been stolen, causing a major impact the results of which are not easy to ascertain. Depending on the type of business you're in, this could range from a brownie recipe being transmitted from one user to another, to top-secret government information being stolen from the system.

As mentioned before, a more restrictive use of the Internet and/or e-mail would have prevented some of the infection. A lot of what you will be able to implement in this area will depend on your company's working environment. You could implement a list of acceptable websites that people could use for legitimate business reasons. You could also institute a known 'bad site' list. However, since that's most of the Internet, those lists could become too long to manage efficiently very quickly, and the proliferation of new web sites every day makes administration of such a list a daunting task. Perhaps you trust people not to spend 50% of their work time on the Internet, and are confident that they would never visit inappropriate web sites, and place no kind of restrictions on Internet use at all. The users will undoubtedly appreciate this policy, but you would have to prepare yourself for the worst, and be very sure that other protection methods are in place.

Part of a Virus Incident outbreak should always be a Lessons Learned meeting, sometimes called Post Mortems. Get every party that was involved in one room or on a conference call and discuss what happened. Don't point fingers and make accusations; doing that is counter-productive. Discuss every step of the Incident Handling process, starting with the preparation. Where did the preparation fail, what could have been done to prevent the infection in the first place, and what needs to happen now? Do the same for the Containment, Eradication, and Recovery phases. What could have been done to speed up the process to Recovery?

Other preventive methods have been discussed in the previous sections, but to recap, these are the recommendations for preventing similar incidents in the future:

- Ensure up to date Anti-Virus software and definition files are installed on ALL devices accessing the network

- Have Corporate policies for Anti-Virus, Acceptable Internet/e-mail use, and network and server security settings
- Disable all non-essential ports on the Firewall
- Disable all non-required services
- Block dangerous attachments at the Mail Gateways (.exe, .vbs, .scr and any other executable file extentions)
- Tighten access control to network resources such as file shares and network printers
- Keep the Operating Systems and Applications patched
- Follow patch information from industry leaders
- Implement both Network and Host-based Intrusion Detection systems

### 3.7 Additional Information and Conclusion

Many resources are available on the Bugbear virus, some of which are:
CA Bugbear Information Center:
*http://www3.ca.com/virusinfo/virus.asp?id=13233*
F-Secure Global Bugbear Worm Information Center:
*http://www.f-secure.com/bugbear/*
Internet Security Systems Alerts:
*http://bvlive01.iss.net/issEn/delivery/xforce/alertdetail.jsp?oid=21301*
Sophos Virus Analyses:
*http://www.sophos.com/virusinfo/analyses/w32bugbeara.html*
McAfee Bugbear Information:
*http://vil.nai.com/vil/content/v_99728.htm*
Symantec Bugbear Information:
*http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html*

Resources describing the IFrame vulnerability are:
Microsoft Security Bulletin:
*http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp*
SecurityFocus Bugtraq:
*http://online.securityfocus.com/bid/696/credit/*

SANS offers a policy template for Anti-Virus practices:
*http://www.sans.org/newlook/resources/policies/Application_Service_Providers.pdf* (PDF Format)
*http://www.sans.org/newlook/resources/policies/Anti-virus_Guidelines.doc* (Word format)

Additionally, a list of other resources used in this paper can be found in the References section.

Viruses are here to stay, and what's worse, they're getting more and more insidious. The Bugbear virus is a sign of things to come. Viruses attacking computer systems on multiple fronts are the wave of the future. Because of this, when creating a Virus Incident Response team, be sure you have representation from enough different disciplines such as desktops, servers, e-mail, and networking. It should be emphasized again: Preparation is the most important step of the Virus Incident Response process, but once you are infected with a virus, having the right people in the right places will allow for a speedy road back to recovery.
Good luck!

## Appendix A - List of security program processes that Bugbear attempts to stop

ACKWIN32.exe
F-AGNT95.exe
ANTI-TROJAN.exe
APVXDWIN.exe
AUTODOWN.exe
AVCONSOL.exe
AVE32.exe
AVGCTRL.exe
AVKSERV.exe
AVNT.exe
AVP32.exe
AVP32.exe
AVPCC.exe
AVPCC.exe
AVPDOS32.exe
AVPM.exe
AVPM.exe
AVPTC32.exe
AVPUPD.exe
AVSCHED32.exe
AVWIN95.exe
AVWUPD32.exe
BLACKD.exe
BLACKICE.exe
CFIADMIN.exe
CFIAUDIT.exe
CFINET.exe
CFINET32.exe
CLAW95.exe
CLAW95CF.exe
CLEANER.exe
CLEANER3.exe
DVP95_0.exe
ECENGINE.exe
ESAFE.exe
ESPWATCH.exe
FINDVIRU.exe
FPROT.exe
IAMAPP.exe
IAMSERV.exe
IBMASN.exe
IBMAVSP.exe

ICLOAD95.exe
ICLOADNT.exe
ICMON.exe
ICSUPP95.exe
ICSUPPNT.exe
IFACE.exe
IOMON98.exe
JEDI.exe
LOCKDOWN2000.exe
LOOKOUT.exe
LUALL.exe
MOOLIVE.exe
MPFTRAY.exe
N32SCANW.exe
NAVAPW32.exe
NAVLU32.exe
NAVNT.exe
NAVW32.exe
NAVWNT.exe
NISUM.exe
NMAIN.exe
NORMIST.exe
NUPGRADE.exe
NVC95.exe
OUTPOST.exe
PADMIN.exe
PAVCL.exe
PAVSCHED.exe
PAVW.exe
PCCWIN98.exe
PCFWALLICON.exe
PERSFW.exe
F-PROT.exe
F-PROT95.exe
RAV7.exe
RAV7WIN.exe
RESCUE.exe
SAFEWEB.exe
SCAN32.exe
SCAN95.exe
SCANPM.exe
SCRSCAN.exe
SERV95.exe
SPHINX.exe
F-STOPW.exe
SWEEP95.exe

TBSCAN.exe
TDS2-98.exe
TDS2-NT.exe
VET95.exe
VETTRAY.exe
VSCAN40.exe
VSECOMR.exe
VSHWIN32.exe
VSSTAT.exe
WEBSCANX.exe
WFINDV32.exe
ZONEALARM.exe

## Appendix B – Possible Subject Lines

Greets!
Get 8 FREE issues - no risk!
Hi!
Your News Alert
$150 FREE Bonus!
Re:
Your Gift
New bonus in your cash account
Tools For Your Online Business
Daily Email Reminder
News
free shipping!
its easy
Warning!
SCAM alert!!!
Sponsors needed
new reading
CALL FOR INFORMATION!
25 merchants and rising
Cows
My eBay ads
empty account
Market Update Report
click on this!
fantastic
wow!
bad news
Lost & Found
New Contests
Today Only
Get a FREE gift!
Membership Confirmation
Report
Please Help...
Stats
I need help about script!!!
Interesting...
Introduction
various
Announcement
history screen
Correction of errors
Just a reminder

Payment notices
hmm..
update
Hello!

## References


F-Secure Website
http://www.F-Secure.com

Description of the iFrame vulnerability – Michael Chait, September 2002
http://www.internetnews.com/dev-news/article.php/1459341

Yet another Bugbear signature – Vjay Larosa, 10 October 2002
http://www.e-secure-db.us/dscgi/ds.py/Get/File-12349/RE_Snort-sigs_Yet_another_BugBear_signature.txt.

Symantec WW32.Bugbear@mm description – September 30 2002
http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html

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

Sophos W32/Bugbear-A
http://www.sophos.com/virusinfo/analyses/w32bugbeara.html

Microsoft Description of iFrame Exploit – March 29 2001
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp