



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at <http://www.giac.org/registration/gcih>

GCIH

Practical Assignment Version 2.1a

Option 1 – Exploit in Action

© SANS Institute 2003, Author retains full rights.

Virus Analysis and Incident Handling: A Case Study

Lin Han

June 2, 2003

Abstract

With new exploits discovered and high exploration techniques developed, viruses become more and more sophisticated and no computer seems to be immune to them. Incident handling together with defense is playing a key role in the anti-virus battle. This paper presents a brief introduction to computer viruses, and analyzes in detail the characteristic, behavior and impact of a mass-mail virus called W32/Yaha.g@MM. It provides a case study to demonstrate step by step how a financial institution successfully handled the incident when W32/Yaha.g@MM virus broke out. The purpose of the paper is to raise the awareness about the importance of incident handling in time-based security.

1 Introduction

Computer viruses are an ongoing threat. New viruses are discovered every day, and thousands are processed each month [1]. By October 2002, the total number of known viruses had surpassed 80,000. Computer viruses now rank as the number one killer of computers in the world [2].

Infection by a computer virus can be devastating. According to a survey by Central Command, an anti-virus software vendor, the majority of personal computer users had experienced at least one virus in the twelve-month survey period, and over half of these infections resulted in a loss of data [2]. Each year, companies spend millions of dollars due to lost productivity and clean-up expenses [6].

With such numbers, it's safe to say that most organizations will regularly encounter virus outbreaks. As new exploits and techniques are discovered continuously, no one who uses computers is immune to viruses.

While up-to-date anti-virus software provides an effective defense against viruses, incident handling plays an important role once a system or a network gets infected. A good incident handling process helps in improving the security of a site by reducing both the detection time and the reaction time.

This paper presents a detailed analysis of the characteristic, behavior and impact of a mass-mail virus – W32/Yaha.g@MM. A case study demonstrates a successful incident handling process in a financial institute when the organization was hit by W32/Yaha.g@MM virus. The paper comes to the conclusions that incident handling has no end – by improving the procedure itself and enhancing security defenses, one incident handling process initiates the next one.

This paper is submitted in partial fulfillment of the requirements for the GIAC Certified Incident Handling Analyst.

The rest of the paper is organized as follows. Some background information, such as the profile of W32/Yaha.g@MM virus and the network infrastructure of a financial institute, is presented in section two. Section three analyzes W32/Yaha.g@MM virus in detail. In section four, an incident handling case study is provided. Finally, section five concludes the paper.

2 Background

Computer viruses first came on the scene around 1981, when a program called “Elk Cloner” was created for the Apple IIe computer. It appeared on some bulletin board systems (BBSs), and never was referred to as a “virus” [3].

On November 3, 1983, the first real virus was conceived of as an experiment to be presented at a weekly seminar on computer security. The concept was initially introduced in this seminar by Fred Cohen, and the name “virus” was thought of by Len Adleman. One week later, the virus was demonstrated on a VAX 11/750 system running Unix [4].

A virus is defined as “a computer program file capable of attaching to disks or other files and replicating itself repeatedly, typically without user knowledge or permission” [5]. Viruses are parasitic in that they need to attach themselves to files (or boot sectors) and then replicate, continuing to spread.

In addition to replication, a virus usually contains a virus payload [6]. Some payloads only display messages or images, while others destroy files, reformat hard drives, or cause other damages. Even if a virus does not have a payload, it can also cause trouble by consuming storage space and memory, and degrading the overall performance of the infected computer.

Several years ago, computer viruses spread extremely slowly and were geographically limited as a result of the fact that computers were not well connected and files were primarily transmitted via diskettes or BBSs. The advent of computer networks introduces new virus distribution mechanisms, and the boundary of computer viruses has been widened. With the extensive use of e-mail, viruses are propagating faster than ever. Viruses attached to e-mail messages can infect an entire enterprise in a matter of minutes. Today, an overwhelmingly large proportion of infections are caused by infected e-mail attachments [2].

2.1 Virus Profile

W32/Yaha.g@MM (named by McAfee) virus was first discovered on June 15, 2002 [7], and became widespread on June 20, 2002 [8]. Figure 1 depicts the virus infection since June 17, 2002 [9].

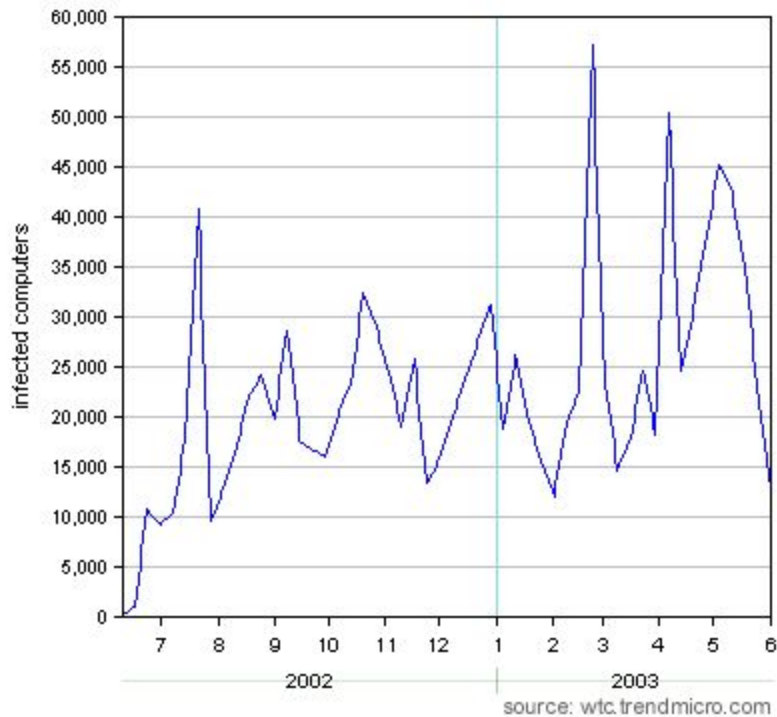


Figure 1 Computers Infected Since June 17, 2002

The virus is also known with the following names [10][11][12][13][9][8][14].

- W32.Yaha.F@mm (by Symantec)
- W32/Lentin.E (by Panda)
- W32/Yaha-E (by Sophos)
- W32/Yaha.E (by Central Command)
- Worm/Lentin.F (by Central Command)
- WORM_YAHA.G (by Trend)
- Yaha.E (by F-Secure)
- I-Worm.Lentin.G (by F-Secure)
- Win32.Yaha.E (by CA)
- Yaha.E@mm
- Yaha
- Worm/Yaha.E
- Win32/Yaha.E.Worm
- Win32/Yaha.E
- Win32/Lentin.F@mm
- W32/Lentin.F@mm
- Lentin.G
- Lentin
- I-worm/Yaha.D
- I-Worm.Lentin.f

Many of the above names start with “W32” or “Win32”. This signifies that the virus is a 32-bit Windows file infector and only infects 32-bit Windows operating systems. All such Windows systems are vulnerable, including

- Windows 95
- Windows 98
- Windows NT
- Windows 2000
- Windows XP
- Windows Me

Macintosh, Unix and Linux are not vulnerable [10].

W32/Yaha.g@MM randomly exploits a known vulnerability in Internet Explorer-based e-mail clients (refer to CVE-2001-0154 [15]), which allows e-mail attachments to be executed automatically on an unpatched system.

Once a system gets infected, the virus attempts to make itself run each time another executable file starts, send a copy of itself to all e-mail addresses found on the infected system, probe the web site <http://www.pak.gov.pk> several times a minute, turn off various anti-virus and firewall protection on the system, and kill Windows Task Manager process.

The virus affects certain versions of Microsoft Internet Explorer if the “Incorrect MIME Header Can Cause IE to Execute E-mail Attachment” vulnerability is exploited [16].

- Microsoft Internet Explorer 5.01
- Microsoft Internet Explorer 5.01 with Service Pack 1
- Microsoft Internet Explorer 5.5
- Microsoft Internet Explorer 5.5 with Service Pack 1

W32/Yaha.g@MM is a modification of Worm/Lentin (Valentine.scr), an Internet worm that spreads by retrieving e-mail addresses from the Windows Address Book and cached web pages (HTM, HTML and HTA files). In addition to these methods, W32/Yaha.g@MM can also spread through contacts found in the MSN Messenger list, Yahoo! Pager contact list, and ICQ database list [13].

2.2 Network Diagram

IFI is a financial institution, which belongs to a big financial group. Its network is a typical medium-sized enterprise network spanning several offices, with a number of Internet gateways, LANs, WAN links, and many servers (e.g., mail servers, DNS servers, web servers and database servers). Figure 2 illustrates the simplified network infrastructure in the whole financial group.

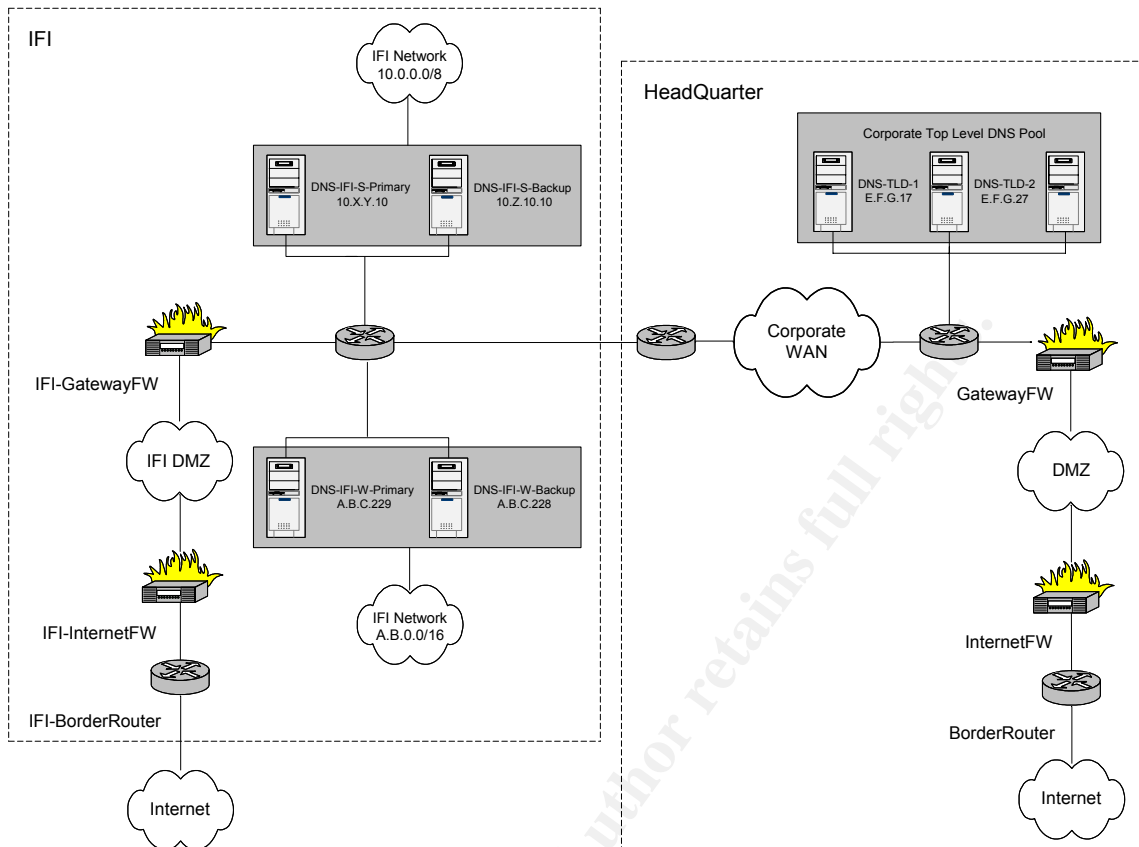


Figure 2 Simplified Network Infrastructure

Information on hardware, operating systems and services of the main network components is listed in the following table¹.

¹ According to the confidentiality policy in the financial group, detailed information about critical network components is not allowed to provide.

	Hardware	Operating System	Service/Application
DNS-IFI-S-Primary	Sun4u Ultra 10	Solaris	Cisco Network Registrar
DNS-IFI-S-Backup	Sun4u Ultra 10	Solaris	Cisco Network Registrar
DNS-IFI-W-Primary	Dell Power Edge 2550	Win2K Server	Cisco Network Registrar
DNS-IFI-W-Backup	Dell Power Edge 2550	Win2K Server	Cisco Network Registrar
DNS-TLD*	Sun4u E220	Solaris	BIND/DNS
IFI-GFW	Sun4u E250	Solaris	Checkpoint FW-1
IFI-InternetFW	Sun4u E250	Solaris	Checkpoint FW-1
GFW	Sun4u E250	Solaris	Checkpoint FW-1
InternetFW	Sun4u E450	Solaris	Checkpoint FW-1
IFI-BorderRouter	Cisco 7120 Series Router	Cisco IOS 12	
BorderRouter	Nortel BCN Router	Site Manager 14	

* DNS-TLD – Corporate Top Level DNS

Table 1 Information on Main Network Components

Due to the huge amount of information, both IFI DNS servers and the corporate top-level DNS pool are configured not to log standard DNS queries and responses.

DNS forwarders in IFI are defined as follows.

DNS Server	1st Forwarder	2nd Forwarder	3rd Forwarder	4th Forwarder
10.X.Y.10 (DNS-IFI-S-Primary)	10.M.N.42	E.F.G.17	A.B.C.229	A.B.C.228
10.Z.10.10 (DNS-IFI-S-Backup)	10.X.Y.10	A.B.C.229	10.M.N.42	A.B.C.228
A.B.C.229 (DNS-IFI-W-Primary)	10.M.N.42	E.F.G.27	10.X.Y.10	10.Z.10.10
A.B.C.228 (DNS-IFI-W-Backup)	10.X.Y.10	A.B.C.229	10.M.N.42	10.Z.10.10

Table 2 DNS Forwarder Configuration

The above relationship is shown in Figure 3.

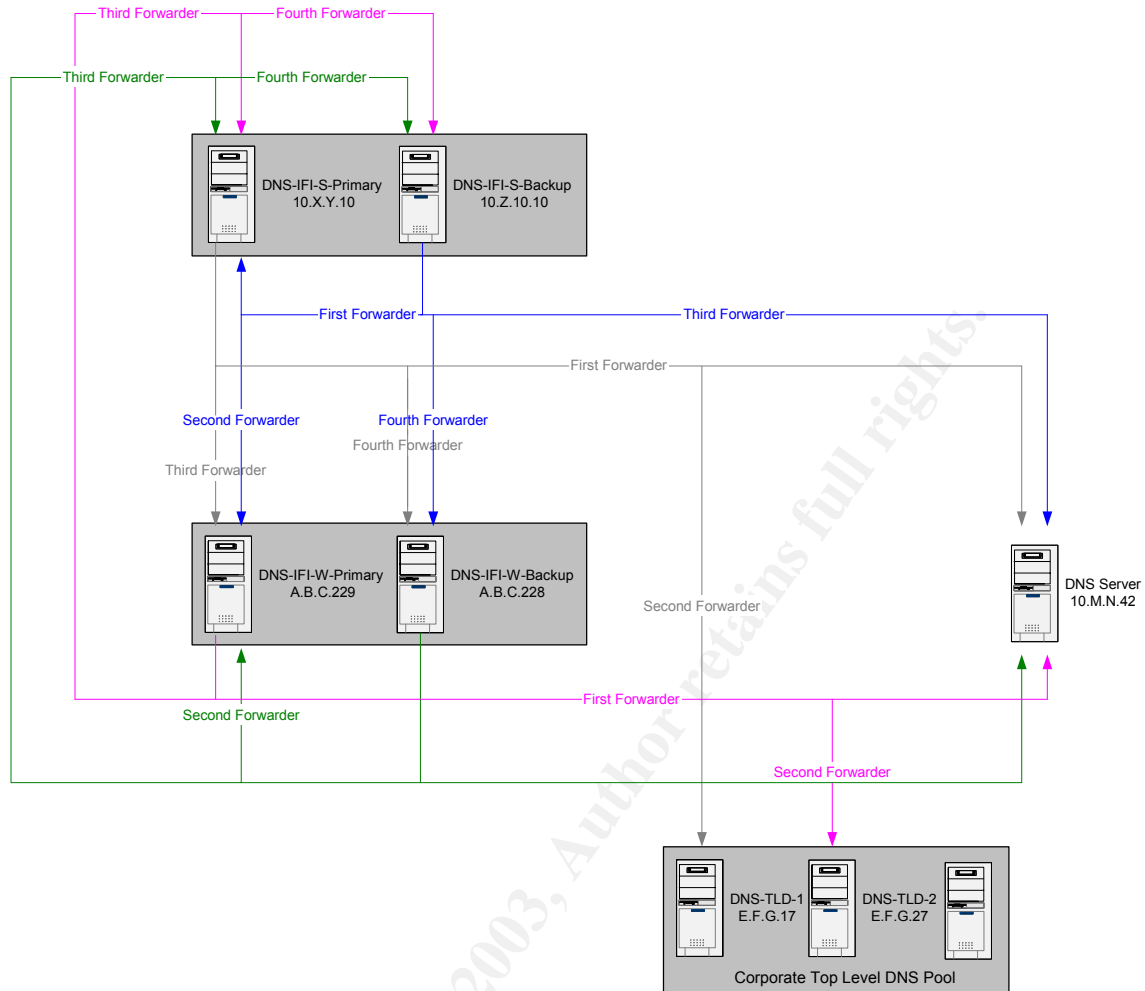


Figure 3 Forwarding for IFI DNS Servers

Even though IFI has its own Internet connectivity, the ACL policy, in both IFI-GWFW (a gateway firewall in IFI) and IFI-InternetFW (an Internet firewall in IFI), is specified to block all outbound DNS queries. (There is no DNS server in IFI DMZ. The IFI external DNS servers are hosted by ISP.)

The corporate top-level DNS pool is a set of corporate DNS root servers, which facilitate resolution of Internet domain names. Only DNS queries originated from the DNS pool are allowed to go through the gateway firewall GWFW and Internet firewall InternetFW.

None of the internal routers routes any traffic destined to the Internet unless the traffic comes from a proxy or DNS server. All border routers employ both ingress and egress filters. For example, the following restrictions are included in the ACL on IFI-BorderRouter (the border router in IFI):

```
access-list inbound deny tcp any any range 512 514 log
access-list inbound deny tcp any any eq 23 log
access-list inbound permit tcp any host <Mail Server> eq smtp
access-list inbound permit tcp any host <Web Server> eq http
```

```
access-list outbound deny ip 10.0.0.0 0.255.255.255 any log
access-list outbound deny ip 192.168.0.0 0.0.255.255 any log
access-list outbound permit tcp host <Mail Server> any eq smtp
access-list outbound permit tcp host <Web Server> any eq http
```

3 Virus in Action

W32/Yaha.g@MM, as its name (@MM) indicates, is a mass-mailing virus. It propagates by using the infected system's default SMTP server to send itself as an e-mail attachment. If such a SMTP server cannot be found, then the virus uses one of many SMTP server addresses which are hardcoded into the virus.

3.1 Protocol

The SMTP, which stands for Simple Mail Transfer Protocol, provides mechanisms for mail transmission directly from the sender host to the recipient host if the two hosts are connected to the same transport service; or via one or more relay SMTP servers if the source and destination hosts are connected to the different transport services [17].

The SMTP design is based on the following communication model. When a user makes a mail request, the SMTP-sender establishes a two-way transmission channel to an SMTP-receiver. The SMTP-receiver may be either the ultimate destination or an intermediate. Once the transmission channel is established, SMTP commands are generated by the SMTP-sender and sent to the SMTP-receiver. SMTP replies are sent from the SMTP-receiver to the SMTP-sender in response to the commands. This process is detailed as follows.

The SMTP-sender sends a *MAIL* command indicating the sender of the mail. The SMTP-receiver responds with an *OK* reply if it can accept mails. The SMTP-sender then sends a *RCPT* command identifying a recipient of the mail. If the SMTP-receiver can accept mails for that recipient, it responds with an *OK* reply; otherwise, it responds with a reply rejecting that recipient (but not the whole mail transaction). The SMTP-sender and SMTP-receiver may negotiate several recipients. Once the recipients have been negotiated the SMTP-sender sends the mail data, terminating with a special sequence. If the SMTP-receiver successfully processes the mail data it responds with an *OK* reply. This dialog is purposely lock-step, one at a time.

RFC 822 [18] defines strictly what message content format is to be passed between hosts. According to the specification, a message consists of header fields and, optionally, a body. The body is simply a sequence of lines containing ASCII characters. It is separated from the headers by a null line (i.e., a line with nothing preceding the CRLF).

In order to send a mail with the message body other than flat ASCII text, MIME (Multipurpose Internet Mail Extensions) is specified in RFC 1521 [19][20]. Using a *Content-Type* header field to specify the data type in the body of a message and the native representation (encoding) of such data, MIME-based mail can transmit image, audio, video, or binary data.

While MIME provides a widely used Internet standard for encoding multiple types of data as e-mail attachments, it poses security problems as well. An attacker can attach a malicious executable to an e-mail message and solicit recipients to execute it. Without the knowledge of what the executable could really do, the recipients usually open the attachment, resulting in their machines being infected.

3.2 Working Process

The virus file of W32/Yaha.g@MM is a Windows portable executable, which is about 30 kilobytes long. It is packed with UPX (Ultimate Packer for eXecutables) file compressor, and the UPX text strings in the body of the virus are replaced with dots. The end of the file is usually random data (garbage) with different length and content. The virus encrypts most of its text strings with a simple crypto algorithm.

W32/Yaha.g@MM spreads itself mainly via e-mail messages built by a complex composition routine. The virus may forge or spoof the source address and construct an e-mail message by randomly selecting components from a wide variety of pre-defined subject lines, message bodies, attachment names and extensions (refer to [7], [8], [9], [10], [12], [13], and [14] for details). Attachments are specified to have double extensions with .bat, .pif or .scr as the final one.

The composed e-mails have different types. Some include "Incorrect MIME Header" exploit. If so, unpatched systems automatically execute the attachment. Otherwise, the attachment must be run manually. In either case, the system running an e-mail client gets infected.

If activated (automatically or manually) on a clean system from a file with the extension of .scr, W32/Yaha.g@MM may display a message box or/and a video effect. The message displayed is chosen from the following list:

- Ur My Best Friend!!
- No Configuration is available Now
- Config
- madd
- U r so cute today #!#!
- True Love never ends
- I like U very much!!!
- U r My Best Friend

A screen saver may be displayed, and the image may shake back and forth.



Figure 4 Sample Screen Saver

The virus usually installs itself to *C:\Recycler* or *C:\Recycled* folder depending on the name of the recycle bin, or to *%Windows%*² folder if these recycle folders are not available. The generated file is hidden under the folder, with a name consisting of four random characters between 'c' and 'y'.

A text file is then stored to *%Windows%* folder, using the same random name as the virus file. This text file contains the following text.

```
<<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>>
```

iNDian sNakes pResents yAha.E

iNDian hACKers,Vxers c0me & w0Rk wITH uS & f*CK tHE GFORCE-pAK shites

bY

sNAkeeYes,c0Bra

```
<<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>> <<<>>>
```

By modifying the default value of the registry key

HKEY_CLASSES_ROOT\exefile\shell\open\command\default

to *""%virus_path%" %1 %"*,

W32/Yaha.g@MM configures itself to be executed whenever other executable files are started. Meanwhile, one of the virus threads continuously refreshes this key to prevent its modification by computer users or by disinfection utilities. The same thread tries to kill Windows Task Manager process.

² *%Windows%* is a variable, which is *C:\Windows* or *C:\Winnt* by default.

The virus saves a Base 64 encoded copy of itself into a file named *kitkat* under a temporary folder *%Windows%\Temp*, which is used in mass-mailing. The file is different from the original virus sample as random data is added to its end (from *munch* file that the virus generates and then deletes). The length of random data differs every time. This technique was supposed to fool some anti-virus scanners that use CRC to detect standalone malware.

Then W32/Yaha.g@MM begins to prepare for its propagation. It gathers e-mail addresses from Windows Address Book, MSN Messenger data, Yahoo! Messenger profiles, ICQ data files, HTML, HTM, HTA, DOC and TXT files in cache folders, desktop and personal folders. The discovered e-mail addresses are stored into a DLL file in *%Windows%* folder, with a name of eight characters repeating the four characters used for the above text file.

The virus sends out itself as an e-mail attachment to all addresses found on an infected system, using a default SMTP server retrieved from the registry

HKEY_CURRENT_USER\Software\Microsoft\Internet Account Manager\Accounts

If the above SMTP server is not found, the virus uses one of the pre-defined SMTP servers.

The virus also has network spread capabilities. One of its threads constantly looks for open shares and directories with the following names.

- WIN
- WIN95
- WIN98
- WINDOWS
- WINNT
- WINME
- WINXP

Once such a share/directory is found, the virus tries to copy itself to it under the name *mstaskmon.exe* and adds an execution string into *win.ini* file on the remote machine to load the virus at system startup. This functionality only works under Windows 9x systems, as *win.ini* file is not used by NT-based systems. However the virus can be activated on an NT-based system if *mstaskmon.exe* file is executed on the machine. The virus runs itself as a service process, thus its task is not visible in Task Manager under Windows 9x systems.

W32/Yaha.g@MM does not start if launched from Internet Explorer cache folder (the folder containing *CONTENT.IE5* string).

The virus has different process killing routines for different types of operating systems. It searches and terminates security-related processes with the designated strings in their names [10][13][8] (refer to Appendix 1 for details).

From time to time, the virus tries to connect to <http://www.pak.gov.pk>, which is probably a faked web site. Otherwise, this attempt can cause a DoS (Denial of Service) attack against the web server if the virus is widespread.

Moreover, W32/Yaha.g@MM regularly searches the following registry keys which might cause performance degradation on an infected system.

HKCU\Software\Microsoft\MessengerService\ListCache\MSN Messenger Service

HKCU\Software\Microsoft\MessengerService\ListCache\NET Messenger Service

HKCU\Software\Yahoo\Pager\profiles\%s\IMVironments\Recent

HKCU\Software\Yahoo\Pager\profiles

HKCU\Software\Mirabilis\ICQ\DefaultPrefs

To avoid starting multiple instances of the virus, a mutex (MUTual EXclusion object) named *pkf*ck* is created.

3.3 Signature

W32/Yaha.g@MM virus leaves several traces on an infected system because of its extensive functionality and alternation to the system. The “signatures” that the virus creates are listed as follows.

- Attempt to connect to the web site, <http://www.pak.gov.pk>, several times a minute;
- Modification of the default value of the registry key *HKEY_CLASSES_ROOT\exefile\shell\open\command\default*, which is set to *""%virus_path%" %1 %*"*;
- Modification of *win.ini* file, with an execution string added to load *mstaskmon.exe* at system startup (only on Windows 9x systems);
- Newly generated files:

- An executable (.exe) file with a random four-character name under *C:\Recycler* or *C:\Recycled* or *%Windows%* folder;
- A text (.txt) file under *%Windows%* folder, with the same name as the above virus file and the following content;

<<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>>

iNDian sNakes pResents yAha.E

iNDian hACKers,Vxers c0me & w0Rk wITH uS & f*CK tHE GFORCE-pAK shites

bY

sNAkeeYes,c0Bra

<<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>> <<<<>>

- A Base 64 encoded file named *kitkat* under *%Windows%\Temp* folder;
- A dynamic link library (.dll) file under *%Windows%* folder, with a name of eight characters repeating the four characters used for the text file mentioned above;
- An executable (.exe) file named *mstaskmon.exe* under open shares or directories;

- Termination of the security-related processes as listed in Appendix 1;
- Termination of Windows Task Manager process;
- A message box or/and a video effect, with the message chosen from the following list:
 - Ur My Best Friend!!
 - No Configuration is available Now
 - Config
 - madd
 - U r so cute today #!#!
 - True Love never ends
 - I like U very much!!!
 - U r My Best Friend
- A screen saver as shown in Figure 4;
- Performance degradation.

Due to the characteristics of the e-mail which W32/Yaha.g@MM uses to propagate itself, an anti-virus software installed on the Internet e-mail gateway can filter the virus by replacing the attachment file.

3.4 Protection

The following lists some basic security “best practices” which help protect against (not limited to) W32/Yaha.g@MM virus:

- Always keep the system patch levels up-to-date. A patch for “Incorrect MIME Header Can Cause IE to Execute E-mail Attachment” vulnerability has been around for over two years already [16].
- Assess system vulnerabilities, including system currency, on a regular basis.
- Install anti-virus software on both e-mail servers and personal computers, and maintain virus signature files current. Most anti-virus software companies, such as Network Associates, Symantec, F-secure, Trend Micro, Sophos, now have signatures for the W32/Yaha.g@MM virus family.
- Configure e-mail servers to block or remove e-mails that contain file attachments that are commonly used to spread viruses, such as .bat, .pif, .scr, .exe, and .vbs files.
- Never open attachments unless they are expected. Also, do not execute software that is downloaded from the Internet unless it has been scanned for viruses. Be aware that simple access to a compromised web site can cause infection if certain browser vulnerabilities are not patched.
- Avoid using web mails; otherwise an e-mail with malicious attachment can bypass network perimeters. Alternatively, conduct content scanning for all HTTP downloads, and filter potentially dangerous contents, e.g., executable files and batch files.

- If signature-based intrusion detection systems are in use, enable the signature for detecting double extensions in e-mail attachments.
- Whenever possible, block or limit the ability to download executable files from the Internet.
- Isolate infected computers quickly to prevent further compromising the organization. Perform a forensic analysis and restore the computers using trusted media.

3.5 Infection in IFI

According to the security policy in IFI, web mail is not recommended. But there is no filter set up on either the Internet firewall or the gateway firewall to block web mail access.

W32/Yaha.g@MM arrived at IFI network through web mail. An innocent user, sitting on the subnet 10.0.0.0/8 in IFI, connected to Yahoo! Mail, and found a new e-mail in his inbox, titled "True Love never ends". Curiously, he opened the e-mail, but left the attachment intact because he knew he should not open an e-mail attachment when it is from a stranger.

Unfortunately, the user was using Microsoft Internet Explorer version 5.5 for Windows 95 with only Service Pack 1 and patch Q279328 installed, which could be affected by "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment" vulnerability [16].



Figure 5 Internet Explore on the User Machine

As configured to enable File Downloads by default, the user's Internet Explorer executed the e-mail attachment automatically without his awareness. The user machine, a Dell OptiPlex GX 200 desktop, was infected by the virus loaded in the attachment.

Then W32/Yaha.g@MM virus generated several new files, saved them on the machine, and modified the registry key. This caused the virus was activated whenever the user started an application – a window popped up, informing that an illegal operation was performed.

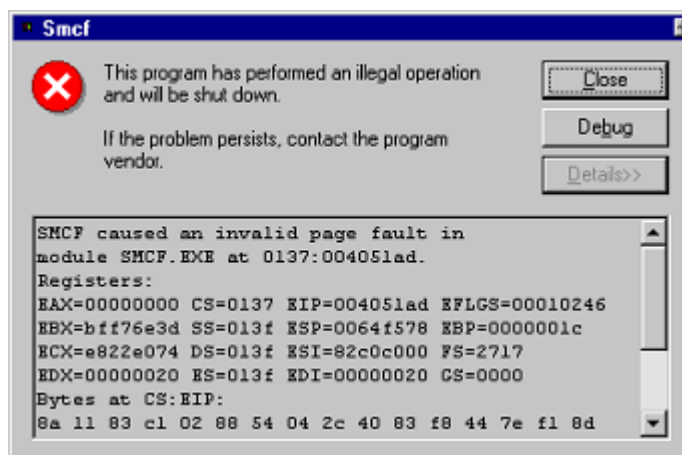


Figure 6 Popped-Up Message Box

The virus also tried constantly to probe the web site <http://www.pak.gov.pk>. As this was probably a faked web site, name resolution generated a lot of traffic between DNS servers on both IFI and headquarter networks, resulting in the network performance degraded and DNS servers heavily loaded (refer to Figure 7).

No.	Time	Source	Destination	Protocol	Info
25971	253.781782	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26207	255.838459	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26209	255.841801	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26285	256.467668	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26286	256.467932	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26289	256.471943	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26290	256.472123	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26489	258.632079	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK
26490	258.632267	. . .229	. . .17	DNS	Standard query A www.PAK.GOV.PK

No.	Time	Source	Destination	Protocol	Info
25853	252.907189	. . .17	. . .229	DNS	Standard query response, No such name
25866	253.017780	. . .17	. . .229	DNS	Standard query response, No such name
25878	253.152033	. . .17	. . .229	DNS	Standard query response, No such name
25891	253.247005	. . .17	. . .229	DNS	Standard query response, No such name
25915	253.386087	. . .17	. . .229	DNS	Standard query response, No such name
26017	254.367096	. . .17	. . .229	DNS	Standard query response, No such name
26035	254.465101	. . .17	. . .229	DNS	Standard query response, No such name
26042	254.539116	. . .17	. . .229	DNS	Standard query response, No such name
26062	254.647557	. . .17	. . .229	DNS	Standard query response, No such name

Figure 7 Packets Captured on the IFI Network

The above infection process is illustrated in Figure 8.

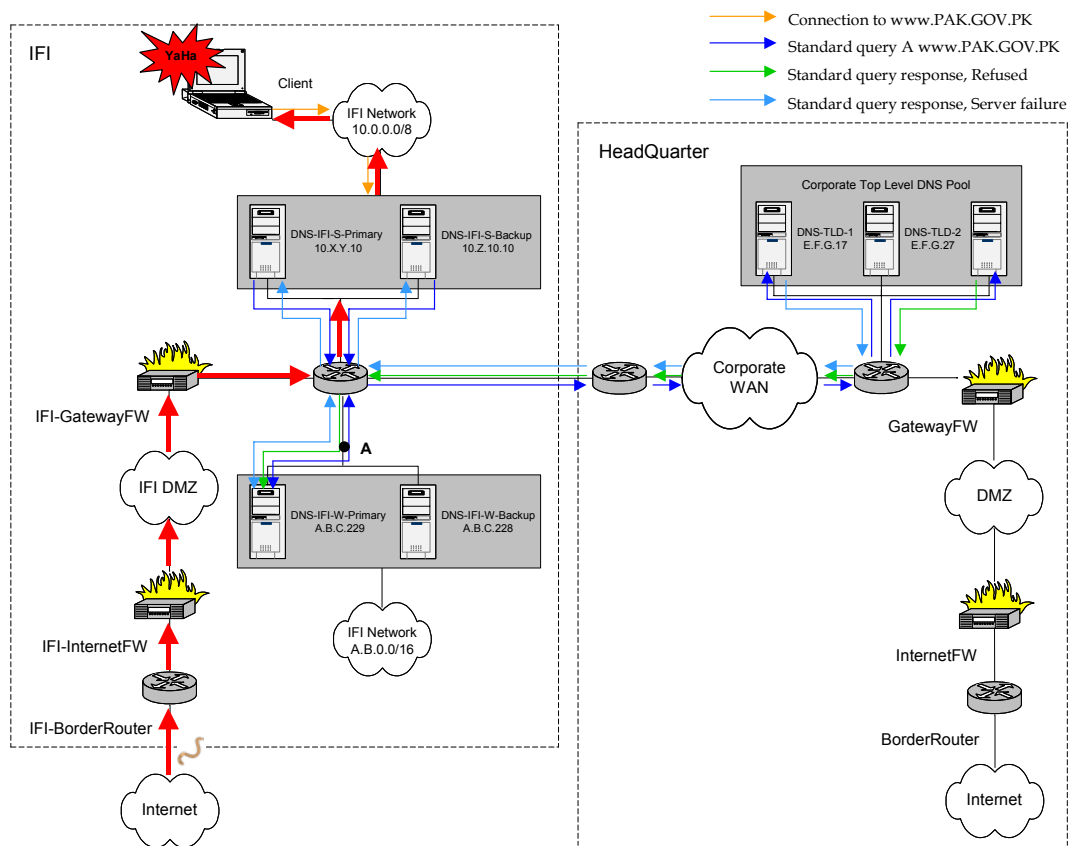


Figure 8 Infection Process in IFI

4 Incident Handling

Incident handling is “an action plan for dealing with intrusions, cyber-theft, denial of service, fire, floods, and other security-related events” [21]. It is important because an incident is going to occur sooner or later – It is not a matter of IF but WHEN an incident is going to happen [21].

There are six steps in incident handling – preparation, identification, containment, eradication, recovery and lessons learned [21][22].

4.1 Preparation

The best way to act on an incident and minimize the chance of a mistake is to have proper procedures in place – planning is everything and preparation plays a key role.

IFI has several security countermeasures existing.

4.1.1 Policy

Security policy has been established in the financial group. It provides high-level direction related to information security as an adjunct to and in support of the enterprise policy.

“Where appropriate and prudent, based on analysis of security exposures and business risks, and as required by Operating Directives (see 8.2), measures must be implemented to prevent, detect, and respond to:

- *unauthorized disclosure of, or attempts to access or to learn the existence or attributes of, financial group information resources;*
- *integrity (authenticity, accuracy, or completeness) or availability failures of information, systems, communications or processes;*
- *attempts to forge or masquerade as authentic information resources or to submit duplicate transactions or replay communications; and*
- *attempts to corrupt, disrupt or deny access to information resources.”*

“Appropriate contingency plans for critical applications and vital processes must be developed and maintained, including maintenance of acceptable security, in anticipation of major disruption of service.”

4.1.2 Process and Procedure

Incident response, reporting and escalation process has been well documented. The developed document on Incident Response and Escalation Procedure details the procedures of incident alert, discovery, investigation (including risk assessment, and solution determination), engagement (including containment, eradication, and recovery), and unwinding (including goodness/weakness identification, procedure update, and lessons communication). It contains some necessary forms – incident response contact list, incident levels and response, incident identification form, incident survey form, incident containment form and incident handling form.

After several practices and tests, the established incident handling process and procedure is getting more and more mature, which exercised successfully during the SQL Slammer Worm propagation at the end of January 2003.

4.1.3 People

Information Security Department was founded in the financial group five years ago. It is responsible for security policy creation and maintenance, internal project consultation, incident handling, intrusion detection, anti-virus management, vulnerability assessment, and key and certificate management.

The incident handling core team consists of a senior manager and security analysts from Information Security. If required, security officers from the Line of Business involved can join the team as local incident handlers. All of the team members are well-trained security professionals.

4.1.4 Technology

The network infrastructure in the whole financial group is secured by several means. It is designed under the concept of “defense in depth”. The networks are categorized into external zone, web zone, application zone, database zone, management zone and internal zone. Firewalls are deployed between each zone, and on the internal critical systems. In addition, network intrusion detection systems are deployed at key access ports to different zones.

Vulnerability assessments are conducted on a quarterly basis to address potential areas of security exposure which can be exploited from the internal networks or the external networks.

The Internet e-mail gateways in IFI have installed McAfee *GroupShield Domino5*, which combines virus detection and cleaning functionality with functions designed to improve attachment scanning and real-time scanning for viruses and malicious codes.

Every Windows server, workstation, desktop and laptop, which is connected to the corporate networks, is required to install McAfee anti-virus software. The latest Virus Definition (DAT) file is put on an internal FTP site after being tested. The anti-virus software is pre-configured to check the FTP site daily to ensure that the virus protection is up-to-date with cures for the newest virus threats.

At present, a pilot project, ePO, is in the works. ePO can enable corporate network administrators to keep the anti-virus software up-to-date once new DAT files are released and tested, and control the configuration of all desktops and servers in the financial group so that the virus security policy can be consistently implemented and enforced.

Additionally, Information Security set up a hotline and a mailbox to facilitate incident reporting.

4.2 Identification

The initial alarm of the incident rose when the CPU usage of the corporate top level DNS servers was above 15% - On each of these DNS servers (Solaris machines), there is a health-checking program running. If the CPU usage is above 15%, the DNS administrators get paged.

After running Sun Solaris *snoop* on the DNS servers, the system administrator found that a lot of traffic came back and forth between IFI and headquarter, which was the consequence of attempts to connect to the web site www.pak.gov.pk. This suspicious activity was reported at once to Information Security through hotline. The incident handling core team was summoned. In a short time, a security officer from IFI was informed and asked to join the incident handling team. As required, an IFI corporate security joined the team as well.

As the traffic, which brought heavy load on the corporate top-level DNS servers, was the standard DNS queries originating from the primary DNS server on IFI subnet A.B.0.0/16 (DNS-IFI-W-Primary in Figure 8, Windows machine), the security officer from IFI employed Network Associates *SnifferPro* to monitor the traffic between the router and the DNS server (point A in Figure 8). The output (refer to Figure 7) showed that the root cause of the troublesome traffic resulted from a huge number of name resolution requests generated from the IFI DNS servers connected to subnet 10.0.0.0/16 (DNS-IFI-S-Primary and DNS-IFI-S-Backup in Figure 8, Solaris machines) – Someone on the subnet had been trying continuously to connect to the web site www.pak.gov.pk.

Further investigation using Sun Solaris *snoop* on the DNS servers DNS-IFI-S-Primary and DNS-IFI-S-Backup discovered the IP address of the questionable machine.

Consulting the IP Address Management helped locate the machine. It is a Dell OptiPlex GX 200 desktop, running Microsoft Windows 95 operating system (refer to Figure 9).

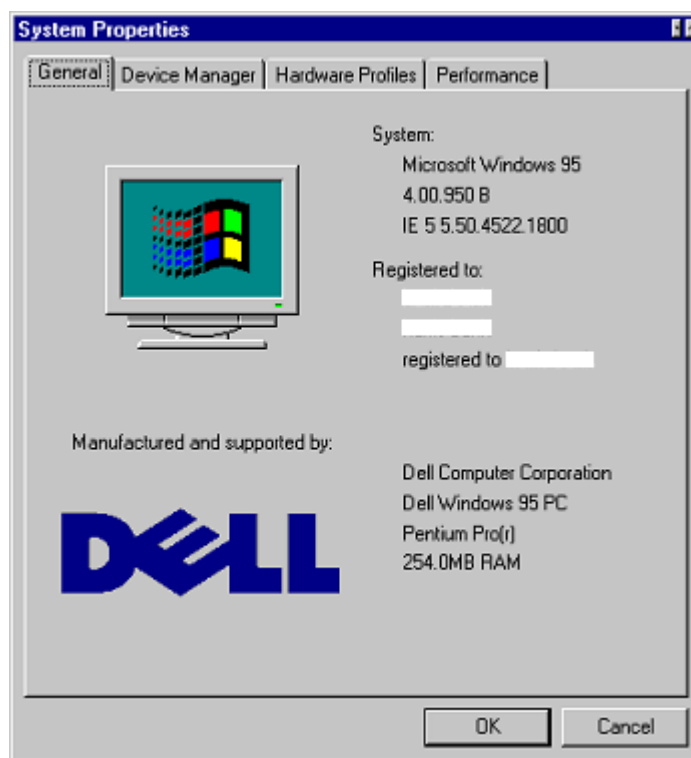


Figure 9 System Properties on the User Machine

One incident handler had been working in threat tracking in the past several months. She recalled a kind of “Yaha” virus causes the same symptom – constantly probing the web site <http://www.pak.gov.pk>. The computer was suspected to be infected by virus.

In order to maintain a provable chain of custody, any evidence was decided not to be destroyed and a clean binary backup of the system would be made before incident handlers started making any modifications.

The identification process took nearly three hours after the first alarm raised.

4.3 Containment

Immediately, the incident handling team contacted the user of the suspicious machine, and instructed him to shut down the system and unplug it from the network. The IFI security officer and the corporate security were sent to the site.

The first thing the incident handler did was to keep the system pristine. He made an image of the system using Symantec *Ghost*.

A virus scan needed to be gone through to make sure whether the machine in question was infected. Based on the information from anti-virus software vendors, “Yaha” virus might contaminate the anti-virus software installed on the infected machine. Therefore, the incident handler rebooted the machine using a bootable

floppy disk made from McAfee Emergency .DAT files. These files remove viruses from locations which can be used to infect critical system files or anti-virus software. The following is the output of the McAfee scan.

McAfee VirusScan for DOS/PM v4.14.0
Copyright (c) 1992-2001 Networks Associates Technology Inc. All rights reserved.
(408) 988-3832 LICENSED COPY - Jan 17 2001

Scan engine v4.1.40 for DOS/PM.
Virus data file v4247 created Feb 13 2003
Scanning for 4941 viruses, trojans and variants.

03/06/2003 12:49:01

Options:
/ALL /ADL /REPORT C:\MCAFEEOUTPUT.TXT

Scanning C: [80261027]
Scanning C:*.*
C:\RECYCLED\SMCF.EXE
Found the W32/Yaha.g@MM virus !!!

Summary report on C:*.*

File(s)
Total files: 10638
Clean: 10636
Possibly Infected: 1
Non-critical Error(s): 1
Master Boot Record(s): 1
Possibly Infected: 0
Boot Sector(s): 1
Possibly Infected: 0
Scanning D: [MS-RAMDRIVE]
Scanning D:*.*

Summary report on D:*.*

File(s)
Total files: 16
Clean: 16
Possibly Infected: 0
Master Boot Record(s): 1
Possibly Infected: 0
Boot Sector(s): 1
Possibly Infected: 0

Time: 00:29.23

By far, the incident handlers knew the "version" of "Yaha" virus, and found the starting point to do further investigations. According to the signatures of W32/Yaha.g@MM virus, they traced the evidences one by one.

Evidence 1

The virus file *smcf.exe* was hidden under *C:\Recycled*. A simple dir command could not list the file.

```
C:\>dir /ah c:\recycled
```

```
Volume in drive C is 80261027
Volume Serial Number is 07D0-031B
Directory of C:\RECYCLED
```

```
SMCF  EXE      29,204  01-27-03  3:07p
      1 file(s) 29,204 bytes
      0 dir(s)  8,828.59 MB free
```

After this detection, the machine was restarted into Windows safe mode.

Evidence 2

Each time when an executable was started, a message box popped up, showing *smcf.exe* was executed (refer to Figure 6).

Evidence 3

The registry key was modified to make *smcf.exe* run once another executable file was started.

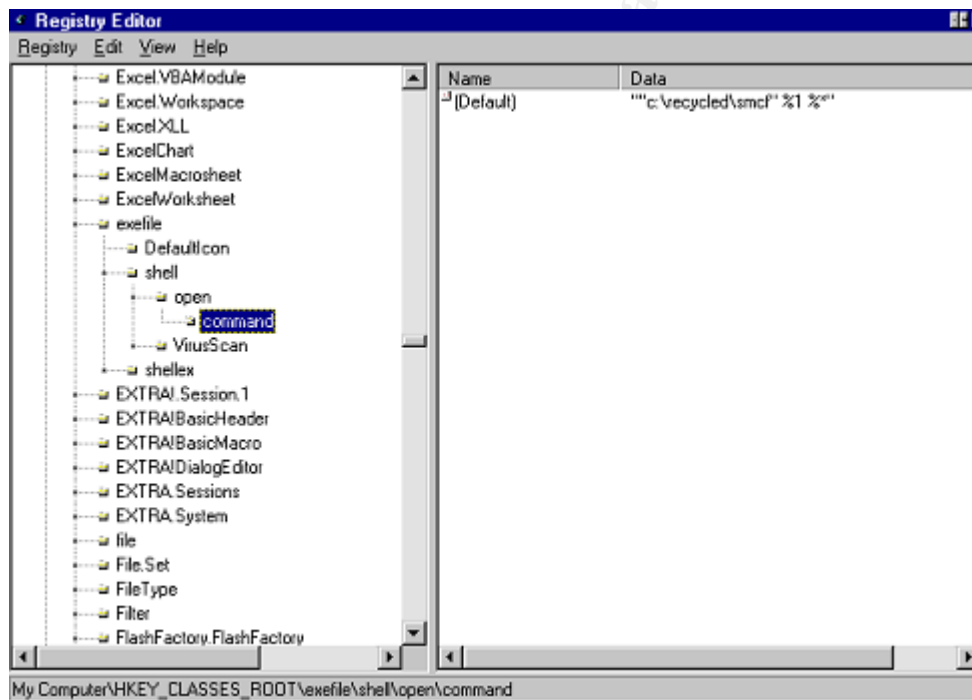


Figure 10 Modified Registry Key

Evidence 4

Two virus-generated files (*smcf.txt* and *smcfsmcf.dll*) were saved under *C:\Windows*³.

³ The screen shot was taken when the backup of the infected system was analyzed. Hence, the file timestamps do not reflect the authenticity.

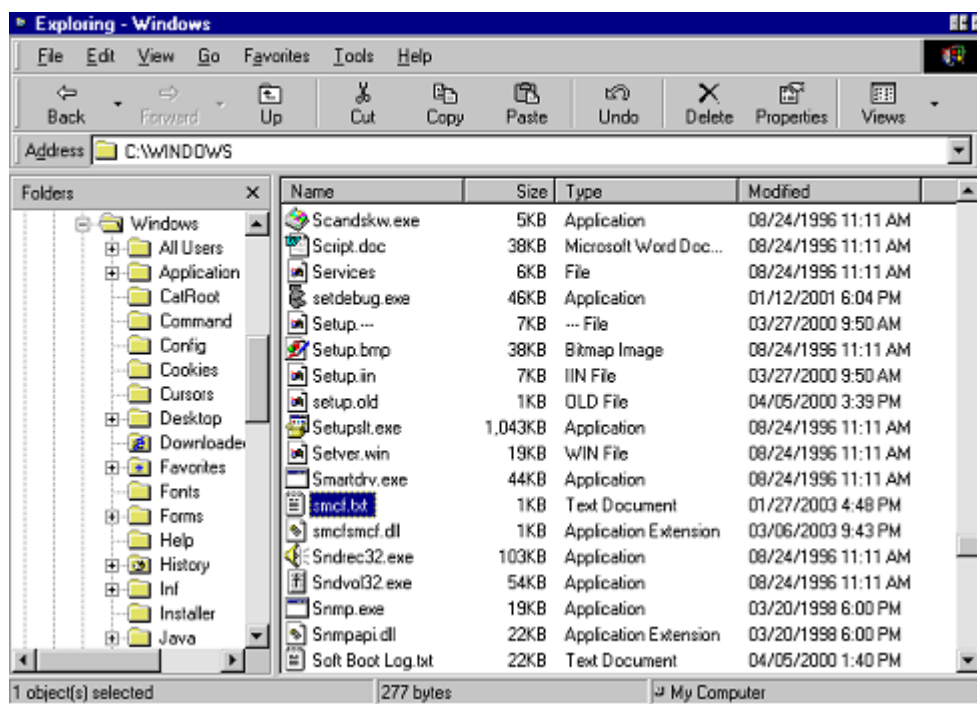


Figure 11 Virus Generated Files

Evidence 5

The text file *smcf.txt* has the following content.

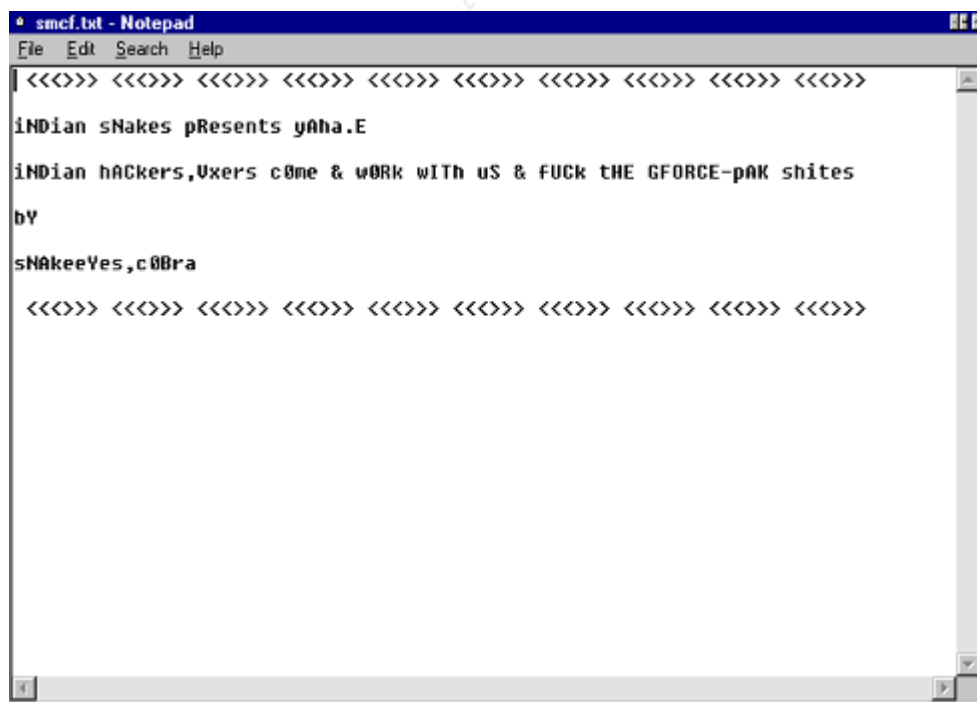


Figure 12 File Content

Evidence 6

The Base 64 encoded virus file *kitkat* was found under *C:\Windows\TEMP*⁴.

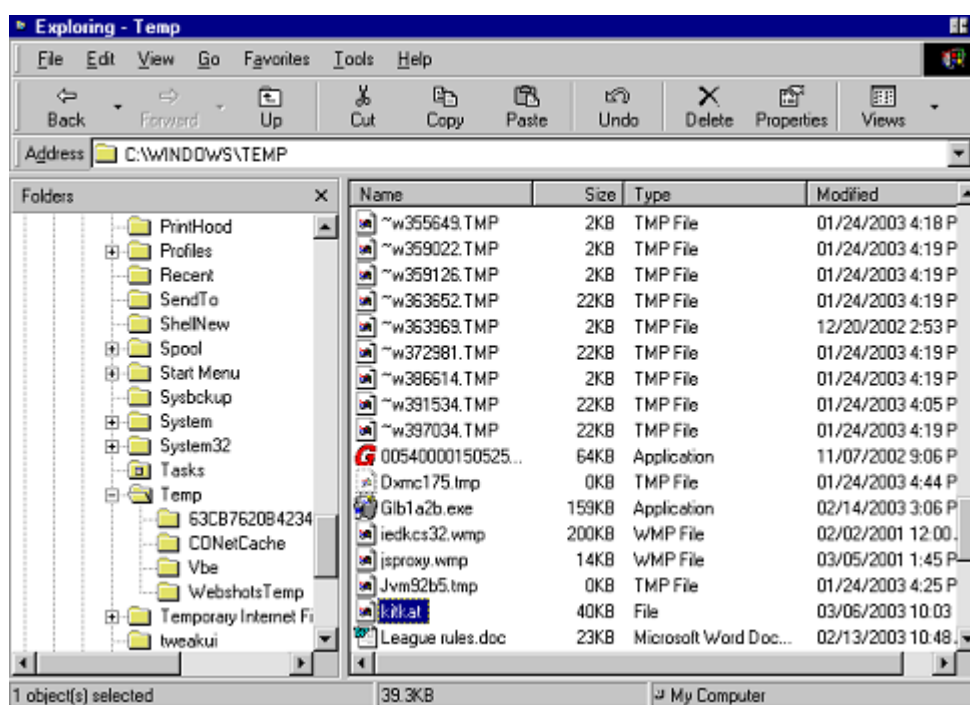


Figure 13 Encoded Virus File

Evidence 7

The system attempted to connect to the web site <http://www.pak.gov.pk> from time to time.

At this point, it was clear that the machine was infected by W32/Yaha.g@MM. This made straightforward the processes of eradication and system recovery.

4.4 Eradication

To eliminate the opportunity for re-infection via the same channel after reloading the system, the incident handlers had to discover when and how the machine was infected by W32/Yaha.g@MM virus.

By checking the timestamps of the files left by the virus, it was not difficult to find out that the infection time was around 4:00 PM on January 27, 2003.

```
C:\>dir /ah c:\recycled
```

```
Volume in drive C is 80261027
Volume Serial Number is 07D0-031B
Directory of C:\RECYCLED
```

```
SMCF   EXE       29,204  01-27-03  3:07p
      1 file(s)    29,204 bytes
```

⁴ The screen shot was taken when the backup of the infected system was analyzed. Hence, the file timestamps do not reflect the authenticity.

0 dir(s) 8,828.59 MB free

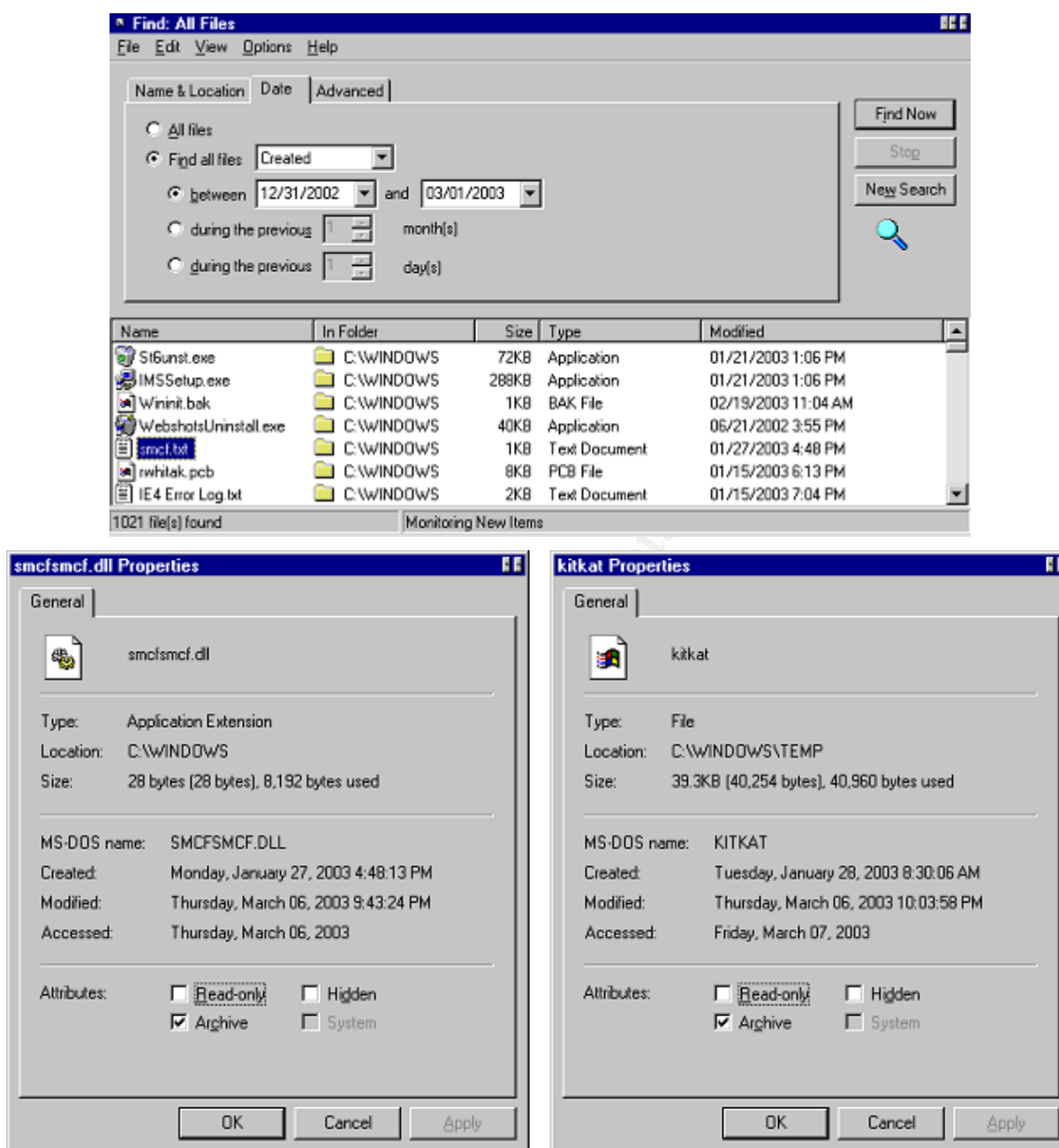


Figure 14 File Timestamps

According to the information on anti-virus software vendors' web sites, W32/Yaha.g@MM virus has two methods to propagate itself – through e-mail attachment or network share.

Basically, the machine was not infected through network shares since no evidence associated with this existed – no *mstaskmon.exe* file was found on the infected system.

Long before the incident happened, McAfee *GroupShield* installed on the Internet e-mail gateways in IFI had been updated to detect W32/Yaha.g@MM virus and clean the virus' payload (attachment). The *GroupShield* log demonstrated the software did detect the virus and filter the payload (refer to Figure 15⁵). Thus, the virus could not come in via e-mails through the Internet e-mail gateway.

Anti-Virus Statistics										
Viruses Population on the Lotus Notes Gateways										
Totals for Gateways										
Week	1	2	3	4	5	6	7	8	9	
Virus Name	1/2/2003	1/9/2003	1/16/2003	1/23/2003	1/30/2003	2/6/2003	2/13/2003	2/20/2003	2/27/2003	YTD Total
Exploit-MIME.gen.b	58	186	268	219	76	72	96	110	98	1183
W32/Klez.e@MM	1	3								4
W32/Klez.h@MM	60	205	285	249	120	97	137	150	144	1447
Exploit-MIME.gen.exe		1		2				1		4
JS/Kak@M		2			1			2	1	6
VBS/Redlof.dam			1							1
VBS/Redlof@M			1							1
W32/Bugbear@MM		1	3	1	1		1	2	1	10
W32/Magistr.b@MM		1	7	3	4	3	2	4	2	26
W32/SuCam@MM		1		1						2
W32/Sobig@MM			49							49
W32/Yaha.g@MM		1	13			1		1		16
W97M/ColdApe.gen			1	1	1				2	5
W97M/Marker.gen		1						1		2
W32/Magistr.a@MM			1							1
W97/Ethan.d					2					2
JS/Fortnight@M						1				1
W97M/Story.gen						1		1	1	3
W97/Class							1			1
W97M/Marker.o								1		1
W97M/Panther.gen									1	1
Weekly Totals w/o Banned Content	119	402	629	476	205	175	237	272	251	2766

Figure 15 Groupshield Anti-Virus Report

It was impossible for the virus to come via internal e-mail either, because there were no other machines ever discovered to be infected.

Then the possible way of the virus infection was through web mails. This was confirmed by the user. He did use Yahoo! Mail some time ago and had no updated anti-virus software on the desktop.

Further investigation revealed Microsoft Internet Explorer on the infected system was vulnerable to "Incorrect MIME Header" (refer to CVE-2001-0154), which executes e-mail attachments automatically on an unpatched system (refer to Figure 5).

Surprisingly, the incident handler found that McAfee anti-virus software on the infected machine was last updated at 10:00 AM October 24, 2002, which was presented by the modification time of *Update.txt* file. Moreover, it seemed the anti-

⁵ The screen shot was taken during forensics analysis.

virus software had not been working since October 24, 2002 –The last time *Vshlog.txt* was modified was 12:02 AM October 24, 2002 (refer to Figure 16).

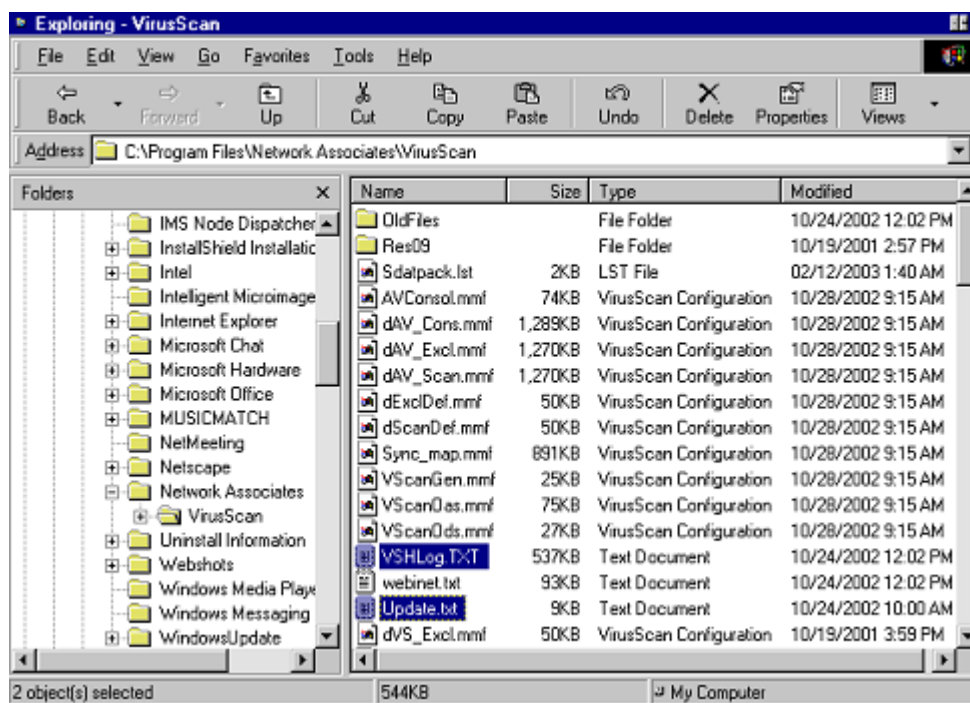


Figure 16 Anti-Virus Software Information on the User Machine

Having determined the cause and symptoms of the incident, the incident handler started out to clean the infected system. He took the following steps.

- Search McAfee Security web site for instructions.
- Download McAfee AVERT Stinger executable, and save it to a floppy disk.
- Check the authenticity of the digital signature.
- Reboot the infected machine into Windows safe mode.
- Run Stinger executable from the floppy disk.

By default, Stinger executable repairs all infected files found (refer to Figure 17⁶).

When the incident handler double-checked the traces ever left by the virus, he found

- the executable file *smcf.exe* was removed from *C:\Recycler*;
- the registry key *HKEY_CLASSES_ROOT\exefile\shell\open\command\default* was set back to *""%1" %**;
- two virus-generated files *smcf.txt* and *smcfsmcf.dll* still existed under *C:\Windows*; and
- the Base 64 encoded virus file *kitkat* was found under *C:\Windows\TEMP* as well.

⁶ The screen shot was taken when the backup of the infected system was analyzed. Hence, the timestamps do not reflect the authenticity.

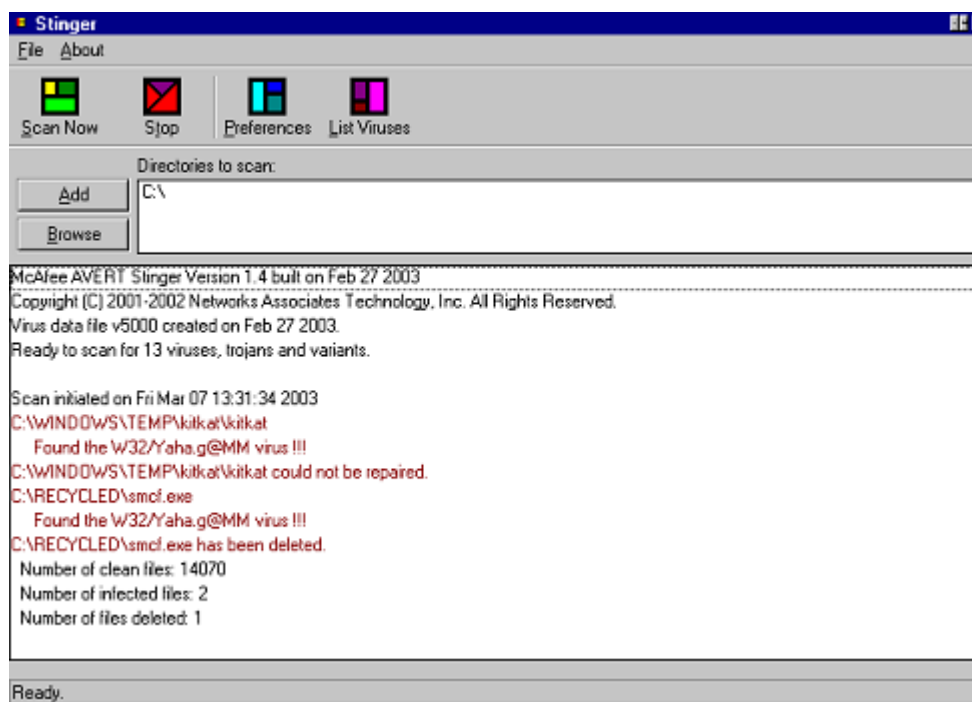


Figure 17 Output from Stinger Executable

The incident handler then deleted the three remaining virus-related files manually so as to fix problems completely. Afterwards, he restarted the machine into Windows normal mode.

4.5 Recovery

Before the system was brought back into operation, the incident handlers must make sure that all the problems or the vulnerabilities have been fixed, and defenses have been improved.

Based on the findings in the eradication process, the incident handler first activated the McAfee anti-virus software on the machine, updated its engine and the Virus Definition file to the latest version, and configured the software to be updated automatically on a daily basis.

Then the machine went through a full system scan to confirm that no remnants of the virus existed, and the system was not infected by some other viruses either.

By following the instructions on Microsoft Security Bulletin (MS01-020) [16], the incident handler patched the user's Internet Explorer to eliminate the "Incorrect MIME Header" vulnerability.

Advice was provided to the user of the machine to avoid using web mails whenever possible.

After all, the machine was ready to go back online. The incident handler left the machine owner to decide when to reconnect the machine to the network.

4.6 Lessons Learned

When looking back at the incident of the W32/Yaha.g@MM virus infection, the incident handlers found themselves having benefited quite a lot from the existing incident response and escalation procedure – each step was clear and there was no time wasted. People from different departments cooperated very well. For example, when locating the infected machine, the network management department responded promptly. This was extremely important in time-based security⁷. It paved the way for incident containment and infection eradication with a minimum expense.

The incident also revealed the deficiencies in IFI networking and security countermeasures. The combination of these was the root cause of the W32/Yaha.g@MM virus infection in IFI.

- DNS forwarders in IFI were configured cyclic. This problem might be solved by introducing more DNS zones.
- The anti-virus software on the infected machine had not been functioning for a long time without being noticed. This violated the corporate security policy.
- The user system was not patched up-to-date.
- Even though web mail was not recommended in IFI, there was no action taken to enforce the policy.

The identified cause of the incident should be used as a basis for new awareness initiatives among technical support, anti-virus management and end users; further, it can motivate the search for closing the gaps in anti-virus process and updating incident handling procedures. A good example will be to do content scanning for all HTTP downloads.

One problem the incident handlers met was that Stinger executable could not eradicate all virus-related files, which was not mentioned on McAfee Security web site. If the incident handler were not experienced, the eradication process would not have been finished completely.

There is a pending issue. The incident was reported on February 19, 2003. Why the long time gap exists between the virus files were installed (on January 27, 2003) and the virus broke out is unclear so far.

To prevent similar incidents to occur in the future, the incident handlers made the following recommendations.

- Keep the system patch levels up-to-date.
- Assess system vulnerabilities, including system currency, on a regular basis.
- Install anti-virus software on both e-mail servers and personal computers, and maintain virus signature files current.

⁷ Time-based security is based on the notion that the security of a site (the protection time) is based on the time it takes to detect an event and the time it takes to react to an event [22].

- Configure e-mail servers to block or remove e-mails that contain file attachments that are commonly used to spread viruses, such as .bat, .pif, .scr, .exe, and .vbs files.
- Enable the signature on intrusion detection systems for detecting double extensions in e-mail attachments.
- Educate people not to open attachments unless they are expected, and not to execute software that is downloaded from the Internet unless it has been scanned for viruses.
- Block access to some vulnerable web sites.
- Avoid using web mails. There might be a demand for establishing a policy to restrict web mails. Alternatively, conduct content scanning for all HTTP downloads, and filter potentially dangerous contents, e.g., executable files and batch files.
- Whenever possible, block or limit the ability to download executable files from the Internet.

5 Concluding Remarks

Computer viruses have been around for several decades and will not seem to disappear in the near future. Sooner or later most organizations will regularly encounter virus outbreaks, which incurs harm and loss. To minimize the cost caused by virus infection, defense is very important, and so is incident handling when virus is in action.

An incident handling process consists of six steps – preparation, identification, containment, eradication, recovery, and lessons learned. These steps serve the handlers as a compass or a roadmap, a way to keep in mind what they are trying to do and the things they need to do next. Through practices and exercises, an incident handling process can become more and more sound. Incident handling should not end up with certain step but go forever – the end of one incident handling process is the beginning of the next one – by improving the procedure itself and security defenses.

Acknowledgement

I would like to take the opportunity to thank Dongmei Huang for her contribution to information gathering. Special thanks go to Matunda Nyanchama and Yuxiao Zhao for their great comments on this paper.

References

- [1] Harker, Jonathan. "Protecting Your Business from the Virus Plague." February 17, 2003. URL: <http://www.accountancyage.com/Features/1132558> (May 28, 2003).
- [2] Davis, Mark. "How to Avoid Computer Viruses." Internet Tips & Tricks. May 28, 2003. URL: <http://home.commspeed.net/tips5.html> (May 28, 2003).

- [3] Norman ASA. "The Norman Book on Computer Viruses." Norman Books. October 2001. URL: <http://download.norman.no/manuals/eng/BOOKON.PDF> (May 25, 2003).
- [4] Cohen, Fred. "Computer Viruses - Theory and Experiments." Computer Virus Defense Techniques. 1984. URL: <http://all.net/books/virus/index.html> (May 25, 2003).
- [5] McAfee Security. "Virus Glossary of Terms." Virus Information. URL: http://www.mcafee.com/anti-virus/virus_glossary.asp (May 25, 2003).
- [6] Trend Micro. "Virus Primer." General Virus Information. URL: <http://www.trendmicro.com/en/security/general/virus/overview.htm> (May 25, 2003).
- [7] McAfee Security. "W32/Yaha.g@MM." Virus Information Library. URL: http://vil.nai.com/vil/content/v_99528.htm (May 25, 2003).
- [8] F-Secure Corporation. "Yaha.E." F-Secure Virus Descriptions. URL: http://www.europe.f-secure.com/v-descs/yaha_e.shtml (May 25, 2003).
- [9] Trend Micro, Inc. "Worm_Yaha.G." Virus Encyclopedia. URL: http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_YAHA.G (June 2, 2003).
- [10] Symantec Corporation. "W32.Yaha.F@mm." Virus Encyclopedia. URL: <http://securityresponse1.symantec.com/sarc/sarc.nsf/html/w32.yaha.f@mm.html> (May 25, 2003).
- [11] Panda Software. "How to Repair the Effects Caused by W32/Lentin.E?" Virus Encyclopedia. URL: http://www.pandasoftware.es/library/W32LentinE_en.htm#INICIO (March 9, 2003).
- [12] Sophos. "W32/Yaha-E." Sophos Virus Analysis: W32/Yaha-E. URL: <http://www.sophos.com/virusinfo/analyses/w32yahae.html> (May 25, 2003).
- [13] Central Command, Inc. "What Is W32/Yaha.E?" Top Virus Threats. http://support.centralcommand.com/cgi-bin/command.cfg/php/enduser/std_adp.php?p_refno=020617-000007 (March 9, 2003).
- [14] Computer Associates International, Inc. "Win32.Yaha.E." Virus Information Center. URL: <http://www3.ca.com/virusinfo/Virus.asp?ID=12380> (May 25, 2003).
- [15] CVE. "CVE-2001-0154." Common Vulnerabilities and Exposures. URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154> (May 25, 2003).
- [16] Microsoft TechNet. "Incorrect MIME Header Can Cause IE to Execute E-mail Attachment." Microsoft Security Bulletin (MS01-020). March 29, 2001. URL: <http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp> (May 25, 2003).
- [17] Postel, Jonathan B. "Simple Mail Transfer Protocol." RFC 821. August 1982. URL: <http://www.ietf.org/rfc/rfc821.txt> (May 25, 2003).
- [18] Crocker, David H. "Standard for the Format of ARPA Internet Text Messages." RFC 822. August 13, 1982. URL: <http://www.ietf.org/rfc/rfc822.txt> (May 25, 2003).

[19] Borenstein, N. and Freed, N. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." RFC 1521. September 1993. URL: <http://www.ietf.org/rfc/rfc1521.txt> (May 25, 2003).

[20] Moore, K. "MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text." RFC 1522. September 1993. URL: <http://www.ietf.org/rfc/rfc1522.txt> (May 25, 2003).

[21] SANS Institute. "Security Essentials." SANS Online Training. URL: <http://www.sans.org/onlinetraining/track1.php> (May 25, 2003).

[22] SANS Institute. "Hacker Techniques, Exploits and Incident Handling." SANS Online Training. URL: <http://www.sans.org/onlinetraining/track4.php> (May 25, 2003).

Appendix 1

SCAM32, SIRC32, WINK, ZONEALARM, AVP32, LOCKDOWN2000, AVP.EXE, CFINET32, CFINET, ICMON, SAFEWEB, WEBSCANX, ANTIVIR, MCAFEE, NORTON, NVC95, FP-WIN, IOMON98, PCCWIN98, F-PROT95, F-STOPW, PVIEW95, NAVWNT, NAVRUNR, NAVLU32, NAVAPSV, NISUM, SYMPROXYSVC, RESCUE32, NISSERV, ATRACK, IAMAPP, LUCOMSERVER, LUALL, NMAIN, NAVW32, NAVAPW32, VSSTAT, VSHWIN32, AVSYNMGR, AVCONSOL, WEBTRAP, POP3TRAP, PCCMAIN, PCCIOMON.