

Analysis of malware.exe

Done by Peter Kruse, CSIS

The file "malware.exe" (MD5: 59a95f668e1bd00f30fe8c99af675691) is a Win32 PE. It's packed with a hacked up UPX clone using "ABC" strings to hide its true packer (Ultimate Packer for eXecutables). Unpacking is trivial.

When the code is run it creates a mutex to ensure that only one instance of the file is running in memory at the same time. Next it drops "WinSec32.exe" into the windows folder. This code is clearly malicious and a typical RBOT. It uses the winsock32 (WS2_32.dll) to access the IRC based Command & Control server located at: "testirc1.sh1xy2bg.NET" using Internet Relay Chat Protocol on TCP port 6667 which is default IRC communication (<http://www.ietf.org/rfc/rfc1459.txt>).

Disassembling the code reveals the following behavior:

```
#000510 0x00401B42=WS2_32!gethostbyname ("testirc1.sh1xy2bg.NET")
```

IRC based communication, including the user name and password to logon to the IRC server, can be found below:

Applies nickname USA[XP]1686425.

Applies username lmqlxadet.

Joins channel #challenge with password happy12.

Sets the channel mode for channel #challenge to +mnst.

In order for the code to run after a reboot of the system it modifies registry and creates several run as keys using the "RegCreateKeyExA" command:

```
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".
```

```
HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices
```

```
"Microsoft Svchost local services"="Winsec32.exe" in key
```

This malware uses kernel32.dll for instructions to be carried out on the infected host. The process calls include:

GetProcAddress (0x7C800000,"LoadLibraryA")

GetProcAddress (0x7C800000,"GetProcAddress")

GetProcAddress (0x7C800000,"VirtualProtect")

GetProcAddress (0x7C800000,"VirtualAlloc")

GetProcAddress (0x7C800000,"VirtualFree")

GetProcAddress (0x7C800000,"ExitProcess")

and

GetProcAddress (0x7C800000,"HeapFree")

GetProcAddress (0x7C800000,"CreateFileA")

GetProcAddress (0x7C800000,"Sleep")

GetProcAddress (0x7C800000,"WriteFile")

GetProcAddress (0x7C800000,"CloseHandle")

GetProcAddress (0x7C800000,"HeapAlloc")

GetProcAddress (0x7C800000,"ExitThread")

Information gathering about the infected host and backdoor functionality include:

GetProcAddress (0x73350000,"ip_gethostname")

GetProcAddress (0x73350000,"ip_accept")

GetProcAddress (0x73350000,"ip_receive_data")

GetProcAddress (0x73350000,"ip_gethostbyname")

GetProcAddress (0x73350000,"ip_transfer_data")

GetProcAddress (0x73350000,"ip_retrieve_socket_data")
GetProcAddress (0x73350000,"ip_getservbyname")
GetProcAddress (0x73350000,"ip_connect")
GetProcAddress (0x73350000,"ip_listen_port")
GetProcAddress (0x73350000,"ip_bind_port")
GetProcAddress (0x73350000,"ip_close")
GetProcAddress (0x73350000,"ip_query_protocol")
GetProcAddress (0x73350000,"ip_reverse_dns")
GetProcAddress (0x73350000,"ip_select")
GetProcAddress (0x73350000,"ip_allocate_socket")
GetProcAddress (0x73350000,"ip_release_socket")

The data is used to categorize the host and later to connect to the machine to carry out commands through its backdoor backdoor functions.

Again it uses the Winsock32 library (through kernel32) for different operations:

GetProcAddress (0x733B0000,"WSAStartup")
GetProcAddress (0x733B0000,"WSASocketA")
GetProcAddress (0x733B0000,"WSAAsyncSelect")
GetProcAddress (0x733B0000,"__WSAFDIsSet")
GetProcAddress (0x733B0000,"WSAIoctl")
GetProcAddress (0x733B0000,"WSAGetLastError")
GetProcAddress (0x733B0000,"WSACleanup")
GetProcAddress (0x733B0000,"socket")
GetProcAddress (0x733B0000,"ioctlsocket")
GetProcAddress (0x733B0000,"connect")
GetProcAddress (0x733B0000,"inet_ntoa")
GetProcAddress (0x733B0000,"inet_addr")

```
GetProcAddress (0x733B0000,"htons")
GetProcAddress (0x733B0000,"htonl")
GetProcAddress (0x733B0000,"ntohs")
GetProcAddress (0x733B0000,"ntohl")
GetProcAddress (0x733B0000,"send")
GetProcAddress (0x733B0000,"sendto")
GetProcAddress (0x733B0000,"recv")
GetProcAddress (0x733B0000,"recvfrom")
GetProcAddress (0x733B0000,"bind")
GetProcAddress (0x733B0000,"select")
GetProcAddress (0x733B0000,"listen")
GetProcAddress (0x733B0000,"accept")
GetProcAddress (0x733B0000,"setsockopt")
GetProcAddress (0x733B0000,"getsockname")
GetProcAddress (0x733B0000,"gethostname")
GetProcAddress (0x733B0000,"gethostbyname")
GetProcAddress (0x733B0000,"gethostbyaddr")
GetProcAddress (0x733B0000,"getpeername")
GetProcAddress (0x733B0000,"closesocket")
```

Next up the code calls another library "ipstack.dll" to call the following commands:

```
GetProcAddress (0x73350000,"ip_downloadcontent")
GetProcAddress (0x73350000,"ip_crackurl")
```

This is to download additional malware to the system from the Command and Control server. To access the Internet and grab the file it calls wininet.dll and connects to the server where the malware is stored:

```
KERNEL32!GetProcAddress (0x771A0000,"InternetGetConnectedState")
KERNEL32!GetProcAddress (0x771A0000,"InternetGetConnectedStateEx")
```

```
KERNEL32!GetProcAddress (0x771A0000,"HttpOpenRequestA")
KERNEL32!GetProcAddress (0x771A0000,"HttpSendRequestA")
KERNEL32!GetProcAddress (0x771A0000,"InternetConnectA")
KERNEL32!GetProcAddress (0x771A0000,"InternetOpenA")
KERNEL32!GetProcAddress (0x771A0000,"InternetOpenUrlA")
KERNEL32!GetProcAddress (0x771A0000,"InternetCrackUrlA")
KERNEL32!GetProcAddress (0x771A0000,"InternetReadFile")
KERNEL32!GetProcAddress (0x771A0000,"InternetCloseHandle")
```

It uses a special browser agent - likely to validate that this indeed is a zombie - and to filter certain unwanted traffic:

```
"Mozilla/4.0 (compatible)"
```

By using the netapi32.dll library the malware is able to add a user account and enumerate a net share and send data across the internal network:

```
GetProcAddress (0x73390000,"NetShareAdd")
GetProcAddress (0x73390000,"NetShareDel")
GetProcAddress (0x73390000,"NetShareEnum")
GetProcAddress (0x73390000,"NetScheduleJobAdd")
GetProcAddress (0x73390000,"NetApiBufferFree")
GetProcAddress (0x73390000,"NetRemoteTOD")
GetProcAddress (0x73390000,"NetUserAdd")
GetProcAddress (0x73390000,"NetUserDel")
GetProcAddress (0x73390000,"NetUserEnum")
GetProcAddress (0x73390000,"NetUserGetInfo")
GetProcAddress (0x73390000,"NetMessageBufferSend")
```

Loading "mpr.dll" (mpr.dll is a module containing functions used to handle communication between the Windows operating system and the installed network providers) located in the windows system folder, the instructions below are carried out:

(0x733D0000,"WNetAddConnection2W")

(0x733D0000,"WNetCancelConnection2A")

(0x733D0000,"WNetCancelConnection2W")

- and shell32.dll is used for some SQL functions:

("odbc32.dll")

(0x74780000,"SQLDriverConnect")

(0x74780000,"SQLSetEnvAttr")

(0x74780000,"SQLExecDirect")

(0x74780000,"SQLAllocHandle")

(0x74780000,"SQLFreeHandle")

(0x74780000,"SQLDisconnect")

/Peter Kruse, 5th of October 2008