Shell Trojan Generators / Droppers / Backdoors Notes

From observations of running the “zwshell.exe” dropper on several infected client systems we observed that the dropper creates different service entry names in the registry upon execution. The service entry name is different from the ones we filled in the creation of the dropper in the dropper/C2 application. Some of the new service names we found are called:

- Nwsapagent
- NWCWorkstation

Others seen during the incident included:

- IAS
- ASP.NET
- IPRIP
- 6to4
- Web

With a search on the Internet we've found a reference dated from 2007 of a trojan that produces the same artifacts in the registry as our testing malware clients. The reference can be found here: http://www.mcafee.com/threat-intelligence/malware/default.aspx?id=143837

The malware referenced also connects to a command and control server called oandpsoftware.com. Additional reference relate to malware detected in 2009 and 2010 with similar artifacts.

Microsoft and McAfee have attributed the dropper/C2 application to Gh0st: http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Backdoor%3aWin32%2fRemosh.A

There are some notable differences between past Gh0st applications and this version (though it is worth noting that one version of the file discovered during the incident is named “ghost”); however functionality is fairly common to these types of “point and click” or (WYSIWYG) remote administration tools. Research into attribution through Trend, McAfee, and Microsoft resources is continuing.
Host-Based Signatures

Related Registry Values

- The malware variant named shelldc.dll is designed to be installed as a service named las with a DisplayName of ASP.NET Service and a Description of “Provides support for out-of-process session states for ASP.NET.”
- The malware variant named recyle64.dll is designed to be installed as a service named las with a DisplayName of ASP.NET Services and no description.
- The malware variant named hpmdp093.dll is designed to be installed as a service named 6to4 with a DisplayName of OfficeScan Support and a Description of “Enables Help and Support Center to run Officescan on this computer “
- The malware variant named ws_18.dll is designed to be installed as a service named Iprip with a DisplayName of Network Management and a Description of “Provides network installation services such as Assign, Publish, and Remove”

File System Residue

- The malware binary will contain between at least 200 and 400 bytes of data at the end of the file, beyond the structure of the PE file format. This data is known as EOF data or an Overlay.
- The malware will end 128 bytes of data, each byte incrementing by one from the number 0. The following is an example from the shelldc.dll binary analyzed:

```
00052d0: 0001 0203 0405 0607 0809 0a0b 0c0d 0e0f ..............
00052e0: 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f ..............
00052f0: 2021 2223 2425 2627 2829 2a2b 2c2d 2e2f !"#$%&'()*+,-./
0005300: 3031 3233 3435 3637 3839 3a3b 3c3d 3e3f 0123456789;:<=>?
0005310: 4041 4243 4445 4647 4849 4a4b 4c4d 4e4f @ABCDEFGHJKLMNO
0005320: 5051 5253 5455 5657 5859 5a5b 5c5d 5e5f PQRSTU VWXYZ[\]^_
0005330: 60                                          ```
- The malware exports a single function named ServiceMain and imports functions from GDI32.DLL named BitBlt and GetDIBits. These imported functions are considered suspicious in a service dll.

Volatile Evidence

- The malware creates a mutex object to ensure that only one instance of the malware is running at a time. Each variant is configured with a mutex name in its configuration data block at the end of the file. The mutex used by each variant is listed below:
  - shelldc.dll – shelldc
  - recyle64.dll – NT1630
  - hpmdp093.dll – w
  - ws_18.dll – shellsa

MD5 Signatures

- The following MD5 signatures have been discovered through live response and forensic analysis:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (Bytes)</th>
<th>MD5 Hash Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>ws_data.dll</td>
<td>20,753</td>
<td>1A0A38E8AAC50CF1825739389E904B44</td>
</tr>
<tr>
<td>ws_data.dll</td>
<td>20,753</td>
<td>549dff76af0dd9e536a6d9c4d499065</td>
</tr>
</tbody>
</table>
Network-Based Signatures

- The malware variants are designed to communicate to a remote host for command and control. The specific host (either hostname or IP address) and TCP port number are specified in the configuration data block at the end of the binary. The host and port that each variant communicates to is listed below:
  - shelldc.dll – shell.is-a-chef.com:3128
  - recycle64.dll – shell.is-a-chef.com:80
  - hpmdp093.dll – 134.146.82.25:53
  - ws_18.dll – shell.is-a-chef.com:80

- The malware initiates communication with the remote host with a 16-byte beacon packet which begins with the bytes 0x01 0x50 and ends with the bytes 0x68 0x57 0x24 0x13
Details
The malware is designed to be installed as a service by a secondary installer tool and provides no self-installation routine of its own. The first operation taken by the malware is to decode its configuration data block. It reads the last 737 bytes of its own file and decodes this data into a data structure used by the program. The algorithm used to decode the data is to XOR each byte with the remainder of index from the beginning of the structure divided by 128. The following python script illustrates the decoding algorithm:

```python
f = open('shelldc.dll', 'rb')
f.seek(-737, 2)
buf = f.read()
buf2 = ''
for i in xrange(len(buf)):
    buf2 += chr(ord(buf[i]) ^ (i % 128))
print buf2  # decoded data
```

Figure 1 - Algorithm to Decode Configuration Block for shelldc.dll

The configuration structure contains the following fields when decoded (fields highlighted with unique colors, descriptions following data fragment).
When the malware is launched as a service (hosted in an svc host.exe process) its exported function ServiceMain is called. This function begins by checking the existence of a mutex object. The name of the mutex is specified in the configuration data block. If the mutex exists, the malware exits. This ensures that only one instance of the malware is running on the current host at a time.

After creating the mutex the malware begins its backdoor loop functionality. It first waits 30 seconds before initiating a connection to the remote command and control server. To communicate with the remote server it sends a beacon packet.

Below you'll find the codes used by the malware that we've found in our network traces.

“All malware communications with the server follow a well-defined protocol. Each transmission begins with a 16-byte packet containing several fields such as a message type code. Message type code 0x5001 denotes the beacon packet. The following table describes the structure of 16-byte message packet.”

<table>
<thead>
<tr>
<th>Offset</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2 Bytes</td>
<td>Message Type Code (50XX for outbound beacon and responses, 60XX for inbound commands)</td>
</tr>
<tr>
<td>2</td>
<td>1 Byte</td>
<td>Flag to signify that additional data will follow this packet</td>
</tr>
<tr>
<td>3</td>
<td>4 Bytes</td>
<td>Length of additional data (if flag is set)</td>
</tr>
<tr>
<td>7</td>
<td>4 Bytes</td>
<td>Variable 1 (unused by the program)</td>
</tr>
<tr>
<td>11</td>
<td>1 Byte</td>
<td>Variable 2 (for commands that need to transmit a small amount of data)</td>
</tr>
</tbody>
</table>
The following table describes the commands accepted by the malware:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6001</td>
<td>Exit the command loop</td>
</tr>
<tr>
<td>6002</td>
<td>Causes the client to respond with a 5003 message with no data (heartbeat)</td>
</tr>
<tr>
<td>6003</td>
<td>Set the service name and group name in the registry to names specified in the additional data payload.</td>
</tr>
<tr>
<td>6004</td>
<td>Move the malware DLL to the same filename plus &quot;.cnt&quot; and sets the current ServiceDll registry value to the DLL name. It then schedules both the malware DLL and &quot;.cnt&quot; file to be deleted upon reboot.</td>
</tr>
<tr>
<td>6005</td>
<td>Retrieve a &quot;+&quot; delimited list of usernames that are currently connected to the host via a Remote Desktop session. (Response message code 5004 with additional data)</td>
</tr>
<tr>
<td>6011</td>
<td>Retrieve a list of currently connected drive letters with basic filesystem information including total size, free space and volume ID. (Response message code 5011 with additional data)</td>
</tr>
<tr>
<td>6012</td>
<td>Retrieve a list of directory entries for a specified directory. Directory specified with additional data following the command message packet. (Response message code 5012 with additional data)</td>
</tr>
<tr>
<td>6013</td>
<td>Shell Execute. Command or document to open is specified in the additional data following the command message packet. (Response message code 5100 with Variable 2 byte set to 2 if execution was successful)</td>
</tr>
<tr>
<td>6014</td>
<td>Move File. Filenames from and to are specified in the additional data following the command message packet. (Response message code 5100 with Variable 2 byte set to 4 if the move failed and 3 if it succeeded)</td>
</tr>
<tr>
<td>6015</td>
<td>Delete File. Filename to delete is specified in the additional data following the command message packet. (Response message code 5100 with Variable 2 byte set to 6 if the delete failed and 5 if it succeeded)</td>
</tr>
<tr>
<td>6016</td>
<td>Copy File. Filenames from and to are specified in the additional data following the command message packet. (Response message code 5100 with Variable 2 byte set to 8 if the copy failed and 7 if it succeeded)</td>
</tr>
<tr>
<td>6017</td>
<td>Set File Attributes. The First 4 bytes of the additional data payload following the command message packet specify the file attributes and data starting at offset 4 in the additional payload specifies the filename. (Response message code 5100 with Variable 2 byte set to 10 if the operation failed and 9 if it succeeded)</td>
</tr>
<tr>
<td>6018</td>
<td>Enter File Management Loop. Commands 6019 – 601D accepted.</td>
</tr>
<tr>
<td>6019</td>
<td>Exits File Management Loop. (Response message code 5013)</td>
</tr>
<tr>
<td>601A</td>
<td>Upload File to Remote Server. Must be in File Management Loop. File name to upload is specified in the additional data payload following the command message packet. The file transfer begins and ends with message code 5014. When the Variable 2 byte is set to 1 it denotes the beginning of the file transfer session and when it is set to 0 it denotes the end of the session. All content following the initial 5014 message to begin the transfer will be sent with response message code 5015 with up to 4K of additional data per message.</td>
</tr>
<tr>
<td>601B</td>
<td>Download A File. Must be in File Management Loop. This command begins a download from the remote server. Additional data may be downloaded into the file with command message code 601C.</td>
</tr>
<tr>
<td>601C</td>
<td>Download Data To Current File. Must be in File Management Loop. This command is used after a 601B command to download up to 4K of additional data per message.</td>
</tr>
<tr>
<td>601D</td>
<td>Retrieve Directory Entries. Must be in File Management Loop. (Response message code 5014 to begin the list and 5015 for each entry)</td>
</tr>
</tbody>
</table>
| 601F | Retrieve a Recursive Directory Size. This command causes the malware to traverse the directory and calculate the total size of data under the directory. Due
to improper coding techniques used by the malware author the program will produce inaccurate results in directories containing large files (greater than 4GB). (Response message code 501F returned)

6021 **Launch Interactive Command Shell.**
6031 **Enter Registry Browser Mode.** Registry key to start with specified in additional data payload. (Responds with message code 5001 with Variable 2 set to 4 to begin)

6032 **Enumerate Registry Subkeys and Values.** Must be in registry browser mode (command 6031). (Responds with return code 5100 with Variable 2 set to 10 if the operation failed and 9 if it succeeded)

6041 **Operate the Desktop Remotely.** The additional data payload specifies a command structure with input operations such as mouse and keyboard. Responses contain screen capture information, providing an effective Remote Desktop implementation. Individual input commands are described later in this document.

---

We've attached a screenshot of a tcp stream of a session between the C&C server and an infected client. All the lines in red are client traffic and the blue lines is the C&C server traffic.

The final value of the message packet is the magic number of 0x13245768 in little endian byte order (hw$). This is also the same magic number that is at the beginning of the decoded configuration data block of the produced malware clients by the zwShell.exe C&C application.

As you can see on the third line of the tcp stream you will find a transmitted password '1234'. After the handshake the client sends his system information to the C&C server. In this session we traversed a directory and retrieved a file.
Once the dropper gets created from the C&C application a port can be specified. This port will be found in the zwShell.ini file. We've already found such references on ports 1026 and 3128 on several systems.

**Instructions to run zwShell.exe**

For some reason zwShell.exe breaks with this error:
It’s not known yet what causes this error (possibly a missing ini file), but there is a quick workaround to get it to run properly:

1. Load zwShell.exe in Olly Debugger

2. Make sure the debugger is not detected by the ASProtect packer. Olly IsDebuggerPresent plugin works fine.

3. Configure Olly to ignore Access violation errors by adding the C0000005 exception in debugging options -> exceptions -> Ignore also following custom exceptions or ranges

4. Start zwShell

5. When the error message pops up, set a breakpoint at 00497FDA and press ok at the error dialog. See screenshot below.

6. Olly breaks at 00497FDA, where BL=0. Make sure the following JNZ jmp is taken by changing BL to 01.

7. zwShell will now run properly.
Screenshots

The following screen shots were collected from a system where the dropper/C2 was run in debug mode and used to infect a client machine.
From HBGary Dump (header added by Cryptome)

251658240

SECRET

[Registry Dump Image]

251658240
The following Screenshot was collected from a VM that was infected with Sver.exe dropper which configured the Connect.dll and Startup.dll backdoor services:

![Screenshot of Connect.dll and Startup.dll backdoor services]

The following Screenshot was collected from a VM that was infected with the Server.exe dropper, which configured the ShellDC.dll backdoor services:

![Screenshot of ShellDC.dll backdoor services]
The following was collected from Sandbox analysis of the Server.exe dropper and ShellDC.dll backdoor:

**Processes:**

<table>
<thead>
<tr>
<th>PID</th>
<th>ParentPID</th>
<th>User</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1656</td>
<td>652</td>
<td>NT AUTHORITY:SYSTEM</td>
<td>C:\WINDOWS\system32\svchost.exe</td>
</tr>
</tbody>
</table>

**Ports:**

<table>
<thead>
<tr>
<th>Port</th>
<th>PID</th>
<th>Type</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>shell.is-a-chef.com</td>
</tr>
</tbody>
</table>

**Explorer Dlls:**

<table>
<thead>
<tr>
<th>DLL Path</th>
<th>Company Name</th>
<th>File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No changes Found

**IE Dlls:**

<table>
<thead>
<tr>
<th>DLL Path</th>
<th>Company Name</th>
<th>File Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No changes Found

**Loaded Drivers:**

<table>
<thead>
<tr>
<th>Driver File</th>
<th>Company Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---------
Monitored RegKeys
Registry Key Value

Hklm\SYSTEM\CurrentControlSet\Services 6to4

Kernel31 Api Log

***** Installing Hooks *****
71ab70df  RegOpenKeyExA (Hklm\System\CurrentControlSet\Services\WinSock2\Parameters)
71ab7cc4  RegOpenKeyExA (Protocol_Catalog9)
71ab737e  RegOpenKeyExA (00000005)
71ab724d  RegOpenKeyExA (Catalog_Entries)
71ab78ea  RegOpenKeyExA (000000000001)
71ab78ea  RegOpenKeyExA (000000000002)
71ab78ea  RegOpenKeyExA (000000000003)
71ab78ea  RegOpenKeyExA (000000000004)
71ab78ea  RegOpenKeyExA (000000000005)
71ab78ea  RegOpenKeyExA (000000000006)
71ab78ea  RegOpenKeyExA (000000000007)
71ab78ea  RegOpenKeyExA (000000000008)
71ab78ea  RegOpenKeyExA (000000000009)
71ab78ea  RegOpenKeyExA (000000000010)
71ab78ea  RegOpenKeyExA (000000000011)
71ab78ea  RegOpenKeyExA (000000000012)
71ab78ea  RegOpenKeyExA (000000000013)
71ab2623  WaitForSingleObject(79c,0)
71ab83c6  RegOpenKeyExA (NameSpace_Catalog5)
71ab737e  RegOpenKeyExA (00000004)
71ab7f5b  RegOpenKeyExA (Catalog_Entries)
71ab80ef  RegOpenKeyExA (000000000001)
71ab80ef  RegOpenKeyExA (000000000002)
71ab80ef  RegOpenKeyExA (000000000003)
71ab2623  WaitForSingleObject(794,0)
71aa1afa  RegOpenKeyExA (HKLM\System\CurrentControlSet\Services\Winsock2\Parameters)
71aa1996  GlobalAlloc()
7c80b511  ExitThread()
40144b  GlobalAlloc()
77de5f5e  WaitForSingleObject(7e4,2bf20)
77de9f8e  RegOpenKeyExA (HKLM\Software\Microsoft\Rpc)
4010c0  RegOpenKeyExA (HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Svchost)
4010c0  RegOpenKeyExA (HKLM\SYSTEM\CurrentControlSet\Services\6to4)
4014f8  CreateMutex(shelldc)
40136f  CreateFileA(C:\WINNT\System32\shelldc.dll)
40138e  WriteFile(h=770)
401060  RegCreateKeyExA
(HKLM\SYSTEM\CurrentControlSet\Services\6to4\Parameters,(null))
401085  RegSetValueExA (ServiceDll)
7c816d55 ExitThread()
7c80cd0c ExitProcess()
***** Injected Process Terminated *****

DirwatchData

-------------------------------------------------
WatchDir Initialized OK
Watching C:\DOCUME~1\xp\LOCALS~1\Temp
Watching C:\WINDOWS
Watching C:\Program Files
Modified: C:\WINDOWS\system32\config\system.LOG
Created: C:\WINDOWS\system32\shelldc.dll
Modified: C:\WINDOWS\system32\shelldc.dll
Modified: C:\WINDOWS\system32
Modified: C:\WINDOWS\system32\wbem\Logs\wbemess.log
Created: C:\WINDOWS\Prefetch\SNIFF_HIT.EXE-1AB02EA8.pf
Modified: C:\WINDOWS\Prefetch\SNIFF_HIT.EXE-1AB02EA8.pf
Created: C:\\DOCUME~1\xp\LOCALS~1\Temp\JET37D0.tmp
Created: C:\DOCUME~1\xp\LOCALS~1\Temp\JET37D0.tmp
Deleted: C:\DOCUME~1\xp\LOCALS~1\Temp\JET35.tmp
Deleted: C:\DOCUME~1\xp\LOCALS~1\Temp\JET37D0.tmp

File: svchost.exe
Size: 14336 Bytes
MD5: 8F078AE4ED187AAABC0A305146DE6716
Packer: File not found C:\iDEFENSE\SysAnalyzer\peid.exe

File Properties:
CompanyName Microsoft Corporation
FileDescription Generic Host Process for Win32 Services
FileVersion 5.1.2600.2180 (xpsp_sp2_rtm.040803-2158)
InternalName svchost.exe
LegalCopyright © Microsoft Corporation. All rights reserved.
OriginalFileName svchost.exe
ProductName Microsoft® Windows® Operating System
ProductVersion

Exploit Signatures:
-------------------------------------------------
Scanning for 19 signatures
Scan Complete: 24Kb in 0 seconds
Urls
-------------------------------------------------

RegKeys
Software\Microsoft\Windows NT\CurrentVersion\Svchost

ExeRefs

File: svchost_dmp.exe_
svchost.exe
svchost.exe

Raw Strings:

File: svchost_dmp.exe_
MD5: d7c3d5fc02b6be4acc707157af3e2337
Size: 24578

Ascii Strings:

!This program cannot be run in DOS mode.
5Rich
.text
`.data
.rsrc
ADVAPI32.dll
KERNEL32.dll
NTDLL.DLL
RPCRT4.dll
SvchostPushServiceGlobals
ServiceMain
Y@PVPVh
VWh @
|h @
95@@
F;5@@
SVW3
_^[t
uV9)
VVVV
t6PV
t!VV
QSV3
Wh @
95@@
;5@@
WhT@
QQSVWd
u-SS
Ph\!
Ph4!
NETAPI32.dll
ole32.dll
Netbios
CoInitializeEx
CoInitializeSecurity
ADVAPI32.dll
KERNEL32.dll
ntdll.dll
RPCRT4.dll
RegQueryValueExW
SetSecurityDescriptorDacl
SetEntriesInAclW
SetSecurityDescriptorGroup
SetSecurityDescriptorOwner
InitializeSecurityDescriptor
GetTokenInformation
OpenProcessToken
OpenThreadToken
SetServiceStatus
RegisterServiceCtrlHandlerW
RegCloseKey
RegOpenKeyExW
StartServiceCtrlDispatcherW
HeapFree
GetLastError
WideCharToMultiByte
lstrlenW
LocalFree
GetCurrentProcess
GetCurrentThread
GetProcAddress
LoadLibraryExW
LeaveCriticalSection
HeapAlloc
EnterCriticalSection
LCMapStringW
FreeLibrary
lstrcpyW
ExpandEnvironmentStringsW
lstrncpyW
ExitProcess
GetCommandLineW
InitializeCriticalSection
GetProcessHeap
SetErrorMode
SetUnhandledExceptionFilter
RegisterWaitForSingleObject
InterlockedCompareExchange
LoadLibraryA
QueryPerformanceCounter
GetTickCount
GetCurrentThreadId
GetCurrentProcessId
GetSystemTimeAsFileTime
TerminateProcess
UnhandledExceptionFilter
LocalAlloc
lstrcmpW
DelayLoadFailureHook
NtQuerySecurityObject
RtlFreeHeap
NtOpenKey
wcscat
wcscpy
RtlAllocateHeap
RtlCompareUnicodeString
RtlInitUnicodeString
RtlInitializeSid
RtlLengthRequiredSid
RtlSubAuthoritySid
NtClose
RtlSubAuthorityCountSid
RtlGetDaclSecurityDescriptor
RtlQueryInformationAcl
RtlGetAce
RtlImageNtHeader
wcslen
RtlUnhandledExceptionFilter
RtlCopySid
RpcServerUnregisterIfEx
RpcMgmtWaitServerListen
RpcMgmtSetServerStackSize
RpcServerUnregisterIf
RpcServerListen
RpcServerUseProtseqEpW
RpcServerRegisterIf
I_RpcMapWin32Status
RpcMgmtStopServerListening
RSDS
svchost.pdb

Unicode Strings:
---------------------------------------------------------------------------------
Parameters
System\CurrentControlSet\Services
nServiceMain
ServiceDll
ServiceDllUnloadOnStop
eventlog
ncacn_np\PIPE\
DefaultRpcStackSize
AuthenticationCapabilities
ImpersonationLevel
AuthenticationLevel
CoInitializeSecurityParam
Software\Microsoft\Windows NT\CurrentVersion\Svchost\Registry\Machine\System\CurrentControlSet\Control\SecurePipeServers\VS_VERSION_INFO
StringFileInfo
040904B0
CompanyName
Microsoft Corporation
FileDescription
Generic Host Process for Win32 Services
FileVersion
5.1.2600.2180 (xpsp_sp2_rtm.040803-2158)
InternalName
svchost.exe
LegalCopyright
Microsoft Corporation. All rights reserved.
OriginalFilename
svchost.exe
ProductName
Microsoft
Windows
Operating System
ProductVersion
5.1.2600.2180
VarFileInfo
Translation

_Shell32.dll Strings info:_

----------------------------------------
!This program cannot be run in DOS mode.
*Ritch27
.text
`.rdata
@.data
.reloc
uY9E
uL9E
uB9E
u.9E t
9M u$9E$t
F@lu
vfH3
@AOu
@ANu
@ANu
@GNu
OAA
@GNu
BAOu
BANu
BANu
BGNu
OAA
BGNu
SVf
D0CPj@
PWW
HZx"
AF:t$"t3VWh
WSSj
jAP
hPa
Wh8a
PSSh
hxa
hla
SVW3
PSSWj
SSSSSh
SVW3
VSP
SQP
QQQQQ
SVWj
SSj
SSh
VWSP
v)Sj
WSSh
veSS
PSS
SSSSSh
SSSSSh
SSh
Ht\Ht$H
SSj
SSh
-SSSSSh
SSSSSh
SVW
^VSP
RPQ
_\WSP
PSSS
SPSSS
@SPj
<SVW3
SSh
SPh
VWP
PPPQ
PPP
VVVVVh1P
VWP
VVVVVh2P
Hff
SUV
(SUV
t$8WV
[SWP
j(QP
j(QP
FfW
PWW
v6PW
v2WW
vZj
hAP
v6WW
hBP
QQSV
WSSj
ShCp
v6SS
hDP
PWW
vNj
Vj j
UVW
SSSSj
SSSSj
SSSSj
SSSSj
SSSSj
PSSH
tSV
SSH
SPPhb9
SUIV3
PWW
SSSh
SSPSS
SSt
SSt
SSt
SSt
SSt
SSsWV
SUIVVPh
SSH
SSH
uYVPP
YYj
YYV
uRFGHt
GIu
GJu
wQn
wec
wFX
qMf
lstrcpyA
Sleep
GlobalFree
GlobalUnlock
GlobalHandle
GlobalLock
GlobalAlloc
lstrlenA
lstrcpynA
lstrcatA
DeleteFileA
CloseHandle
ReadFile
SetFilePointer
CreateFileA
GetVersionExA
GlobalMemoryStatus
GetComputerNameA
FreeLibrary
GetProcAddress
LoadLibraryA
MoveFileExA
MoveFileA
GetVolumeInformationA
GetDiskFreeSpaceExA
GetDriveTypeA
SetErrorMode
FindClose
FindNextFileA
GlobalReAlloc
lstrcmpA
FindFirstFileA
GetFileSize
WriteFile
TerminateProcess
GetExitCodeProcess
PeekNamedPipe
CreateProcessA
CreatePipe
CopyFileA
GetTempPathA
GetSystemDirectoryA
WaitForSingleObject
lstrcmpiA
CreateThread
GetCurrentThreadId
SetFileAttributesA
GetTickCount
GetModuleFileNameA
ReleaseMutex
CreateMutexA
OpenMutexA
KERNEL32.dll
CloseDesktop
SetThreadDesktop
OpenInputDesktop
GetThreadDesktop
ReleaseDC
GetDC
PostMessageA
OpenDesktopA
WindowFromPoint
GetCursorPos
SetCursorPos
MapVirtualKeyA
keybd_event
mouse_event
GetSystemMetrics
USER32.dll
ControlService
CloseServiceHandle
OpenServiceA
OpenSCManagerA
RegCloseKey
RegSetValueExA
RegOpenKeyExA
RegQueryValueExA
SetServiceStatus
RegEnumKeyExA
RegEnumValueA
RegisterServiceCtrlHandlerA
ADVAPI32.dll
WS2_32.dll
ShellExecuteA
SHELL32.dll
DeleteObject
DeleteDC
GetDIBits
BitBlt
SelectObject
CreateCompatibleBitmap
CreateCompatibleDC
GDI32.dll
Server.dll
ServiceMain
Oct 17 2005
Start
ProcessorNameString
HARDWARE\DESCRIPTION\System\CentralProcessor\0
GroupName
ServiceName
SYSTEM\CurrentControlSet\Services\open
Description
DisplayName
Type
PolicyAgent
WTSSession
WTSQuerySessionInformationA
WTSEnumerateSessionsA
wtsapi32.dll
ServiceDll
\Parameters
.cnt
svchost.exe
\cmd.exe
exit
Winlogon
c:\windows\system32\shelldc.dll
shell
shelldc
ias
ASP.NET Service
Provides support for out-of-process session states for ASP.NET.
shell.is-a-chef.com
C:\Documents and Settings\xp\Desktop\New Folder\Ugtxgt0gzg
hell.is-a-chef.com
=9>E>g>p>{>
0G0X0n0{0
4M4S4e4k4s4
5:5V5q5x5
6"-6=6C6N6T6
6J7O7X7~7
?D?b?k?{?
8#9)969Q9X9z9
9C:`:g:x:
="/>|>=|}t>{>
0M0W0)0I0r0w0)0
0"1)12181>1D1J1P1V1\1b1h1n1t1z1
2"2(2.242:2@2F2L2R2X2^2d2|2p2v2|2
3S3*30363\3B3H3N3T3Z3`3f3I3r3x3~3
wmckd
!"#$%&'(*+,-./0123456789:\;<=?>@ABCD\^_abcdefghijklm`=?^7T&
!"#$%&'(*+,-./0123456789:\;<=?>@ABCDEFG