Mandiant Malicious Program Analysis Report
PRINKUI.DLL [AGENT.BTZ V2.04] (DOWNLOADER)

Summary

This malware family is commonly referred to as "Agent.BTZ" in the public domain. Its primary method of propagation is through compromised removable media (USB thumb drives, external hard drives, etc.) using an `autorun.inf` installer. Once installed the Agent.BTZ malware acts as a backdoor framework, capable of downloading commands and additional malware, system survey collection, remotely executing files, process injection, and infecting new removable media as it is connected to the host. Each instance of the Agent.BTZ malware will have a randomized .DLL filename. The malware has internal versioning and a robust logging component that tracks its activity across hosts.

File Characteristics

Filename	Size	MD5 checksum	Compile Time
prinkui.dll	180,224	1B1E985F38D6A5A21BD3F7955252D2A2	2008-06-09T17:23:56Z

Figure 1: File Characteristics for prinkui.dll

Host Based Signatures

Related Registry Values:

- The malware creates the key:
  - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\StrtdCfg
- The malware stores its configuration parameters as eighteen (18) individual sub keys of the `StrtdCfg` key above:
  - REG_BINARY: Id
  - REG_DWORD: Timout
  - REG_DWORD: IsActive
  - REG_DWORD: BSlp
  - REG_DWORD: SDCnt
  - REG_DWORD: LastValue
  - REG_DWORD: StVal
  - REG_DWORD: EmtParam
  - REG_DWORD: HtParam
  - REG_DWORD: CMValue
  - REG_DWORD: ILevelCount
  - REG_BINARY: IList
  - REG_DWORD: IListLen
  - REG_DWORD: Installed
  - REG_BINARY: IPlace
  - REG_BINARY: ISFValue
  - REG_BINARY: LastId
  - REG_DWORD: NTries
- For persistence, the malicious DLL is added as an in-process server to be loaded by `explorer.exe`. The CLSID is generated during runtime:
  - HKEY_CLASSES_ROOT\Software\Classes\CLSID\*<generated_clsid>*\InprocServer32

File System Residue:

- The malware creates multiple temporary files:
  - *%SystemRoot%*\1.txt
  - *%SystemRoot%*\system32__1.dat
  - *%TEMP%*\6D73776D706461742E746C62FA.tmp
- The malware copies itself as a randomized filename:
  - *%SystemRoot%*\system32*<random_name>*.dll
- The malware creates an creates an obfuscated log files:
  - *%SystemRoot%*\system32\mswmpdat.tlb
  - *%SystemRoot%*\system32\wmcache.nld
- The malware may also create these additional obfuscated files:
  - *%SystemRoot%*\system32\winview.ocx

  - `%SystemRoot%\system32\mssysmgr.ocx`
- The malware may download/execute the following files:
  - `%TEMP%\$1F.dll`
  - `%SystemRoot%\system32\tapi32d.exe`
  - `%SystemRoot%\system32\typecli.exe`
  - `%SystemRoot%\system32\msnet.exe`
  - `%SystemRoot%\system32\msnet32.exe`

Volatile Evidence:

- Registers a Window Class named “`zQWwe2esf34356d`”
- Contains “Java Virtual Mashine” misspelling in FileDescription metadata field
- Creates Mutex Object(s) named “`Mutex_Log`” under `explorer.exe`
  - `Mutant \BaseNamedObjects\Mutex_Log`

Network Based Signatures

- The malware will download:
  - `http://worldnews.ath.cx/update/img0008/228925349891072.jpg`

Details

The Agent.BTZ malware is typically installed via compromised removable media (USB thumb drives, external hard drives, etc.) using an `autorun.inf` installer; Windows hosts with AutoRun functionality enabled are vulnerable. The malware is installed by calling its “InstallM” export function:

```
rundll32.exe .\\<random_name>.dll,InstallM
```

Figure 2: Sample Autorun.inf Install Command

The malware binary is not packed, however the majority of its internal strings are obfuscated. Whenever the malware’s “DllMain” function is invoked, it removes the string obfuscation by XOR-ing each character with 0x55. Dynamically imported function names and strings used for network communication remain obfuscated until they are used by the malware.

During installation, the Agent.BTZ malware creates the following registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\StrtdCfg`. This key contains configuration settings for the malware, stored in eighteen (18) different sub keys:

```
Type:         Sub Key:      Initial Value:
REG_BINARY:   Id            // 0x000C29D634D00000
REG_DWORD:    Timout        // 0x00
REG_DWORD:    IsActive      // 0x00
REG_DWORD:    BSlp          // 0x00
REG_DWORD:    SDCnt         // 0x00
REG_DWORD:    LastValue     // 0x4BA13E57
REG_DWORD:    StVal         // 0x00
REG_DWORD:    EmtParam      // 0x2A30
REG_DWORD:    HtParam       // 0x093A80
REG_DWORD:    CMValue       // 0x00
REG_DWORD:    ILevelCount   // 0x03
REG_BINARY:   IList         // [binary data blob]
REG_DWORD:    IListLen      // 0x00
REG_DWORD:    Installed     // 0x01
REG_BINARY:   IPlace        // [binary data blob] “worldnews.ath.cx”
REG_BINARY:   ISFValue      // [binary data blob]
REG_BINARY:   LastId        // 0x7460713274607132
REG_DWORD:    NTries        // 0x00
```

Figure 3: Sample Registry Configuration Data for Installed Agent.BTZ

These configuration settings are updated frequently while the malware is operational.

The malware will then copy itself to `%SystemRoot%\system32`. It will create a randomized file name using characters from existing DLL files within `system32`; therefore each instance of the Agent.BTZ malware DLL will have a different filename. For persistence across reboots, the copied DLL is registered as an in-process server to be loaded by `explorer.exe` (`HKEY_CLASSES_ROOT\Software\Classes\CLSID\<generated_clsid>\InprocServer32`). The CLSID is generated at the time of install. The Agent.BTZ malware DLL will then be loaded automatically whenever `explorer.exe` starts on the compromised host.

The malware will attempt to download: `http://worldnews.ath.cx/update/img0008/228925349891072.jpg` once every 24 hours. The DLL will spawn an instance of the default web browser (`IEXPLORE.EXE`, `firefox.exe`) to make the download request. If this download is successful, the data would be saved as `%TEMP%\$1F.dll`. The Agent.BTZ malware is capable of injecting the downloaded binary into the process space of Internet Explorer.

The malware has a robust logging component, recording its activity on the compromised host to the file `%SystemRoot%\system32\mswmpdat.tlb`. This log file is encrypted with a 97-character XOR key:

```
key:
1dM3uu4j7Fw4sjnbcwlDqet4F7JyuUi4m5Imnxl1pzxI6as80cbLnmz54cs5Ldn4ri3do5L6gs923HL34x2f5cvd0fk6c1a0s
```

Figure 4: Mswmpdat.tlb Log File XOR Key

The sample log provided below depicts the installation of the malware, and what appear to be attempts to create additional processes (these executables are not created by the malware):

```
11:26:41 17.03.2010 Log begin:
11:26:41 TVer=2.2
11:26:41 AppendLog=1
11:26:41 Installing to C:\WINDOWS\system32\ufshim.dll
11:26:41 Copying C:\BTZ\prinkui.dll to C:\WINDOWS\system32\ufshim.dll (0)
11:26:41 Native Id: 7CEE1527
11:26:41 Log end.
11:26:46 17.03.2010 Log begin:
11:26:46 TVer=2.2
11:26:46 AppendLog=1
11:26:46 Timout: 604800 (sec); Passive interval: 1268839606 (sec)
11:26:46 Creating ps C:\WINDOWS\system32\tapi32d.exe (2)
11:26:46 Creating ps C:\WINDOWS\system32\typecli.exe (2)
11:26:46 Creating ps C:\WINDOWS\system32\msnet.exe (2)
11:26:46 Creating ps C:\WINDOWS\system32\msnet32.exe (2)
11:26:46 Cndr: 1 NoSl: 1 IsInt: 1
11:26:46 Log end.
```

Figure 5: Sample Decrypted Mswmpdat.tlb Log File

The malware will also collect system information from the compromised host and save it to the file `%SystemRoot%\system32\wmcache.nld`. This information is stored in an XML document that is also encrypted using the same 97-character XOR key:

```
<?xml version="1.0" encoding="unicode"?>
<Cfg>
<Ch>
<TVer>2.1</TVer>
<AppendLog>0</AppendLog>
<add key="Id" value="228925349891072" />
<add key="PVer" value="Ch 2.04" />
<add key="Http address" value="worldnews.ath.cx" />
<add key="Http timeout" value="10080" />
<add key="Processing volumes" value="0" />
<add key="Cure mode" value="0" />
```

```
<add key="Infecting level" value="3" />
<add key="File to send" value="" />
<add key="Folder" value="img0008" />
<add key="Time" value="17:03:2010 16:41:05" />
<add key="Bias" value="4294967288" />
<add key="PcName" value="VICTIM_HOSTNAME" />
<add key="UserName" value="user" />
<add key="WinDir" value="C:\WINDOWS" />
<add key="TempDir" value="C:\DOCUME~1\user\LOCALS~1\Temp\" />
<add key="WorkDir" value="C:\WINDOWS\" />
<add key="Cndr" value="0" />
<add key="NetAdapter" value="AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler
Miniport">
<add key="MAC" value="00:0C:29:D6:34:D0" />
<add key="Type" value="6" />
<add key="Ip" value="">
<add key="Ip" value="192.168.160.129" />
</add>
<add key="Mask" value="255.255.255.0" />
<add key="Gateway" value="">
<add key="Gateway" value="" />
</add>
<add key="DHCPServer" value="">
<add key="DHCPServer" value="192.168.160.254" />
</add>
<add key="DnsServer" value="">
<add key="DnsServer" value="192.168.160.130" />
</add>
</add>
<add key="List" value="">
</add>
<add key="NList" value="">
</add>
<add key="IdChain" value="">
</add>
</Ch>

</Cfg>
```

Figure 6: Sample Decrypted Wmcache.nld XML Document

Whenever a new removable disk is connected to the compromised system, the Agent.BTZ malware will infect the disk, copying itself to the root of the volume alone with an `autorun.inf` installer. It may also create a file named `thumb.db` in the root of the volume (though the purpose/content of the file is unknown).

The file metadata of this version contains a typographic error in the "FileDescription" field: "Java Virtual Mashine".

```
CompanyName:        Microsoft Corporation
FileDescription:    Java Virtual Mashine
FileVersion:        5.00.3805.0000
InternalName:       MSJAVAVM
LegalCopyright:     Copyright (C) Microsoft Corp. 1997-2000.
OriginalFilename:   MsJavaVM.dll
ProductName:        Microsoft(R) Windows (R) Operating System
ProductVersion:     5.00.3805.0000
```

Figure 7: Misspelled FileDescription Metadata

STRINGS (DECRYPTED)

```
USERNAME
PcName
Bias
Time
%02d:%02d:%04d %02d:%02d:%02d
Folder
File to send
Infecting level
Cure mode
Processing volumes
Http timeout
Http address
%s<add key="%s" value="%s" />
<TVer>2.1</TVer>
<AppendLog>0</AppendLog>
<?xml
%s<?xml version="1.0"
encoding="unicode"?>
<Cfg>
</Cfg>
%s<add key="%s" value="%s">
%s</add>
Java Virtual Mashine
OriginalFilename
MsJavaVM.dll
http://worldnews.ath.cx/update
/img0008/
index_20.jpg
\Internet Explorer\iexplore.exe
WriteProcessMemory
VirtualAllocEx
VirtualProtectEx
$1f.dll
Software\Microsoft\MSDACEng
Java.Runtime.52
\InprocServer32\
Software\Classes\CLSID\{1AEFA55F-60A6-
4817-B2D5-
12E2E48617F4}\InprocServer32\ThreadingM
odel
SOFTWARE\Microsoft\Windows\CurrentVersi
on\ShellServiceObjectDelayLoad\
{1AEFA55F-60A6-4817-B2D5-12E2E48617F4}
mstmdm.dll
Software\Microsoft\Windows\CurrentVersi
on\Run
Software\Microsoft\Windows\CurrentVersi
on\StrtdCfg
SYSTEM\CurrentControlSet\Control\CrashI
mage
wmcache.nld
Timout
IsActive
BSlp
SDCnt
LastValue
[autorun]
open=
shell\open=Explore
```

```
shell\open\Command=rundll32.exe
.\\%s,InstallM
shell\open\Default=1
rundll32.exe
autorun.inf
desktop.dll
thumb.db
thumb.dd
mssysmgr.ocx
mswmpdat.tlb
tapi32d.exe
typecli.exe
dswiz.dat
.dll
autorun
shell\open\command
rParam
dmcompos.dat
mfc42l00.pdb
winview.ocx
isuninst.bin
mswmpdat.tlb
wmcache.nld
%s\NativeList
Adding %s (%u)
~fgh
%s\%s\%s
Deleting %s (%u)
error (%u)
Adding %s to %s
Stop copying files.
Size of %s - %u
Copy file %s to %s (%u)
Processing command:
%s%%s08x.tmp
<FOUNDED>
%s\List
%08X is native.
wowmgr_is_loaded
f0fe
%s%s%s
Media arrived: "%c:"
Label:"%s"
FS:%s
SN:%08X
Apartment
UpdateCheck
zQWwe2esf34356d
IPlace
ILevelCount
CMValue
StVal
HtParam
Creating ps %s (%u)
%s%s\%s
%s\%s%s
Fails open %X\%s (%u) for writing
NTries
LastId
ISFValue
```

```
Installed
IListLen
IList
EmtParam
~DFBC
Native id failed (%u)
Native Id: %08X
%c%c%cMON%c%c
Set Value %s
Creating %s
ID: %s
Copying %s to %s (%u)
explorer.exe %s
%s "%s",Entry
Installing to %s
%s%s\%s%s
Deleted %s
DnsServerList:
DHCP Server:
SecWinsServer:
PriWinsServer:
GatewayList:
Mask:
%02X:%02X:%02X:%02X:%02X:%02X
Waiting %u (sec)
Timout: %u (sec); Passive interval: %u
(sec)
Queue %s
Achtung! Set to %s
Queue empty
Cndr: %u NoSl: %u IsInt: %u
No Int
Trying to delete %s
%s%s%s/
--D %s OK
%s%s%s%s%s
%I64u.jpg
According size not found
Not enough space
kernel32.dll
Sending %s to %s/%s
%I64u%02d%02d%02d%02d%02d.jpg
Windows NT %d.%d; SV1)
Win32
Size of %s is %u(B)
Finish run instruction.</CHCMD>
Exception in run instruction.
Del %s (%d)B ... OK
Error(%d) Del %s(%d)B
Run %s ... OK
Error(%d) run %s
write file %s (%dB)
Error(%d) write file %s
Error(%d) create file %s
Add address %s
Del record send file: %s
Find file (%dB)... OK
Error(%d) make file.
Send file %s 1 time
Add file to get: %s
Unknown instruction
Run cmd: %s
Set enable cure mode: %u
```

```
Clear NList
Clear List
Set enable expansion lvl: %u
Set expansion lvl: %u
Switch active mode time %d
Run instruction: %d
ID:%u%010u(%02d:%02d:%02d
%02d/%02d/%04d)
Error: pos(%d) > CmdSize(%d)
Cmd already done
Command Id:%u%010u(%02d:%02d:%02d
%02d/%02d/%04d)
Error: Can't detect del after
Del after %d
<CHCMD>
MakeFile Error(%d) del file %s
MakeFile Error(%d) copy file to temp
file %s
fuckouff
No console output
Error(%d) CreateProcess.
cmd /c %s
Error(%d), create %s.
057e-885f-ed33.tmp
/runas
advapi32.dll
FF.tmp
1dM3uu4j7Fw4sjnbcwlDqet4F7JyuUi4m5Imnxl
1pzxI6as80cbLnmz54cs5Ldn4ri3do5L6gs923H
L34x2f5cd0fk6c1a0s
Help
APPDATA
%s\1.txt
%s\Temp
%s\system32
AppendLog=%u
TVer=%s
%02d.%02d.%04d Log begin:
%s\system32\winview.ocx
release mutex - %u (%u)(%u)
waitResult: %u for %u
_hMutex: %u
Mutex_Log
Size of log(%dB) is too big, stop
write.
Log: Size of log(%dB) is too big, stop
write.
Log: Error(%d) get file size.
Error in slprintf copy bytes
%02d:%02d:%02d
Log end.
\system32\win.com
FA.tmp
%s\__1.dat
Fails open %X\%s (%u)
C:\WINDOWS\\system32\
C:\WINDOWS\\system32\mswmpdat.tlb
C:\WINDOWS\\system32\wmcache.nld
```