

Blackwood APT Group Has a New DLL Loader

Security News :: 1/29/2024



Overview

This week, the SonicWall Capture Labs threat research team analyzed a sample tied to the Blackwood APT group. This is a DLL that, when loaded onto a victim's computer, will escalate privileges and attempt to install a backdoor for communications monitoring and diversion. It has evasive capabilities and, as of this writing, is targeting companies and individuals in Japan and China.

Technical Overview

The sample is detected as a 32-bit DLL (Figure 1) with no packer or protector. It has minimal strings and no obvious obfuscation or encryption.

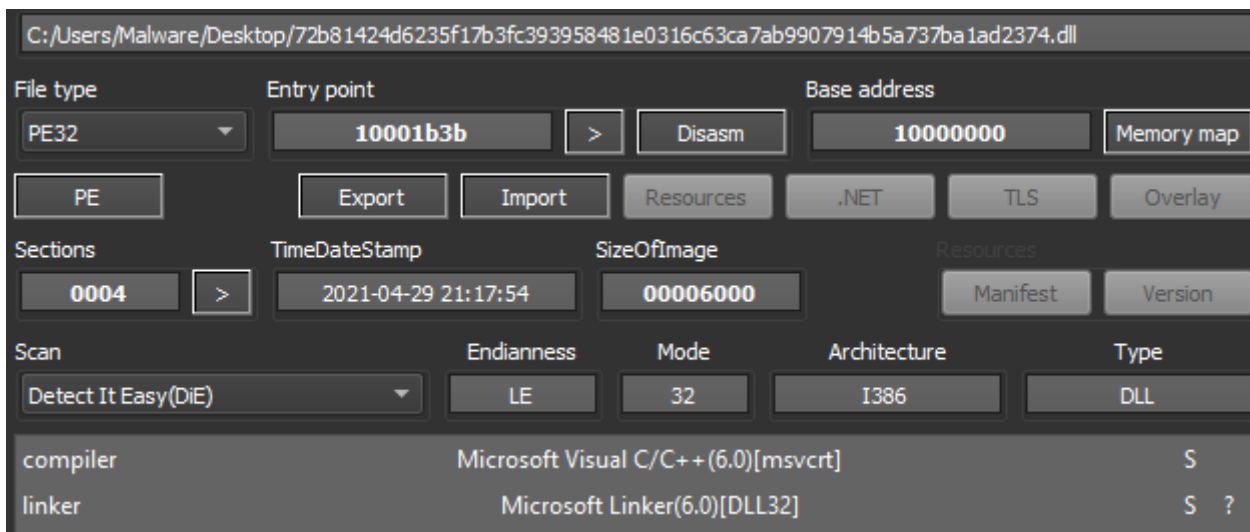


Figure 1: Sample detection

Strings show several API calls of concern, including GetCurrentProcessID, OpenProcess and VirtualAlloc – all of which are used to load malicious DLLs into memory. There are also two files listed: '3333333333333333.txt' and 'Update.ini', as shown in Figure 2.

blacklist (3)	hint (20)	value (164)
-	utility	SET
-	utility	Update
-	function	VirtualAllocEx
x	function	OpenProcess
x	function	GetCurrentProcessId
-	function	CoUninitialize
-	function	CoGetObject
-	function	CoInitialize
-	function	IIDFromString
-	function	initterm
-	function	_adjust_fdiv
-	function	_stricmp
-	format-string	D\$%s
-	file	KERNEL32.dll
-	file	ole32.dll
-	file	MSVCRT.dll
-	file	agent.dll
-	file	3333333333333333.txt
-	file	Update.ini

Figure 2: Static string detection

The name of the file is shown as 'agent.dll' (Figure 3) and there is one anonymous export that is only shown as an ordinal value when looking at the file with multiple tools.

indicator (31)	detail
strings > blacklist	count: 3
functions > blacklist	count: 3
checksum > invalid	expected: 0x0000D5B5
file > name > original	name: agent.dll
file > signature	name: Microsoft Visual C++ 6.0 DLL (Debug)
exports > functions	type: anonymous, count: 1

Figure 3: Original name and anonymous export

When dynamically analyzing the sample, it has multiple anti-analysis capabilities that prevent most of its function from being observed. It will look for debuggers, processor features and security settings in the registry (Figure 3). There are also locale checks that, when failed, will kill the process.

2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\1aff6089-e863-4d36-bdfd-3581f07440be
2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\1aff6089-e863-4d36-bdfd-3581f07440be
2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\c7e09e2a-c663-5399-af79-2fccc321d19a
2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\703fcc13-b66f-5868-ddd9-e2db7f381ffb
2:00:0...	DLLLoader32_...	6740	RegQueryKey	HKLM
2:00:0...	DLLLoader32_...	6740	RegQueryKey	HKLM
2:00:0...	DLLLoader32_...	6740	RegOpenKey	HKLM\Software\WOW6432Node\Microsoft\OLE\Tracing
2:00:0...	DLLLoader32_...	6740	RegOpenKey	HKLM\SOFTWARE\Microsoft\Ole\Tracing
2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\1aff6089-e863-4d36-bdfd-3581f07440be
2:00:0...	DLLLoader32_...	6740	RegQueryValue	HKLM\System\CurrentControlSet\Control\WMI\Security\1aff6089-e863-4d36-bdfd-3581f07440be

Figure 4: WMI registry keys being queried for security checks

The anonymous export at address 0x10001A70 is the file calling 'Rundll32.exe' for process injection, as shown in Figure 5.

```

00011990 81EC 14010000 sub esp,114
00011996 . 57 push edi
00011997 . 33D2 xor edx,edx
00011999 . B9 40000000 mov ecx,40
0001199E . 33C0 xor eax,eax
000119A0 . 8D7C24 15 lea edi,dword ptr ss:[esp+15]
000119A4 . 885424 14 mov byte ptr ss:[esp+14],d1
000119A8 . F3:AB rep stosd
000119AA . 66:AB stosw
000119AC . AA stosb
000119AD . B0 6C mov al,6C
000119AF . 68 04010000 push 104
000119B4 . 884424 0C mov byte ptr ss:[esp+C],al
000119B8 . 884424 0D mov byte ptr ss:[esp+D],al
000119BC . B0 65 mov al,65
000119BE . C64424 08 72 mov byte ptr ss:[esp+8],72
000119C3 . 884424 11 mov byte ptr ss:[esp+11],al
000119C7 . 884424 13 mov byte ptr ss:[esp+13],al
000119CB . 8D4424 18 lea eax,dword ptr ss:[esp+18]
000119CF . C64424 09 75 mov byte ptr ss:[esp+9],75
000119D4 . 50 push eax
000119D5 . 52 push edx
000119D6 . C64424 12 6E mov byte ptr ss:[esp+12],6E
000119DB . C64424 13 64 mov byte ptr ss:[esp+13],64
000119E0 . C64424 16 33 mov byte ptr ss:[esp+16],33
000119E5 . C64424 17 32 mov byte ptr ss:[esp+17],32
000119EA . C64424 18 2E mov byte ptr ss:[esp+18],2E
000119EF . C64424 1A 78 mov byte ptr ss:[esp+1A],78
000119F4 . 885424 1C mov byte ptr ss:[esp+1C],d1
000119F8 . FF15 1C200010 call dword ptr ds:[<&GetModuleFileNameA]
sub_10001990 calls RunDLL32.exe
edi:EntryPoint
edx:"MZ蠕"
ecx:EntryPoint, 40:'@'
edi:EntryPoint
6C:'l'
65:'e'
72:'r'
[esp+18]: "MZ蠕"
75:'u'
edx:"MZ蠕"
6E:'n'
64:'d'
33:'3'
32:'2'
2E: '.'
78:'x'

```

Figure 5: Export address calls sub_10001990, which creates 'rundll32.exe'

Controlling the program's execution allows the check for a UAC bypass to be generated. The DLL will attempt to escalate privileges via CMSTPLUA interface^[1]. The following strings are created, as shown in Figures 5 and 6:

- Elevation:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}
- Elevation:Administrator!new:{F885120E-3789-4FD9-865E-DC9B4A6412D2}

```

0001147A . 8B 24000000 mov ebx,0
0001147C . 8D 41000000 mov ebp,41
0001147E . 50 push eax
00011480 . 51 push ecx
00011482 . C74424 18 00000000 mov dword ptr ss:[esp+18],0
00011484 . 66:C74424 1C 7800 mov word ptr ss:[esp+1C],78
00011486 . 66:895C24 20 mov word ptr ss:[esp+20],bx
00011488 . 66:895C24 22 mov word ptr ss:[esp+22],bx
0001148A . 66:C74424 24 3500 mov word ptr ss:[esp+24],35
0001148C . 66:C74424 26 3100 mov word ptr ss:[esp+26],31
0001148E . 66:C74424 2A 3000 mov word ptr ss:[esp+2A],30
00011490 . 66:C74424 2C 4500 mov word ptr ss:[esp+2C],45
00011492 . 66:897424 2E mov word ptr ss:[esp+2E],s1
00011494 . 66:C74424 30 3300 mov word ptr ss:[esp+30],33
00011496 . 66:897C24 32 mov word ptr ss:[esp+32],d1
00011498 . 66:895C24 34 mov word ptr ss:[esp+34],bx
0001149A . 66:897424 38 mov word ptr ss:[esp+38],s1
0001149C . 66:895424 3A mov word ptr ss:[esp+3A],dx
0001149E . 66:897424 42 mov word ptr ss:[esp+42],s1
000114A0 . 66:895C24 44 mov word ptr ss:[esp+44],bx
000114A2 . 66:C74424 46 3600 mov word ptr ss:[esp+46],36
000114A4 . 66:C74424 48 3500 mov word ptr ss:[esp+48],35
000114A6 . 66:C74424 4A 4500 mov word ptr ss:[esp+4A],45
000114A8 . 66:897424 4C mov word ptr ss:[esp+4C],s1
000114AA . 66:C74424 50 4300 mov word ptr ss:[esp+50],43
000114AC . 66:C74424 54 4200 mov word ptr ss:[esp+54],42
000114AE . 66:895424 56 mov word ptr ss:[esp+56],dx
000114B0 . 66:896C24 58 mov word ptr ss:[esp+58],bp
000114B2 . 66:C74424 5A 3600 mov word ptr ss:[esp+5A],36
000114B4 . 66:895424 5C mov word ptr ss:[esp+5C],dx
000114B6 . 66:C74424 5E 3100 mov word ptr ss:[esp+5E],31
000114B8 . 66:C74424 62 4400 mov word ptr ss:[esp+62],44
000114BA . 66:C74424 66 7D00 mov word ptr ss:[esp+66],7D
000114BC . 66:C74424 68 0000 mov word ptr ss:[esp+68],0
000114DE . FF15 54200010 call dword ptr ds:[<&IIDFromString>]
34:'A'
41:'A'
ecx:L"{F885120E-3789-4FD9-865E-DC9B4A6412D2}"
78:'{'
35:'S'
31:'1'
30:'0'
45:'E'
33:'3'
36:'6'
35:'S'
45:'E'
43:'C'
42:'B'
36:'6'
31:'1'
44:'D'
7D:'})'

```

[1] <https://gist.github.com/hfiref0x/196af729106b780db1c73428b5a5d68d>

```

00011785 . 898424 5C010000 mov dword ptr ss:[esp+15C],eax
00011787 . 8D5424 10 lea edx,dword ptr ss:[esp+10]
00011789 . 8D8424 60010000 lea eax,dword ptr ss:[esp+160]
0001178B . 52 push edx
0001178D . 8D8C24 40010000 lea ecx,dword ptr ss:[esp+140]
0001178F . 50 push eax
00011791 . 8D9424 8C000000 lea edx,dword ptr ss:[esp+8C]
00011793 . 51 push ecx
00011795 . 52 push edx
00011797 . C78424 4C010000 2400 mov dword ptr ss:[esp+14C],24
00011799 . C78424 60010000 0400 mov dword ptr ss:[esp+160],4
0001179B . FF15 4C200010 call dword ptr ds:[<&CoGetObject>]
0001179D . 85C0 test eax,eax
0001179F . 0F8C 89010000 jnz agent.1000197F
000117A1 . 8B424 10 mov ecx,dword ptr ss:[esp+10]
000117A3 . 8B11 mov edx,dword ptr ds:[ecx]
000117A5 . 8B42 0C mov eax,dword ptr ds:[edx+C]
000117A7 . 8B7A 08 mov edi,dword ptr ds:[edx+8]
000117A9 . 85C0 test eax,eax
edx:L"Elevation:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}"
[esp+140]: "etAvaiTableNetworkList"
24:'5'
edx:L"Elevation:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}"
24:'5'
edx:L"Elevation:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}"
edx+C:L"ion:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}"
edx+8:L"ation:Administrator!new:{FCC74B77-EC3E-4DD8-A80B-008A702075A9}"

```

Figures 6 (top) and 7 (bottom): A function creates GUIDs for privilege escalation

The two files that are listed within the strings are also referenced during runtime (Figure 7), but despite multiple attempts at controlling execution, the files were not observed on test systems.

```
000121C . 66:AB stosw
000121E . AA stosb
000121F . 8D4424 18 lea eax,dword ptr ss:[esp+18]
0001223 . 8D8C24 1C010000 lea ecx,dword ptr ss:[esp+11C]
000122A . 50 push eax
000122B . 68 04010000 push 104
0001230 . 51 push ecx
0001231 . 68 88420010 push agent.10004288
0001236 . 68 70410010 push agent.10004170
000123B . 68 6C410010 push agent.1000416C
0001240 . FF15 18200010 call dword ptr ds:[<&GetPrivateProfileS]
0001246 . 8B35 14200010 mov esi,dword ptr ds:[<&DeleteFileA>]
000124C . 85C0 test eax,eax
000124E . 76 15 jbe agent.10001265
0001250 . 68 E8030000 push 3E8
```

eax: "C:\\Users\\Malware\\Desktop\\Update.ini"
10004170: "Update"
1000416C: "SET"

eax: "C:\\Users\\Malware\\Desktop\\Update.ini"

Figure 8: Update.ini is referenced but never created

Protection

To ensure SonicWall customers are prepared for any exposure that may occur due to this malware, the following signatures have been released:

- MalAgent.Blackwood

IOCs

- 72B81424D6235F17B3FC393958481E0316C63CA7AB9907914B5A737BA1AD2374