Demystifying Qbot Malware

Stories

The latest cybersecurity trends, best practices, security vulnerabilities, and more

By Adithya Chandra and Sushant Kumar Arya · August 24, 2022

Executive summary

The Trellix SecOps Team has observed an uptick in the Qbot malware infections in recent months. Qbot has been an active threat for over 14 years and continues to evolve, adopting new infection vectors to evade detection mechanisms.

At Trellix, we are committed to protecting our customers from upcoming and emerging threats on the their network, inclusive of those that are found being exploited in the wild. Trellix SecOps strives to build advanced detection features, improving product's overall Threat Detection capabilities. Over the next few sections of this blog, we will highlight TTPs of Qbot malware, detection capabilities in Trellix products. and some detection strategies which help protect customers against this and future attacks of similar nature.

Introduction

Qbot, also known as QakBot, QuackBot and Pinkslipbot, is a common trojan malware designed to steal passwords. Over time this malware has evolved from simple infostealer malware to an infostealer with a backdoor functionality. The malware has been active since 2008 and is primarily used by financially motivated actors. Qbot actors have also served as 'Initial Access Brokers' to many ransomware partners including REvil/Sodinokibi. In the latest versions of Qbot Payload, we have observed significant changes in its TTPs, including new delivery vectors to evade detection mechanisms.

In 2022-Q1&Q2, Trellix has detected ~500K Qbot URLs in FAUDE (FireEye Advanced URL Detection Engine) with a peak of 189,313 URLs in Apr-2022.

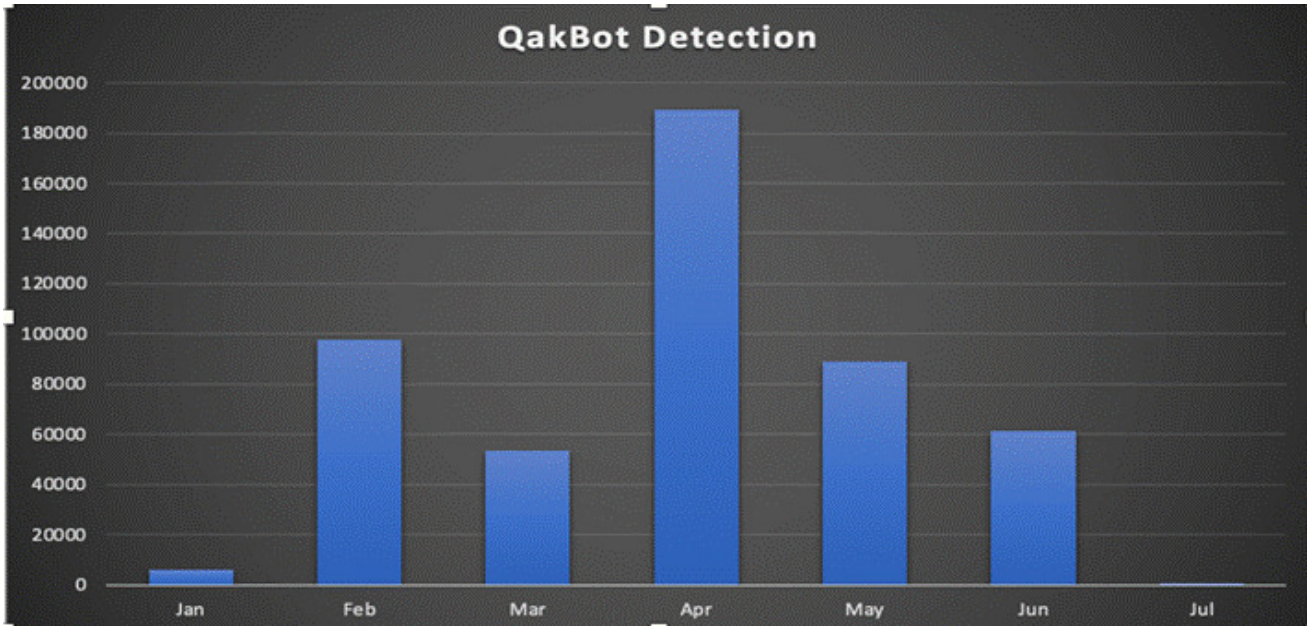


Figure 1: Qbot infection trend over Q1-Q2

Trellix has been studying this malware and discovered a significant uptick in the spread of Qbot malware over the first half of 2022 using several new techniques. We put together a comprehensive analysis detailing its TTPs, IOCs, Detection & Hunting Schemas and defence mechanisms from Trellix products.

Qbot threat landscape

The Qbot threat landscape with reference to the geopolitical regions and industry verticals has changed from time to time and we have compiled regions and verticals targeted as shown in below section:

Figure 2: QbBot threat landscape summary

Infection vector: email

Initially Qbot was distributed by Emotet malware, but currently the major infection vector is malspam email campaigns with multiple variants. Over the next section, we discuss the prevalent variants across customers.

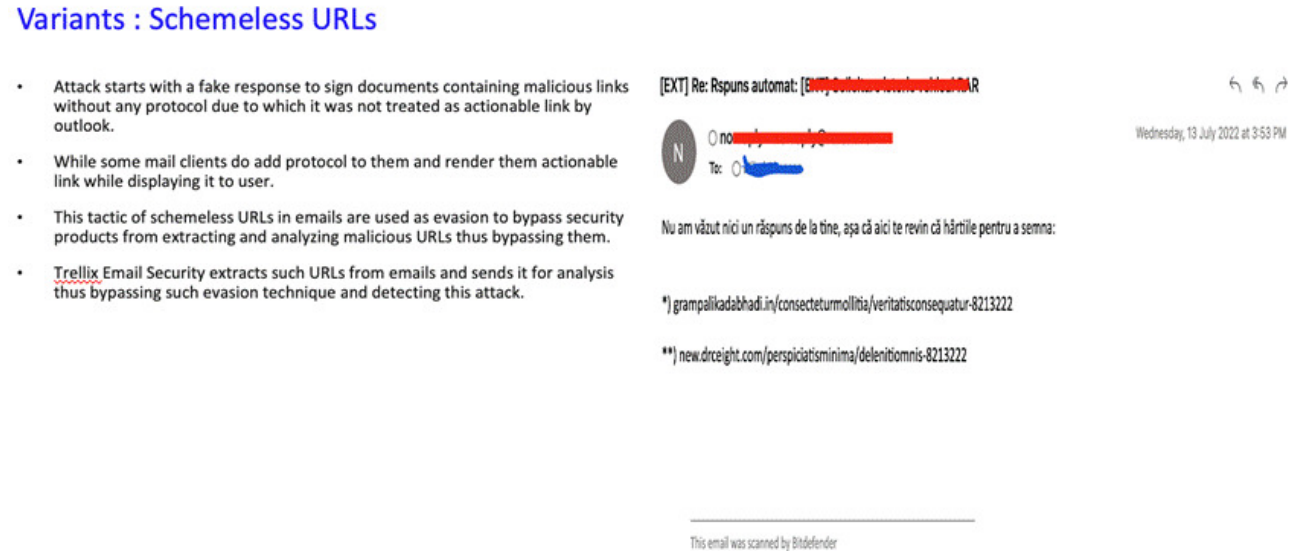


Figure 3: Qbot Email (Variant: Schemeless URLs)

Variants : Malicious URLs

- Attack starts with a fake voice mail with a button to play voice message.
- Button to play voice message embeds malicious URL to download zip containing malicious doc/xls.
- Trellix Email Security follows the attack chain from Email -> URL -> Zip -> XLS/DOC
- Trellix Email Security extracts and downloads the final payload and execute it in sandbox and detects it.

Figure 4: Qbot Email (Variant: Malicious URLs)

Variants : Encrypted malicious attachment

- Attack starts with a hijacked email from a legit conversation containing malicious encrypted zip/xls as attachment.
- XLS contains macro to download Qbot payload and execute them on opening.
- Trellix Email Security extracts zip and decrypts malicious XLS using the password provided in body.
- Trellix Email Security detects this malicious attack based on the dynamic behavior of the executed XLS file.

Figure 5: Qbot Email (Variant: Encrypted Malicious Attachment, Thread Hijack)

Variants : Malicious HTML attachment

- Attack starts with an email containing malicious html attachment.
- This html once opened automatically drops zip containing malicious xls attachment.
- Trellix Email Security extracts html file and sends it for analysis and thus detects this attack.

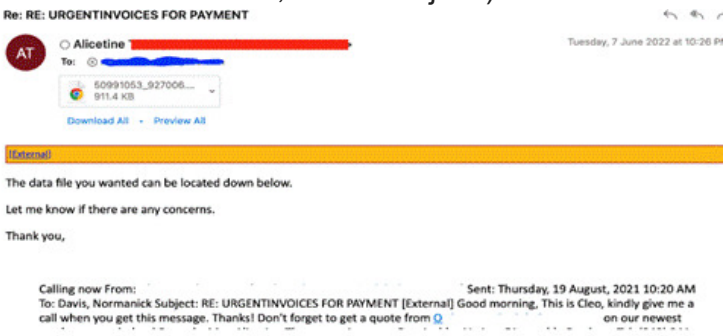


Figure 6: Qbot Email (Variant: Malicious HTML Attachment)

MITRE mappings

Attack Behaviour	Example	MITRE ATT&CK
Malicious email with an *.html* attachment.	Threat actor lured user to open malicious email with malicious *.html* attachment.	Initial Access / Phishing (T1566)
The *.html* file opens in a browser and uses *HTML Smuggling* to drop an embedded *.ZIP* file to the hard drive.	User manually click the HTML file from downloads directory. Process: Chrome.exe Commandline: --single-argument *.html Path:*\\User\\Downloads*	Defense Evasion / Obfuscated Files or Information: HTML Smuggling (T1027.006)
Password-protected *zipped file which contains an .ISO image.*	User unzips the password protected zip file which contain an ISO file.	Defense Evasion / Subvert Trust Controls: Mark-of-the-Web Bypass (T1553.005)
User executed malicious Windows Shortcut, which executes calc.exe from mounted ISO image.	User clicks the malicious *LNK file* from the *ISO file*. As rest of the items will be hidden, only lnk file will be visible to the user.	Execution / User Execution (T1204.002)
calc.exe loads adversary crafted WindowsCodecs DLL.	*Process: Calc.exe* *Sysmon event id: 7* *ImageLoaded: C:\Users\User\Downloads\..\WindowsCodecs.dll*	Defense Evasion / Hijack Execution Flow: DLL Side-Loading (T1574.002)
Calc.exe spawns Microsoft signed binary (*RegSvr32.exe*) to executes *Qbot dll (loader)*	*ParentProcess: Calc.exe* *Process: Regsvr32.exe*	Defense Evasion / System Binary Proxy Execution (T1218)
RegSvr32.exe(Qbot loader dll) spawns and *injects Explorer. (Recent versions has seen injecting to explorer.exe, wermgr.exe, msra.exe etc)*	*ParentProcess: Regsvr32.exe Process: Explorer.exe/ wermgr.exe*	Defense Evasion / Process Injection (T1055)
Explorer creates scheduled task	*ParentProcess: Explorer.exe* *Process:schtasks.exe*	Persistence/Scheduled Task/Job: Scheduled Task

Explorer creates new registry entries	*Symon event id: 13* *Event Action: Registry Value Set*	Defense Evasion/Modify Regsitry
Explorer connects with C2	*Sysmon event id: 3* *Process: Explorer.exe*	System Binary Proxy Execution/ Command & Control
Explorer executes a well-known sequence of Qbot discovery commands.	*Explorer.exe* spawns *whoami, arp, ipconfig, net view, cmd, nslookup, nltest, net share, route, netstat, net localgroup, qwinsta and other discovery activities via WMI queries.*	Discovery / System Information Discovery (T1082)

Table 1: MITRE mappings

Breaking down the Qbot malware

The most prevalent way Qbot infects its victims is via email. The emails used in the latest campaign carry an HTML file (TXRTN_2636021.html). The user downloads the HTML attachment and opens it in their browser which downloads a password-protected ZIP archive (TXRTN_2636021.zip) with an ISO file inside.

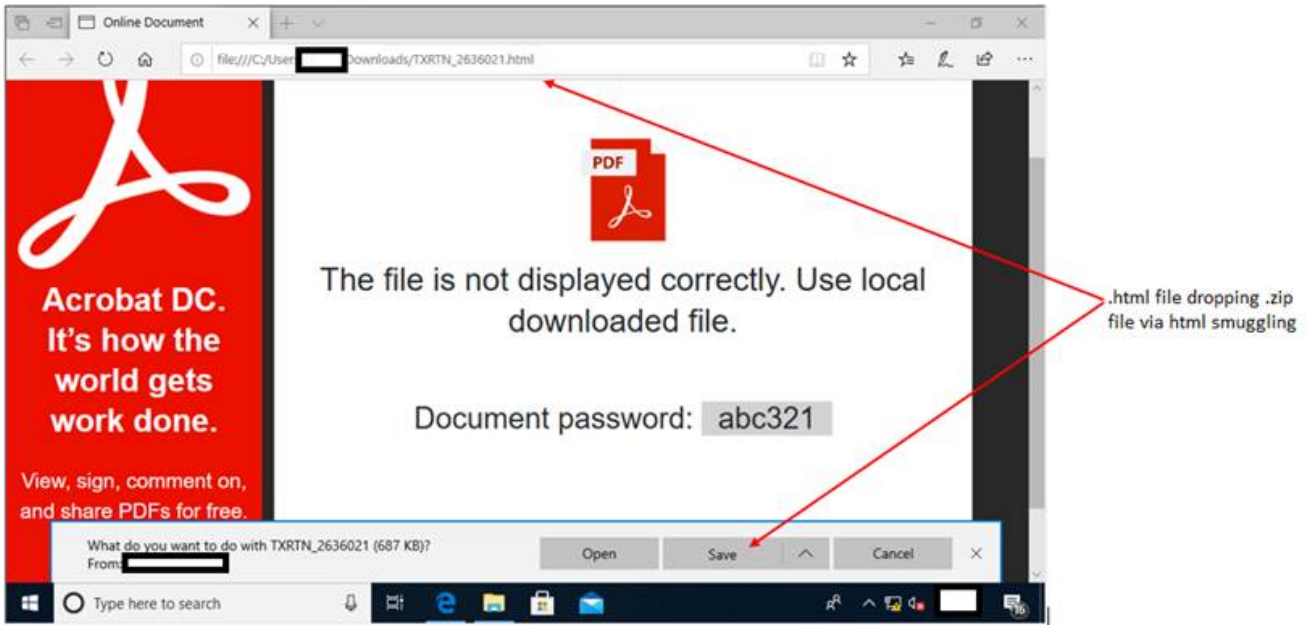


Figure 7: The html file renders in the user browser showing the password to open the dropped zip file

HTML smuggling: highly evasive malware delivery technique

(MITRE: Defense Evasion/Obfuscated Files or Information: HTML Smuggling (T1027.006))

HTML smuggling is an evasive payload delivery method that helps an attacker smuggle a payload past content filters and firewalls by hiding malicious payloads inside of seemingly benign HTML files. On opening the html file in vscode/ notepad ++ we can easily see how this

is being done. There is a very long base64encoded variable that is present in the javascript section of this HTML file. The javascript assembles the zip package and drops it to the device.

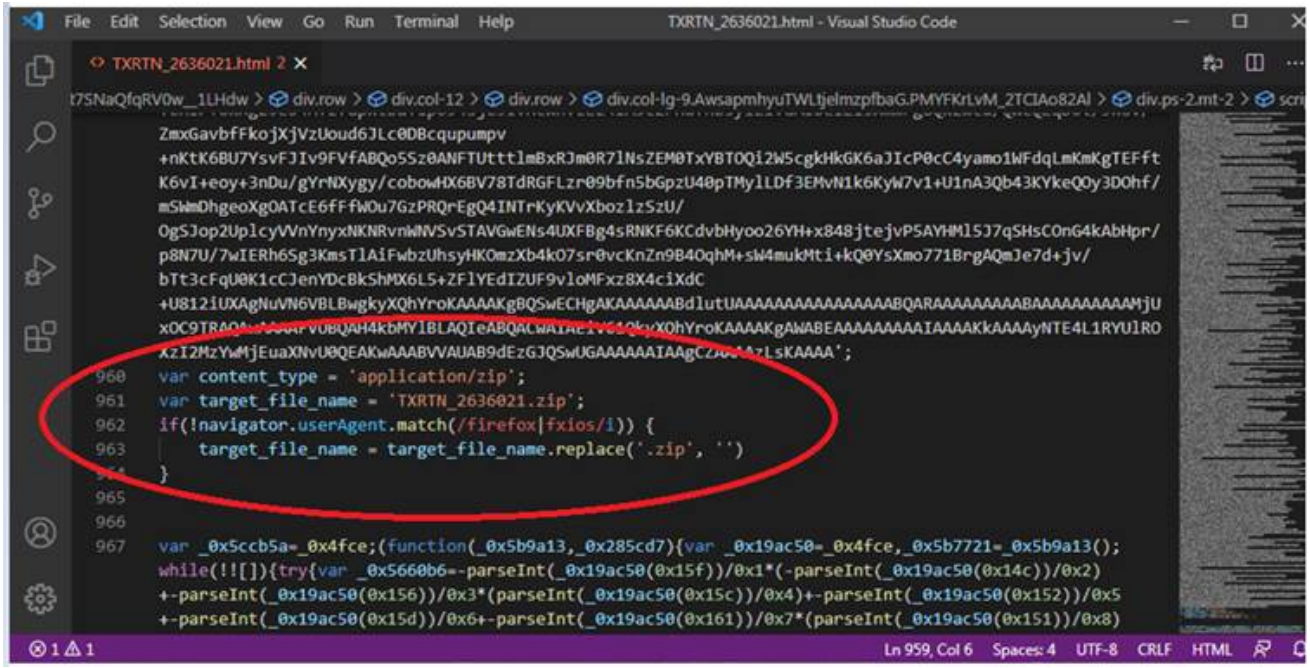


Figure 8: HTML content showing the smuggled zip payload

On Unzipping the ZIP file with the password shown the in html document, user gets the ISO file.

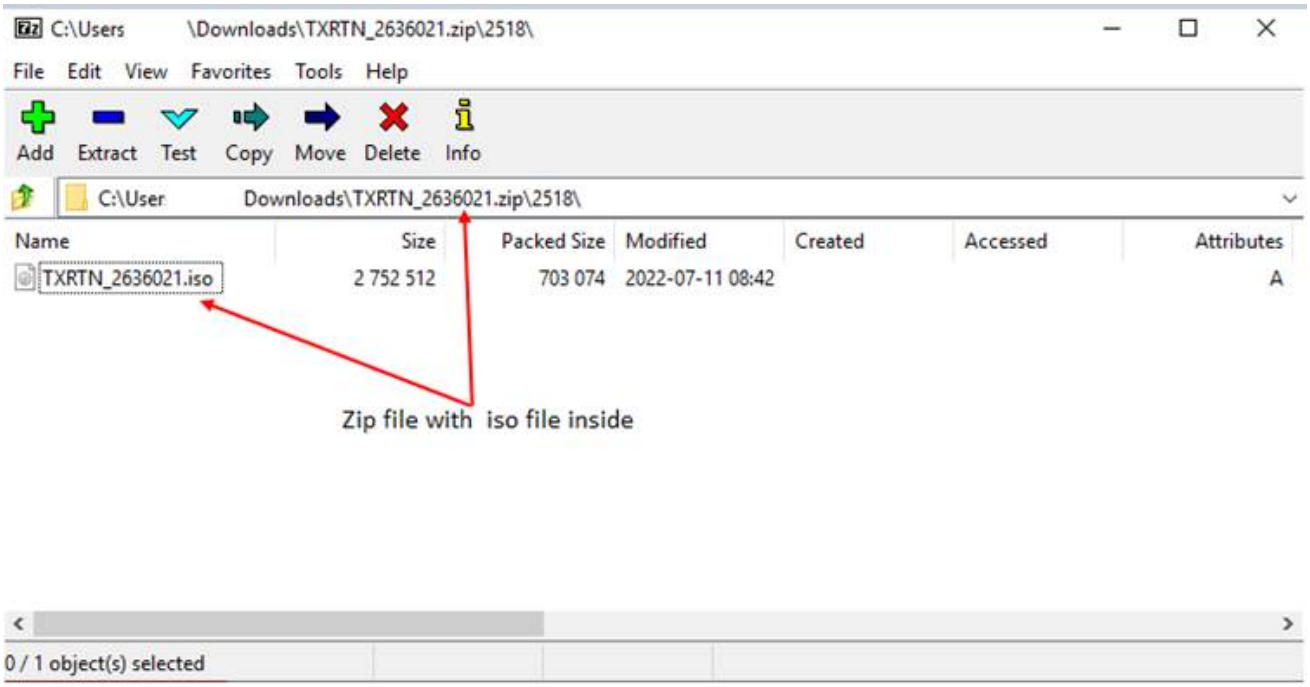


Figure 9: 7zip snip showing the zip file and iso

On mounting the ISO, user sees only the ‘LNK’(Shortcut) file; the rest of which are hidden.

(MITRE: Defense Evasion/Subvert Trust Controls: Mark-of-the-Web Bypass (T1553.005))

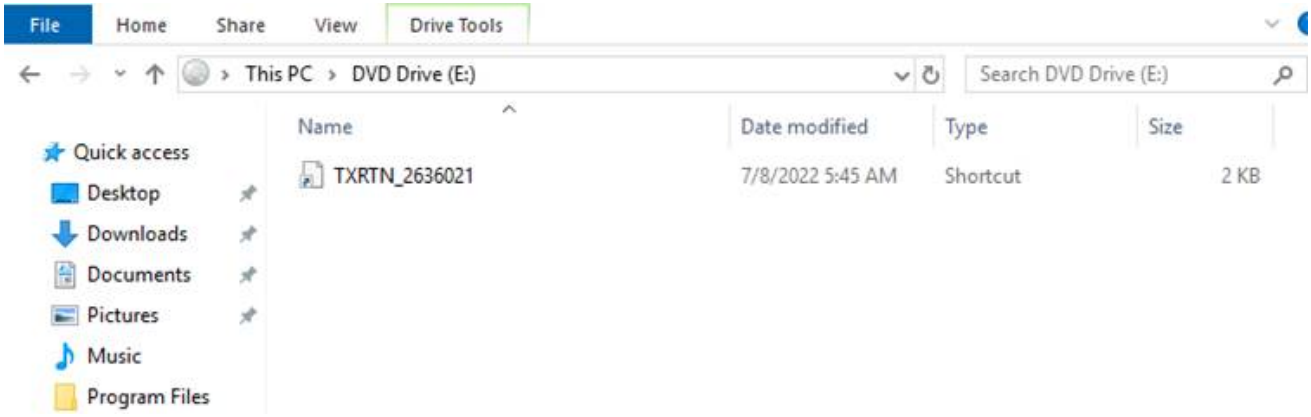


Figure 10: Lnk File shown to User

On unpacking this ISO, there are 4 files inside of it as shown below. The ISO contains a .LNK file(shortcut), *calc.exe* (Windows Calculator), and two DLL files, namely WindowsCodecs.dll and Qbot payload named 102755.*dll*

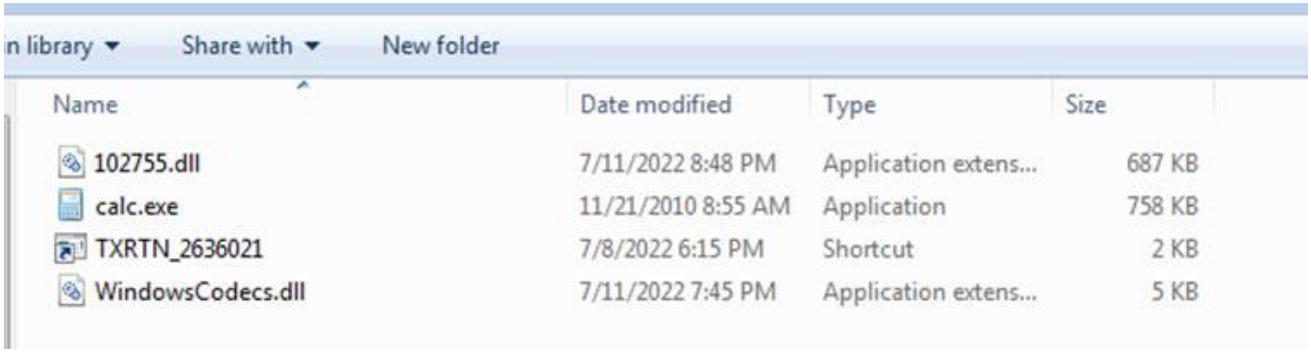
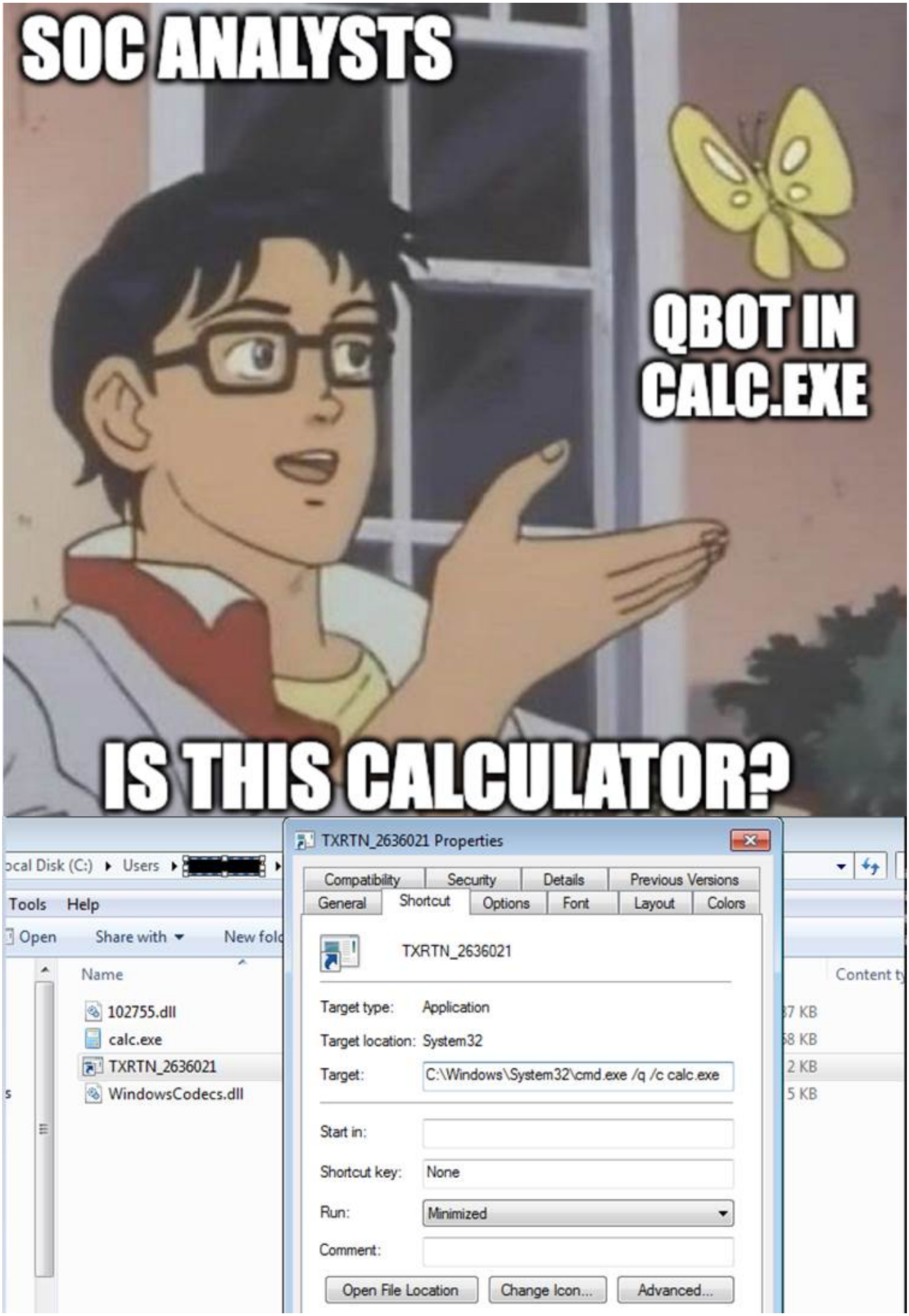


Figure 11: ISO contents showing dll's, lnk file and calc.exe

The user clicking the shortcut file triggers the Qbot malware infection by executing the 'calc.exe' through the Command Prompt.


SOC ANALYSTS
QBOT IN CALC.EXE
IS THIS CALCULATOR?
TXRTN_2636021 Properties
Compatibility
Security
Details
Previous Versions
General
Shortcut
Options
Font
Layout
Colors
TXRTN_2636021
Target type: Application
Target location: System32
Target: C:\Windows\System32\cmd.exe /q /c calc.exe
Start in:
Shortcut key: None
Run: Minimized
Comment:
Open File Location
Change Icon...
Advanced...
Tools
Help
Open
Share with
Name
102755.dll
calc.exe
TXRTN_2636021
WindowsCodecs.dll

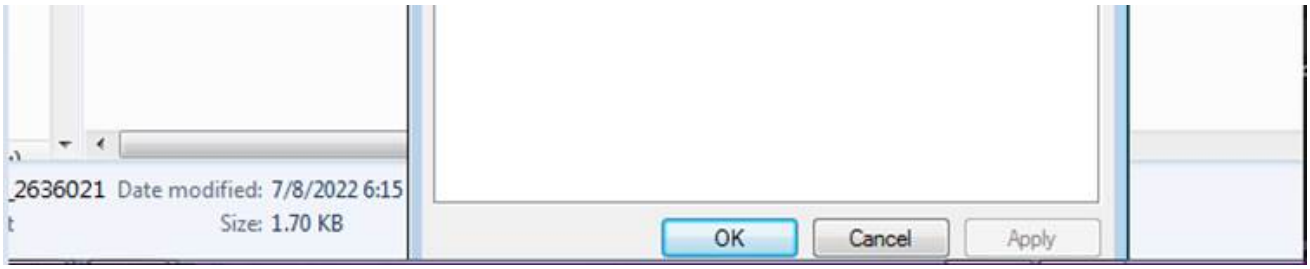


Figure 12: The lnk file properties shows the target with which it is associated

Why a Windows 7 calculator?

DLL Sideloading:

(MITRE: Defense Evasion/Hijack Execution Flow: DLL Side-Loading (T1574.002))

Windows allows applications to load DLLs at runtime. Applications can specify the location of DLLs to load by specifying a full path, using DLL redirection, or by using an application manifest. If none of these methods are used, it attempts to locate the DLL by searching a predefined set of directories in a set order.

When the shortcut loads, the Windows 7 Calculator, it automatically searches for and attempts to load the legitimate WindowsCodecs DLL file. However, it does not check for the DLL in certain hard coded paths and will load any DLL file with the same name if placed in the same folder as the Calc.exe executable.

In the below snip, you can see that Calc.exe was placed in the desktop path and executed. Once it starts execution, it searches and loads WindowsCodecs DLL. However, the DLL was not present in the desktop path, hence it shows in 'Result' tab as "NAME NOT FOUND".

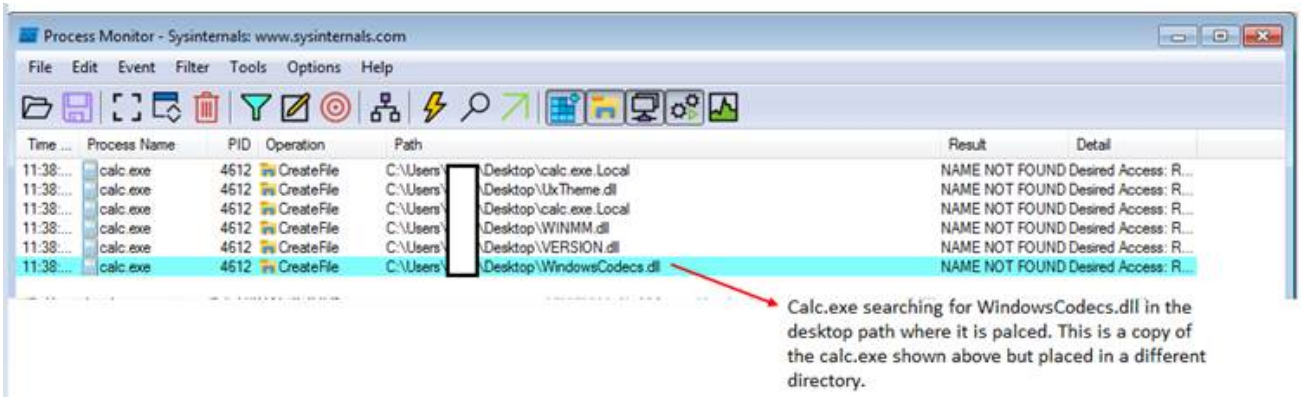


Figure 13: Calc.exe sideloading windowscodecs.dll

This doesn't work anymore with the Windows 10 version of the calculator application. Hence the threat actors behind Qbot, bundled Windows 7 calc.exe along with this.

Now What?

The attacker takes advantage of this flaw by creating their own malicious WindowsCodecs.dll file that launches the other dll file, which is nothing but the QuakBot malware(loader module). Thus, by launching Qbot through a trusted program like the Windows Calculator, attackers can easily fool security analysts and some security software.

How is it so stealthy?

The bundled calculator app is a legit benign Windows 7 calculator. But with the aforementioned flaw, the calculator loads the ‘windowscodecs.dll’ from the same directory, which is specifically crafted by the adversary. On executing the calculator, it loads windowscodecs.dll which in turn loads the Qbot loader payload via regsvr32.

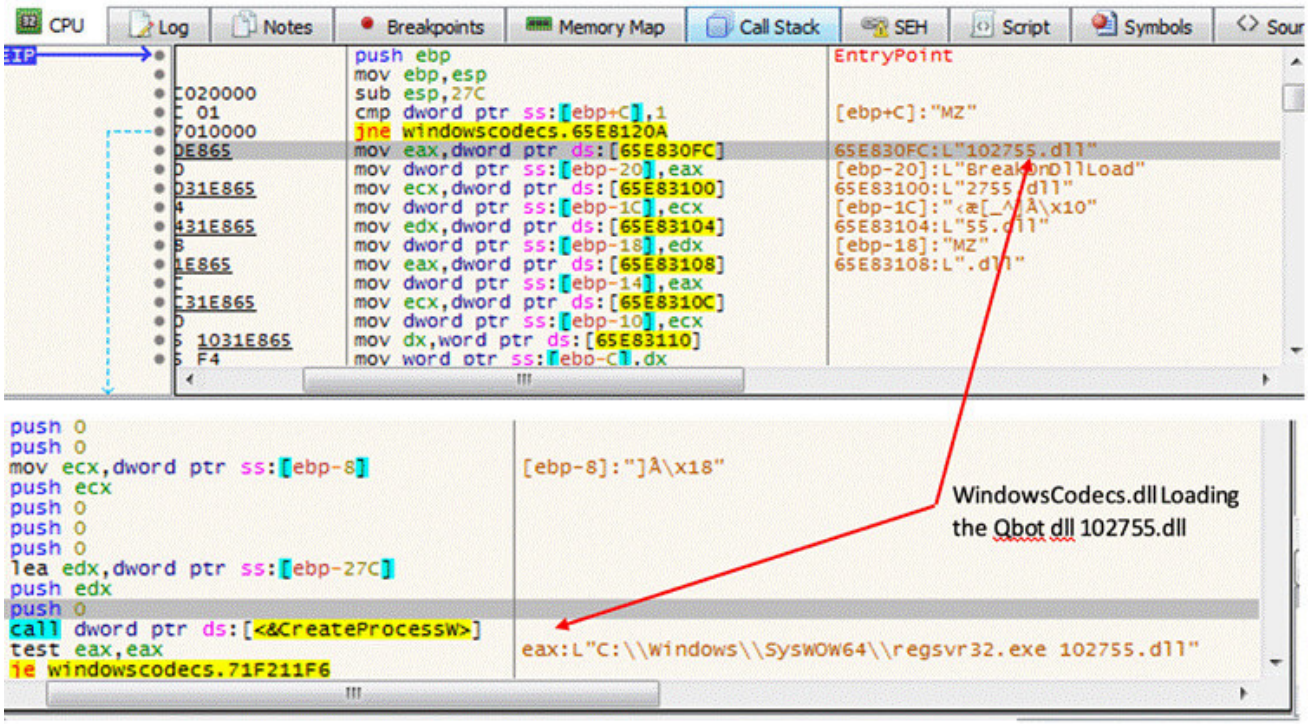


Figure 14: Windows codecs leveraging regsvr32 via CreateProcessW to load the Qbot loader DLL

This Qbot loader DLL is a x32 bit Delphi compiled binary, with no export functions.

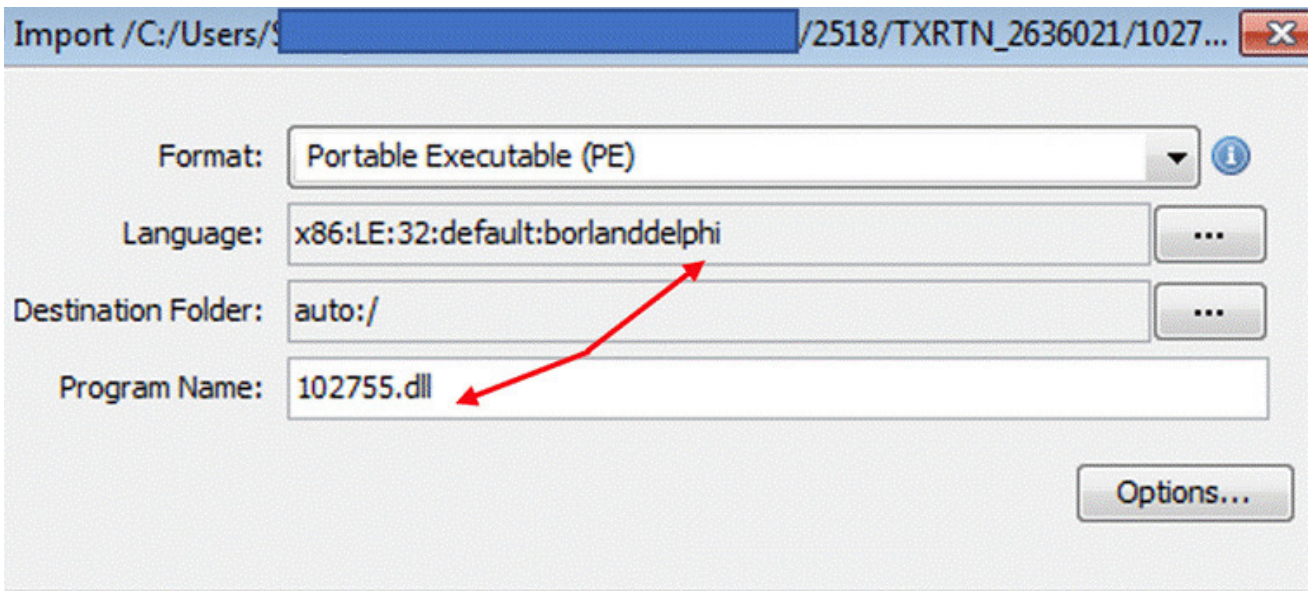


Figure 15: Qbot loader DLL (Delphi compiled 32bit binary)

But as it is executed and gets unpacked, the core Qbot payload is a VC (C/C++) compiled binary (DLL) with export functions.

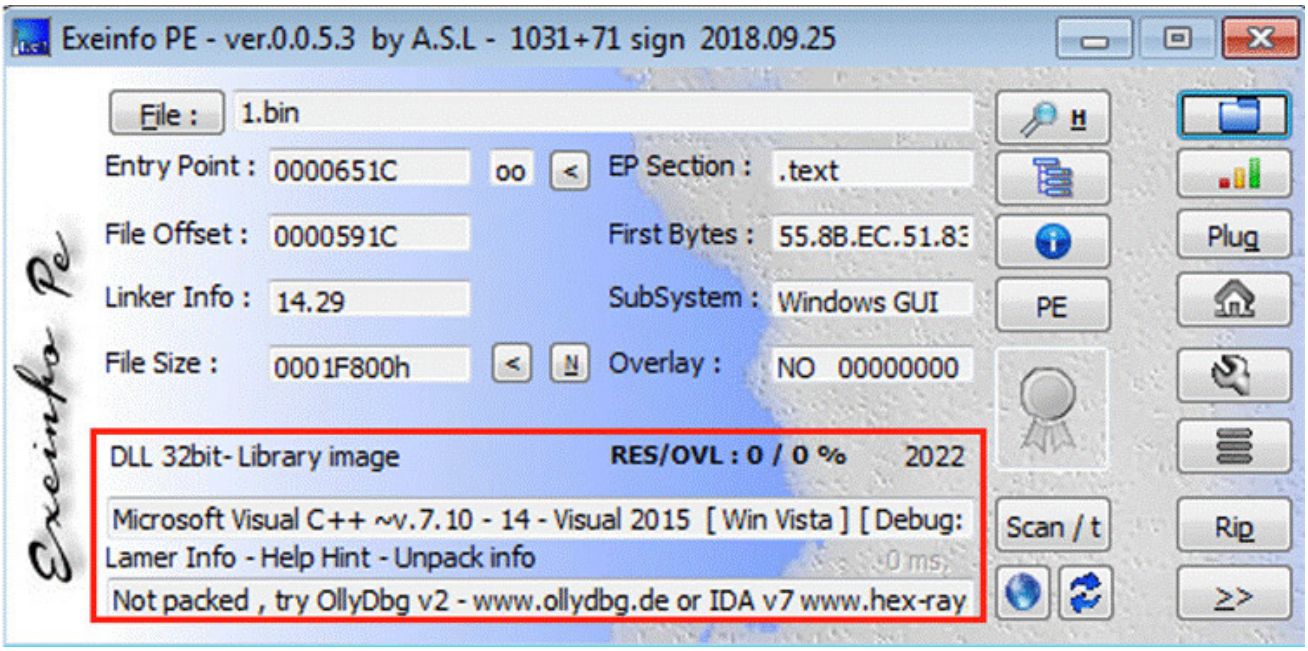


Figure 16: Qbot core module (C/C++ compiled 32bit binary)

On analyzing this core module further, it does the same checks as seen in previous versions of Qbot payloads.

Below shows the malware payload checking for Windows Defender Emulation using WinAPI GetFileAttributes “C:\INTERNAL__empty”.

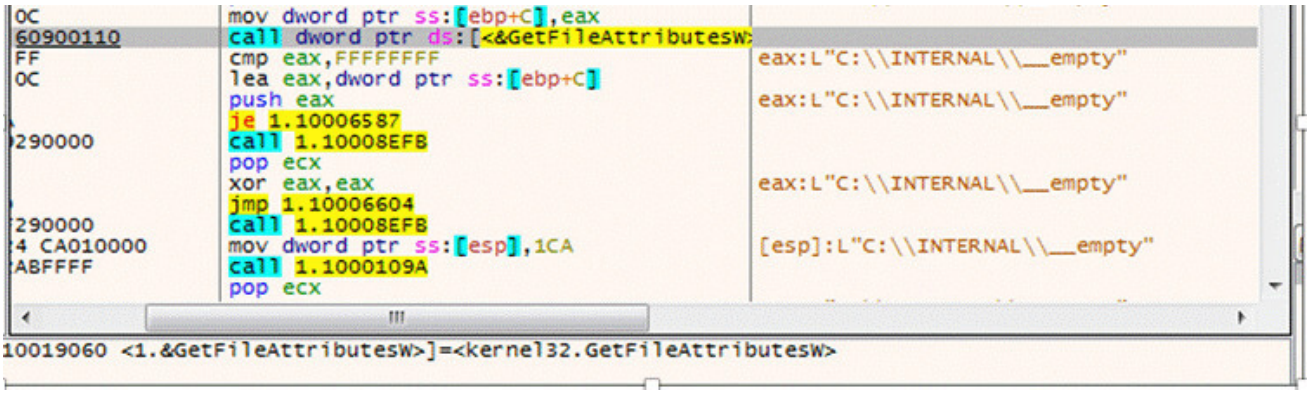


Figure 17: Qbot checking GetFileAttributesW

The payload also uses flag SELF_TEST_1 which appears to be for self-debugging purposes to check if the machine is infected or not.

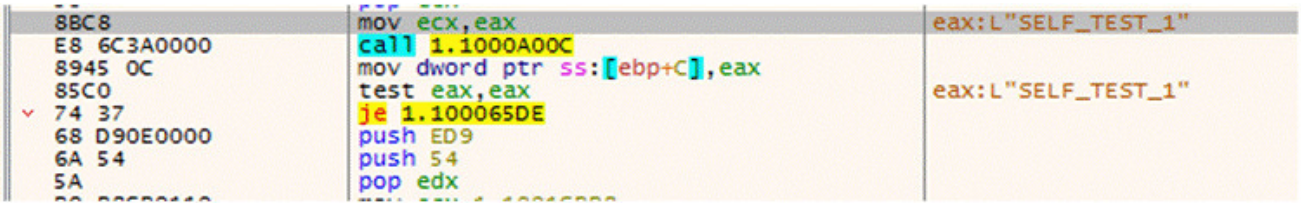


Figure 18: Qbot self-check using SELF_TEST_1 flag

After the self-check debugging test, the malware creates a new thread and starts its execution.

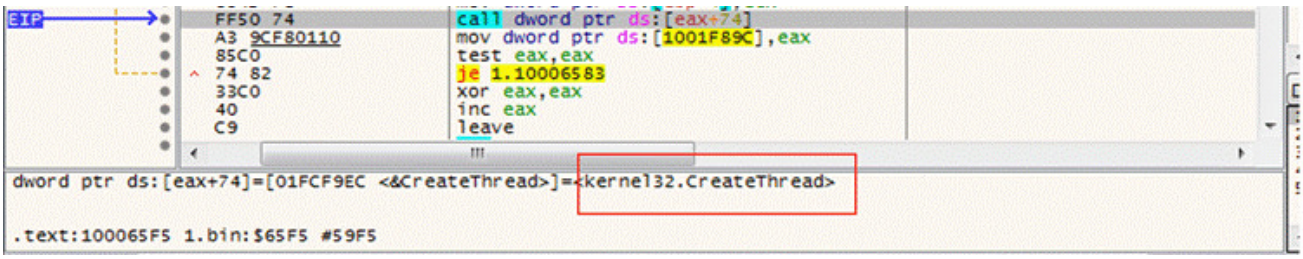


Figure 19: Qbot creating remote thread

Later, this payload module loads/imports DLLs, enumerating the system process along with anti-debug checks and leading the payload to inject to explorer.exe/wermgr.exe via process hollowing.

How does it gain persistence?

As Qbot has evolved, it continues to rely on the use of scheduled tasks and registry keys/runkeys for its persistence with variations on its implementation.

As the core payload gets executed, the Qbot gains its persistence via 3 steps:

1. Copying itself to below mentioned folder: %AppData%\Roaming\Microsoft\{RandomStrings}
2. Creating a registry value pointing to the above payload
3. Scheduled task to launch the loader 102755.dll

Folder Creations:

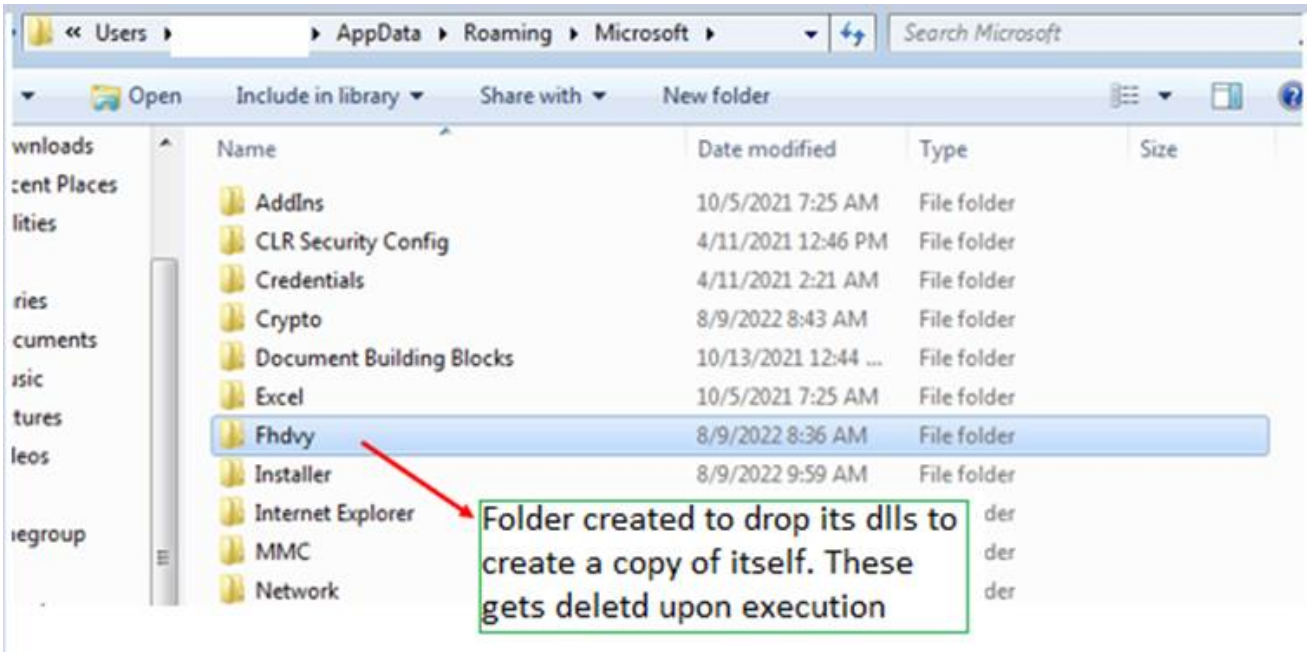


Figure 20: Folder created in AppData Roaming, dropping DLL's

Dropped DLLs are loaded via regsvr32.exe as shown in below:

Figure 21: Executing Dropped DLLs via RegSvr32

Registry entries:

(MITRE: Persistence/Defense Evasion/Modify Regsitry)

In the latest payload versions, Qbot has moved from creating its config file in ".dat" format. Now, it writes its cloned dDLL entry in the victim host along with its botnet/campaign ID as encrypted registry keys to the 'HKCU\Software\Microsoft\' Hive.

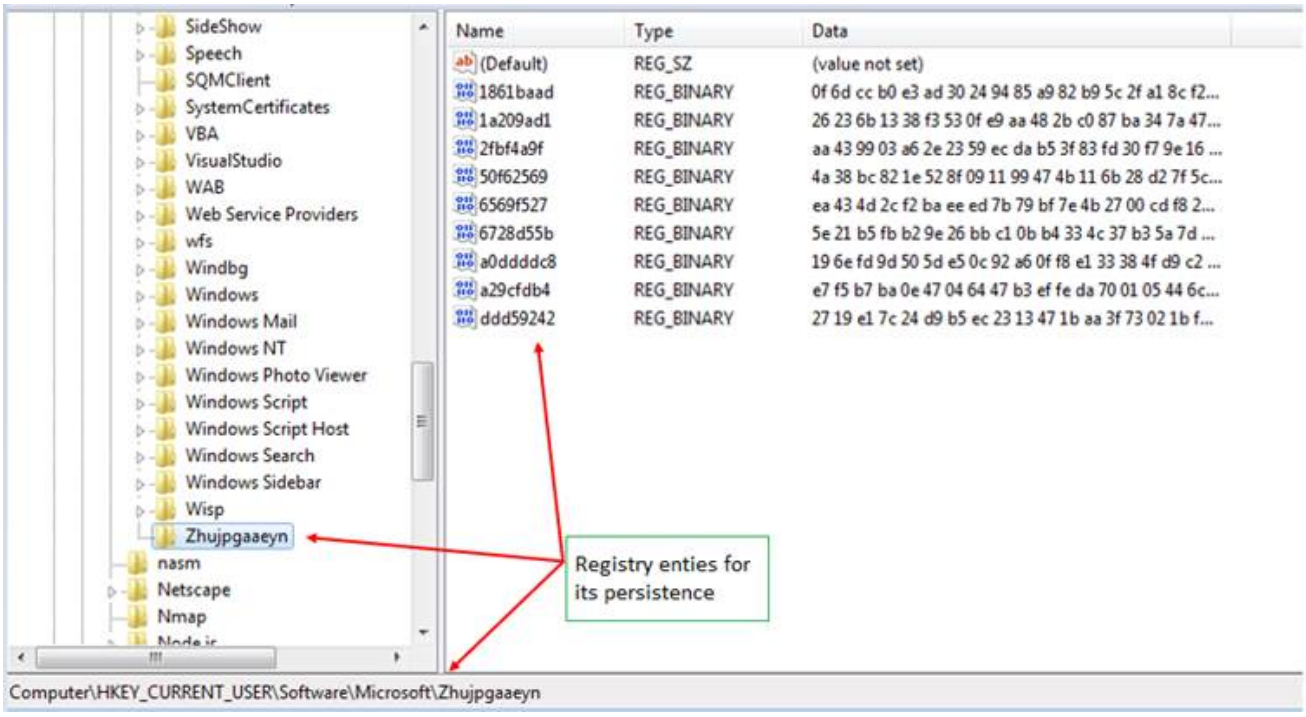


Figure 22: Registry keys created in HKCU Hive

Registry entries are encrypted using system dependant password hash, config IDs, etc., which upon decrypting reveals the BotID/Campaign ID, time of the infection, DLL loader path, etc.

Figure 23: Decrypted registry keys

Scheduled task persistence:

(MITRE: Persistence/Scheduled Task/Job: Scheduled Task)

One of the most significant ways payload versions of Qbot differ are their creation of Scheduled Tasks for persistence. Recent Qbot payloads have been seen using random generated task names during its creations.

The injected Explorer.exe creates scheduled task leveraging schtasks.exe

Figure 24: Scheduled tasks creation

Above command is base64decoded and can be decoded which results in below figure:

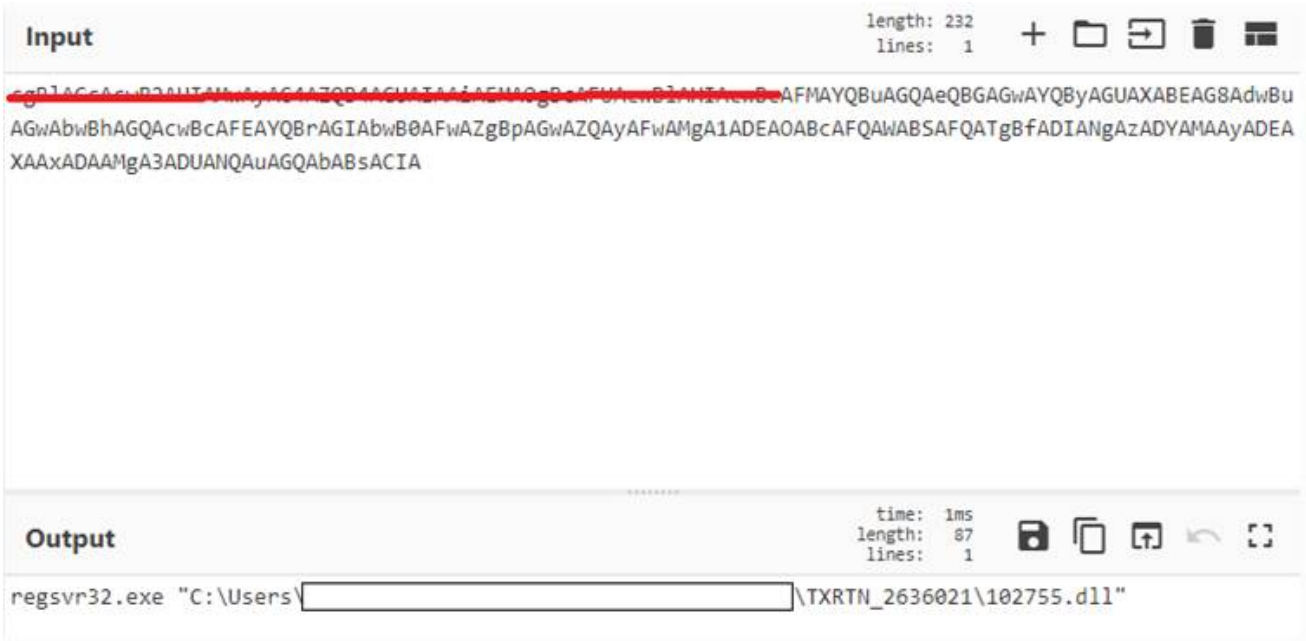


Figure 25: Decoded scheduled task command pointing to loader DLL

Calling home: C2 communications

(MITRE: System Binary Proxy Execution/Command & Control)

Once it executes and successfully infects the victim, it calls home. It pings each of the IPs from its hardcoded C2 list. As the IP responds, it sends the POST request with the victim fingerprinting data.

The injected process (explorer.exe/ wermgr.exe), pings every one of its IP in the C2 lists leveraging wininet, dnsapi and sleep function. Sleep function is set so the requests to IPs aren't creating bottlenecks in communication.

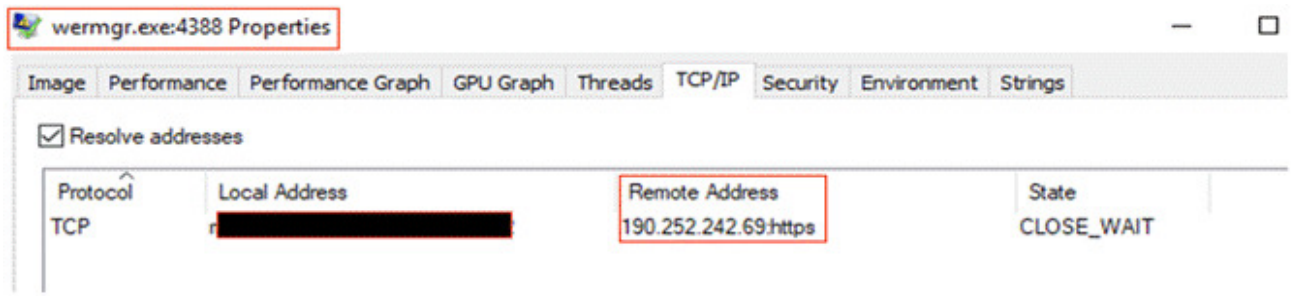


Figure 26: wermgr process initiated a C2 connection

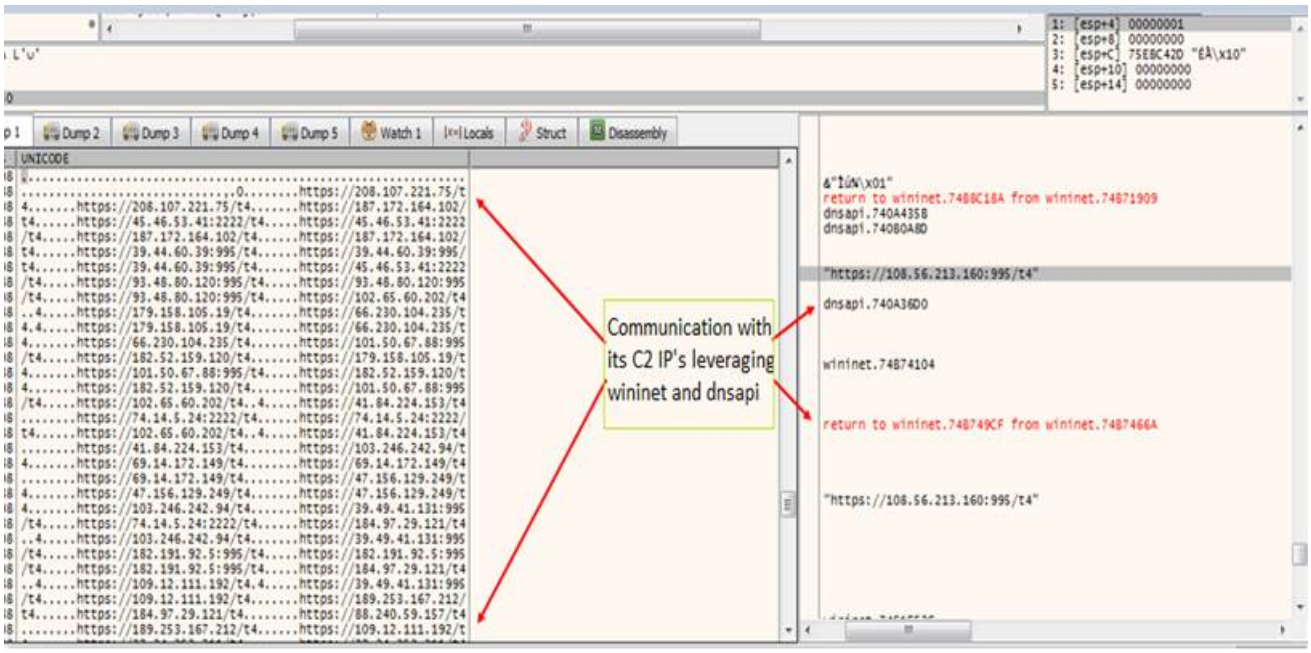


Figure 27: C2 communications

To those C2s which responds back to ping requests, the Qbot sends POST requests which consists of victim fingerprinting details as seen below:

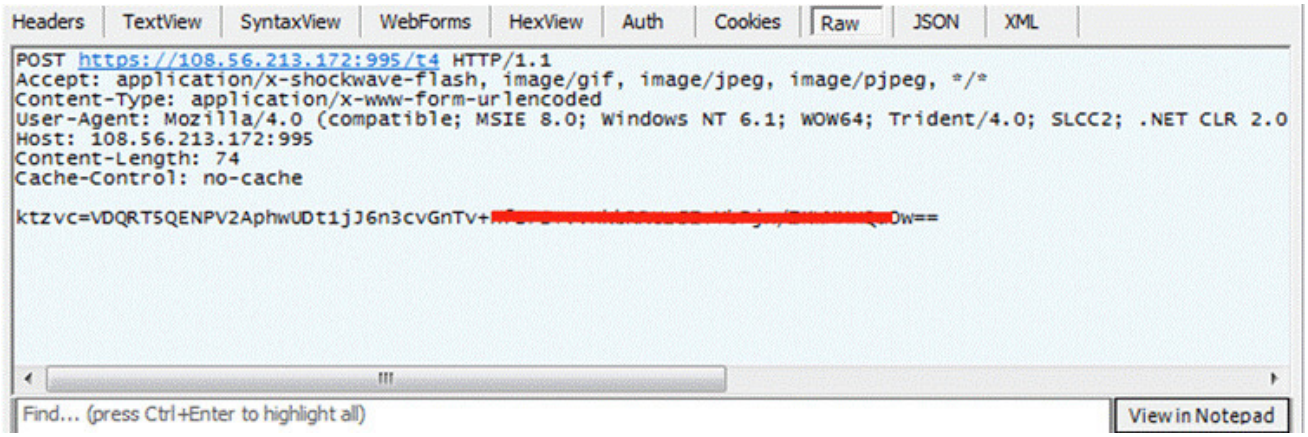


Figure 28: POST requests to its C2 IP

Attack flow

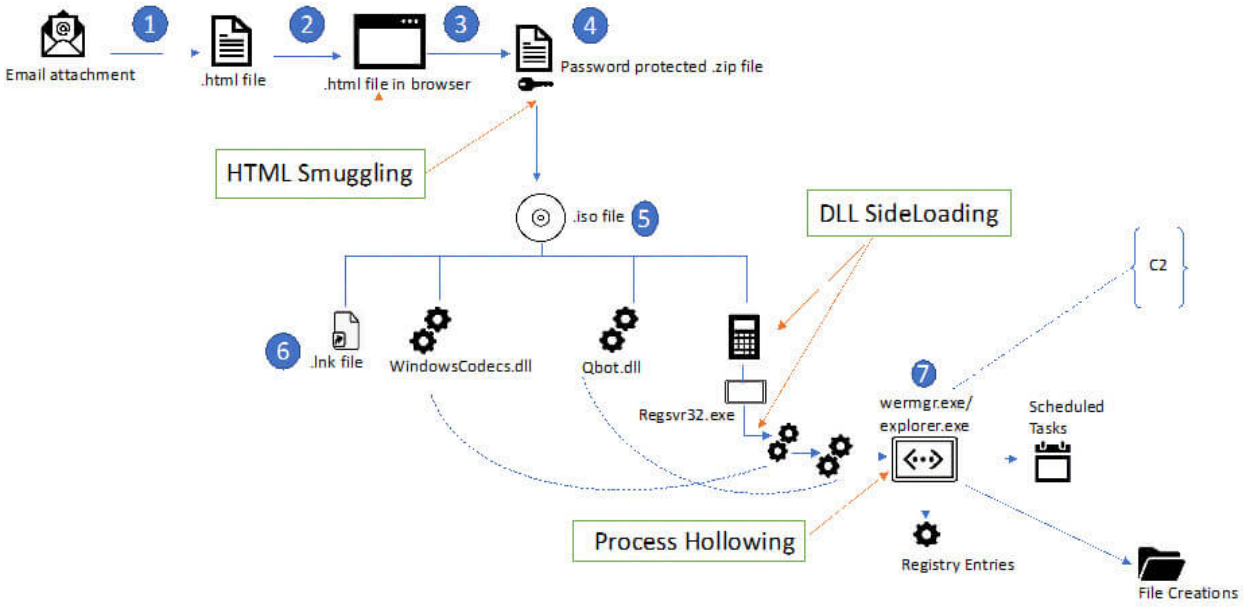


Figure 29: Infection flow overview

Appendix

Detection opportunities

The following sections provide the specific Trellix Products detections, along with custom sigma rules and IOCs to help surface Qbot and related threats.

Trellix Products	Detection Signatures

Trellix Network Security **Trellix VX** **Trellix Cloud MVX** **Trellix File Protect** **Trellix Malware Analysis** **Trellix SmartVision** **Trellix Email Security** **Trellix Detection as Service**	**Static:**
	FE_Trojan_HTM_Phish_209
	FE_Trojan_HTML_Phish_347
	FEC_Dropper_HTML_Generic_11
	FEC_Dropper_HTML_Generic_13
	FE_Loader_Win32_Generic_340
	FE_Loader_Win32_Generic_499
	Dynamic:
	Suspicious Codeinjection on System Processes
	Suspicious Process Schtask Activity
	Suspicious Dllloaded Regsvr on Dropped Dll
	Suspicious Codeinjection on Known Benign File Location
	Suspicious Codeinjection Activity
	Suspicious File Dropped Dll Executed
	Malicious Trojan Indicator
	Malicious Dropper Indicator
Trellix Endpoint Security (HX)	Trojan.GenericKD.49349603
	Trojan.GenericKD.50615557
	Trojan.GenericKD.50616470
	Generic.mg.491e9489c9e11f8b
	POSSIBLE INJECTED EXPLORER (METHODOLOGY)
	QAKBOT G (FAMILY)

Trellix Helix	WINDOWS METHODOLOGY [Suspicious ISO mounts]
	WINDOWS METHODOLOGY [Suspicious Scheduled tasks]
	WINDOWS METHODOLOGY [Regsvr32 -Suspicious child process]
	MALWARE METHODOLOGY [QBot CALC Behaviour]

Helix hunting queries

metaclass:windows ((class=ms_defender category:`advancedhunting-devicefileevents` action:`filecreated`) OR (source:`microsoft-windows-sysmon` eventid=11)) filename:[`*.html`,`*.zip`,`*.iso`,`*.lnk`] filename:/TXRTN_[0-9]{7}/ ?

OR

metaclass:windows ((class=ms_defender category:`advancedhunting-devicefileevents`) OR (source:`microsoft-windows-sysmon` eventid=[11,23,26]) OR (source:`microsoft-windows-security-
auditing` eventid=4663 category:`file system`) filename:[`*.html`,`*.zip`,`*.iso`,`*.lnk`] filename:/TXRTN_[0-9]{7}/ ?

Custom Sigma rules:

```
title: suspicious DLL SideLoading by calc
status: Experimental
description: Detects the DLL-Sideloading of windowscodecs.dll by calc.exe.
date: 04/08/2022
logsource:
  product: windows
  category: image_load
detection:
  selection:
    - ImageLoaded|endswith:
      - '\WindowsCodecs.dll'
    - Image|endswith:
      - 'calc.exe'
  filter:
    - Imageloaded|startswith:
      - 'C:\Windows\System32\'
      - 'C:\Windows\Syswow64\'
```

```
    condition: selection and not filter
  falsepositives:
    - Unknown
  level: high
  tags:
    - DLL Side-Loading
    - T1574.002

=======

  title: Suspicious calc child process
  status: Experimental
  description: Detects the suspicious child process of calc
  date: 04/08/2022
  logsource:
    category: process_creation
    product: windows
  detection:
    selection:
      ParentImage|endswith:
        - '\\calc.exe'
      Image|endswith:
        - '\\regsvr32.exe'
    condition: all of them
  falsepositives:
    - Unknown
  level: high
  tags:
    - SystemBinary Proxy Execution
    - T1218.010

=======

  title: Suspicious process injection to explorer
  status: Experimental
  description: Detects the suspicious regsvr32 child process
  date: 04/08/2022
  logsource:
    category: process_creation
    product: windows
  detection:
```

```
    selection:
      ParentImage|endswith:
        - '\\regsvr32.exe'
      Image|endswith:
        - '\\Explorer.exe'
    condition: all of them
  falsepositives:
    - Uknown
  level: high
  tags:
    - SystemBinary Proxy Execution
    - T1218.010

=======

  title: Suspicious commands arguments from Explorer
  status: Experimental
  description: Detects the suspicious commandlines from explorer
  date: 04/08/2022
  logsource:
    category: process_creation
    product: windows
  detection:
    selection:
        ParentImage|endswith:
        - '\\Explorer.exe'
      commandLine|contains:
        - 'whoami /all'
        - 'arp -a'
        - 'ipconfig /all'
        - 'net view /all'
        - 'cmd /c set'
        - 'nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs'
        - 'nltest /domain_trusts /all_trusts'
        - 'net share'
        - 'netstat -nao'
        - 'net localgroup'
        - 'qwinsta'
      condition: all of them
  falsepositives:
    - Uknown
  level: high
```

```
  tags:
    - System Information Discovery
    - T1082

=======

  title: Suspicious commands arguments from Wermgr.exe
  status: Experimental
  description: Detects the suspicious commandlines from wermgr.exe
  date: 04/08/2022
  logsource:
    category: process_creation
    product: windows
  detection:
    selection:
      ParentImage|endswith:
        - '\\Wermgr.exe'
      commandLine|contains:
        - 'whoami /all'
        - 'arp -a'
        - 'ipconfig /all'
        - 'net view /all'
        - 'cmd /c set'
        - 'nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs'
        - 'nltest /domain_trusts /all_trusts'
        - 'net share'
        - 'netstat -nao'
        - 'net localgroup'
        - 'qwinsta'
    condition: all of them
  falsepositives:
    - Uknown
  level: high
  tags:
    - System Information Discovery
    - T1082

=====
```

```
  title: Explorer Initiated a network Connection
  status: experimental
  description: |
    Adversaries may abuse explorer.exe to proxy execution of malicious payloads and
  connect with C2.
  references:
  date: 04/08/2022
  logsource:
      category: network_connection
      product: windows
  detection:
      selection:
        Initiated: 'true'
        Image|endswith: '\explorer.exe'
      condition: selection
  falsepositives:
      - Legitimate explorer.exe connections over networks
  level: medium
  tags:
      - Defense Evasion
      - T1218.007

=====

  title: WerMgr Initiated a Network Connection
  status: experimental
  description: |
    Adversaries may abuse wermgr.exe to proxy execution of malicious payloads and
  connect with C2.
  references:
  date: 04/08/2022
  logsource:
      category: network_connection
      product: windows
  detection:
      selection:
        Initiated: 'true'
        Image|endswith: '\wermgr.exe'
      condition: selection
  falsepositives:
      - Legitimate wermgr.exe connections over networks
```

```
level: medium
tags:
    - Defense Evasion
    - T1218.007
```

IOCs:

Files

FileName: TXRTN_2636021.html
Hash:5cb20a0bfc5e3e2ae8398b1840adf7ae

— —

FileName:TXRTN_2636021.iso
Hash:17be394b5cd6d74c3709e39f02cd1aa3

— —

FileName: WindowsCodecs.dll
Hash:491e9489c9e11f8b9d3d77239559a194

— —

FileName: 102755.dll
Hash:217f7ddedf40dbe456ce13bf01bd74fc

— —

FileName: zhujpga.dll/ mfvffncbov.dll (cloned dll's)
Hash:217f7ddedf40dbe456ce13bf01bd74fc

Network communications

http://94.59.15.56:2222
http://94.59.15.204:2222
http://94.59.15.166:2222
http://94.36.193.62:2222
http://94.36.193.38:2222
http://94.36.193.154:2222
http://93.48.80.99:995
http://93.48.80.92:995
http://93.48.80.238:995
http://92.132.132.196:2222
http://92.132.132.160:2222
http://92.132.132.112:2222

http://89.211.209.7:2222
http://89.211.209.252:2222
http://89.211.209.156:2222
http://86.97.246.37:1194
http://86.97.246.230:1194
http://86.97.246.217:2222
http://86.97.246.171:1194
http://86.97.246.157:1194
http://86.97.246.143:2222
http://86.97.246.133:2222
http://86.213.75.210:2078
http://86.213.75.17:2078
http://86.213.75.13:2078
http://84.241.8.223:32103
http://84.241.8.203:32103
http://84.241.8.149:32103
http://84.241.8.131:32103
http://81.158.239.89:2078
http://81.158.239.219:2078
http://81.158.239.163:2078
http://81.158.239.10:2078
http://80.11.74.71:2222
http://80.11.74.238:2222
http://80.11.74.179:2222
http://80.11.74.146:2222
http://74.14.5.212:2222
http://74.14.5.178:2222
http://72.252.157.62:995
http://72.252.157.250:990
http://72.252.157.245:990
http://72.252.157.233:995
http://72.252.157.212:990
http://72.252.157.106:995
http://70.51.137.22:2222
http://70.51.137.209:2222
http://70.51.137.204:2222
http://70.51.137.15:2222
http://67.69.166.80:2222
http://67.69.166.36:2222
http://67.69.166.245:2222
http://63.143.92.90:995
http://63.143.92.26:995
http://63.143.92.221:995

http://63.143.92.15:995
http://46.100.25.55:61202
http://46.100.25.153:61202
http://46.100.25.145:61202
http://45.46.53.7:2222
http://45.46.53.77:2222
http://45.46.53.221:2222
http://40.134.246.56:995
http://40.134.246.216:995
http://40.134.246.149:995
http://39.57.56.30:995
http://39.57.56.206:995
http://39.57.56.201:995
http://39.53.124.45:995
http://39.53.124.148:995
http://39.53.124.135:995
http://39.52.59.37:995
http://39.52.59.234:995
http://39.52.59.184:995
http://39.52.221.84:995
http://39.52.221.39:995
http://39.52.221.205:995
http://39.49.41.55:995
http://39.49.41.28:995
http://39.49.41.181:995
http://39.44.60.65:995
http://39.44.60.51:995
http://39.44.60.187:995
http://39.41.16.33:995
http://39.41.16.31:995
http://39.41.16.109:995
http://38.70.253.70:2222
http://38.70.253.56:2222
http://38.70.253.213:2222
http://38.70.253.154:2222
http://37.208.131.96:50010
http://37.208.131.249:50010
http://37.208.131.230:50010
http://37.208.131.224:50010
http://37.186.58.41:995
http://37.186.58.18:995
http://37.186.58.153:995
http://32.221.224.83:995

http://32.221.224.7:995
http://32.221.224.201:995
http://32.221.224.102:995
http://24.178.196.74:2222
http://24.178.196.228:2222
http://24.178.196.227:2222
http://24.178.196.177:2222
http://24.158.23.45:995
http://24.158.23.219:995
http://24.158.23.204:995
http://24.158.23.104:995
http://217.165.157.245:995
http://217.165.157.243:995
http://217.165.157.121:995
http://217.128.122.182:2222
http://217.128.122.112:2222
http://217.128.122.108:2222
http://201.172.23.70:2222
http://201.172.23.6:2222
http://201.172.23.174:2222
http://201.172.23.102:2222
http://196.203.37.228:80
http://196.203.37.212:80
http://196.203.37.190:80
http://196.203.37.106:80
http://186.90.153.39:2222
http://186.90.153.237:2222
http://186.90.153.182:2222
http://186.90.153.116:2222
http://182.191.92.39:995
http://182.191.92.238:995
http://182.191.92.221:995
http://177.94.65.55:32101
http://177.94.65.158:32101
http://177.94.65.146:32101
http://177.189.180.240:32101
http://177.189.180.207:32101
http://177.189.180.135:32101
http://176.45.218.186:995
http://176.45.218.13:995
http://176.45.218.125:995
http://174.80.15.53:2083
http://174.80.15.34:2083

http://174.80.15.165:2083
http://173.21.10.75:2222
http://173.21.10.31:2222
http://173.21.10.116:2222
http://172.115.177.254:2222
https://190.252.242.69:443
http://172.115.177.201:2222
http://172.115.177.127:2222
http://172.115.177.112:2222
http://121.7.223.219:2222
http://121.7.223.20:2222
http://121.7.223.143:2222
http://120.150.218.51:995
http://120.150.218.202:995
http://120.150.218.139:995
http://120.150.218.119:995
http://111.125.245.20:995
http://111.125.245.205:995
http://111.125.245.203:995
http://108.56.213.36:995
http://108.56.213.172:995
http://108.56.213.142:995
http://106.51.48.3:50001
http://106.51.48.248:50001
http://106.51.48.139:50001
http://104.34.212.96:32103
http://104.34.212.33:32103
http://104.34.212.152:32103
http://103.133.11.48:995
http://103.133.11.241:995
http://103.133.11.181:995
http://103.116.178.228:995
http://103.116.178.196:995
http://103.116.178.195:995
http://101.50.67.85:995
http://101.50.67.74:995
http://101.50.67.72:995
http://100.38.242.94:995
http://100.38.242.193:995
http://100.38.242.149:995
http://100.38.242.134:995
http://1.161.79.219:995
http://1.161.79.166:995

http://1.161.79.139:995
https://182.52.159.207
https://89.101.97.204
https://86.97.10.196
https://108.56.213.172:995
https://103.133.11.241:995
https://86.97.246.133:2222
https://39.41.16.109:995

The information presented here describes computer security research for educational purposes only and the convenience of Trellix customers. Trellix conducts research in accordance with its Vulnerability Reasonable Disclosure Policy | Trellix. Any attempt to recreate part or all of the activities described is solely at the user's risk, and neither Trellix nor its affiliates will bear any responsibility or liability.