# **Demystifying Qbot Malware**

**// trellix.com**/en-us/about/newsroom/stories/threat-labs/demystifying-qbot-malware.html



Register Now Learn More

## **Stories**

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

# **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.

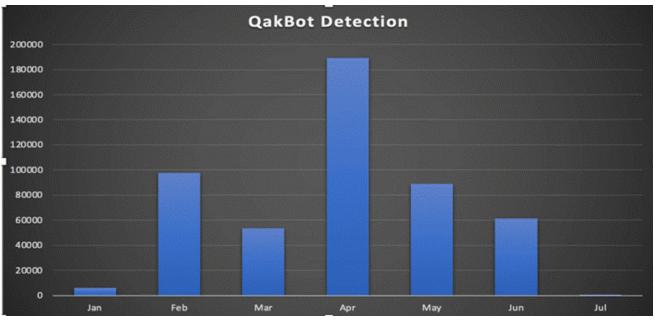


Figure 1: Qbot infection trend over Q1-Q2

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:

## QBot Threat Landscape **Detection Summary**

- Regions targeted by Qbot
  - · Australia, Bahrain, Belgium, Brazil, Canada, Colombia, Croatia, Cyprus, Denmark, Egypt, France, Germany, Hong Kong, India, Ireland, Israel, Italy, Japan, Jordan, Kenya, Korea, , Republic of, Macao, Malaysia, Mexico, Nigeria, Pakistan, Peru, Philippines, Qatar, Saudi Arabia, Singapore, South Africa, Spain, Sweden, Taiwan, Thailand, Turkey, United Kingdom, United States

Figure 2: QbBot threat landscape summary

## Infection vector: email

- Verticals targeted by Qbot
  - Aerospace/Defense Contractor, Education, Energy/Utilities, Entertainment/Media/Hospitality, Financial Services, Government: Federal, Government: State & Local, Healthcare, High-Tech, Insurance, Legal, Manufacturing, Retail, Service Provider, Services/Consulting, Software Development, Telecom, Transportation, VAR/VAD/OEM

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.

#### Variants : Schemeless URLs

- Attack starts with a fake response to sign documents containing malicious links without any protocol due to which it was not treated as actionable link by outlook.
- While some mail clients do add protocol to them and render them actionable link while displaying it to user.
- This tactic of schemeless URLs in emails are used as evasion to bypass security products from extracting and analyzing malicious URLs thus bypassing them.
- Trellix Email Security extracts such URLs from emails and sends it for analysis thus bypassing such evasion technique and detecting this attack.

[EXT] Re: Rspuns automat: [ENT] Collection in a state of R	ちょう
	Wednesday, 13 July 2022 at 3 53 PM
Nu am văzut nici un răspuns de la tine, asa că aici te revin că hârtiile pentru a semna	

\*) grampalikadabhadi.in/consecteturmollitia/veritatisconseguatur-8213222

\*\*) new.drceight.com/perspiciatisminima/delenitiomnis-8213222

Figure 3: Qbot Email (Variant: Schemeless URLs)

This email was scanned by Bitdefender

#### 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 Obot 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.

Re: RE:	URGENTINVOICES FOR PAYMENT	6 4
AT	C Alicetine	Tuesday, 7 June 2022 at 10:26
-	50991053_927006 911.4 KB	
	Download All + Preview All	
IExternal		
	a file you wanted can be located down below. know if there are any concerns.	
Thank y		
Te	alling now From: o: Davis, Normanick Subject: RE: URGENTINVOICES FOR PAYMENT [Externa all when you get this message. Thanks] Don't forget to get a quote from Q	

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

#### **MITRE** mappings

Attack Behaviour	Example	MITRE ATT&CK
Malicious email with an <i>.html</i> attachment.	Threat actor lured user to open malicious email with malicious <i>.html</i> attachment.	<u>Initial Access /</u> <u>Phishing (T1566)</u>
The <i>.html</i> file opens in a browser and uses <i>HTML</i> <i>Smuggling</i> to drop an embedded <i>.ZIP</i> 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.	<u>Defense Evasion /</u> <u>Subvert Trust Controls</u> <u>Mark-of-the-Web</u> <u>Bypass (T1553.005)</u>
User executed malicious Windows Shortcut, which executes calc.exe from mounted ISO image.	User clicks the malicious <i>LNK file</i> from the <i>ISO file</i> . As rest of the items will be hidden, only lnk file will be visible to the user.	Execution / User Execution (T1204.002)
<i>calc.exe</i> 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)
<i>Calc.exe</i> spawns Microsoft signed binary ( <i>RegSvr32.exe</i> ) to executes <i>Qbot dll</i> ( <i>loader</i> )	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	<i>Symon event id: 13 Event Action: Registry Value Set</i>	<u>Defense</u> Evasion/Modify <u>Regsitry</u>
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.	<u>Discovery / System</u> 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.

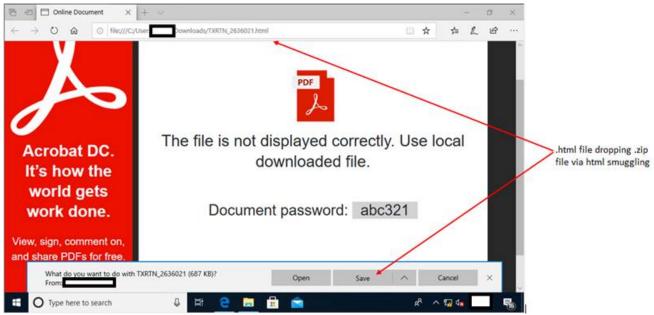


Figure 7: The html file renders in the user browser showing the password to open the dropped zip file

# 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.

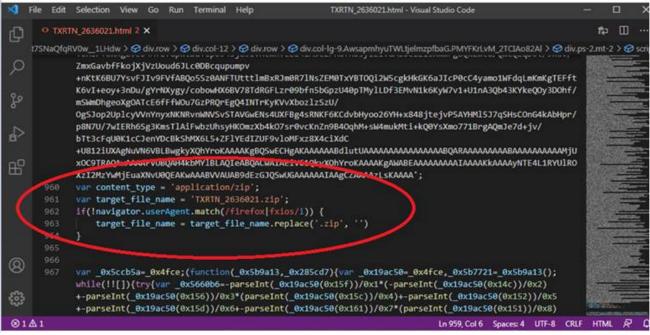


Figure 8: HTML content showing the smuggled zip payload

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

	:\Users	\0	ownloa	ds\TXRT	N_26360	21.zip\2	518\				-		×
File	Edit Vie	ew Fa	avorites	Tools	Help								
4	-	V	-	-	×	บี							
Add	Extract	Test	Сору	Move	Delete	Info							
1	C:\Us	er	Dov	vnloads\	TXRTN_2	2636021	.zip\2518\						~
Nam	e				Siz	e	Packed Size	Modified	Created	Accessed		Att	ributes
© T)	KRTN_2636	021.is	0		2 752 51	2	703 074	2022-07-11 08:42					А
			-										
				-									
				7	in file	with i	so file insi	da					
				-	up me	with	so me msi	ue					

< >
0 / 1 object(s) selected

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))

File Home Sh	are View Drive Tools			~
$\leftrightarrow \rightarrow \sim \uparrow \odot$	This PC > DVD Drive (E:)	~ č	Search DVD	Drive (E:)
	Name	Date modified	Туре	Size
🖈 Quick access	TXRTN_2636021	7/8/2022 5:45 AM	Shortcut	2 KB
Desktop	*	1/0/2022 3:43 AW	SHOREOL	2 10
🕹 Downloads	*			
🔮 Documents	*			
Pictures	*			
👌 Music				
Program Files				

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* 

Vame	Date modified	Туре	Size
🚳 102755.dll	7/11/2022 8:48 PM	Application extens	687 KB
📄 calc.exe	11/21/2010 8:55 AM	Application	758 KB
TXRTN_2636021	7/8/2022 6:15 PM	Shortcut	2 KB
WindowsCodecs.dll	7/11/2022 7:45 PM	Application extens	5 KB

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

# ISTHIS CALCULATOR?

;

H

IGETE

_		TXRTN_26	36021 Prop	erties			×		-
ocal Dis	k (C:) > Users >	Compatibili	ty Se	curity	Details	Previous	Versions		- 4
Tools	Help	General	Shortcut	Options	Font	Layout	Colors		
] Open	Share with 👻 New fold		TXRTN_2	636021					1.25
Î	Name 102755.dll calc.exe	Target type	: Application: System					87 KB 88 KB	Conte
	TXRTN_2636021	Target:	C:\Wit	ndows\Syste	em32\cmd.e	xe /q /c cald	c.exe	2 KB	
s	(3) WindowsCodecs.dll	Start in: Shortcut ke	y: None					5 KB	
		Run:	Minimi	ized			•		
		Comment: Open Fi	le Location	Chang	ge Icon	Advance	ed		

ent t

2636021 Date modified: 7/8/2022 6:15 t Size: 1.70 KB	OK Cancel Apply

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".

7			24	₽7 ■₩₽₽₩₩		
	Process Name	PID Operation	Path		Result	Detail
11:38:	calc.exe	4612 🕞 CreateFile	C:\Users\	Desktop\calc.exe.Local	NAME NOT FOU	ND Desired Access: R
11:38	calc.exe	4612 TreateFile	C:\Users\	\Desktop\UxTheme.dll	NAME NOT FOU	ND Desired Access: R
11:38:	calc.exe	4612 📻 CreateFile	C:\Users\	Desktop\calc.exe.Local	NAME NOT FOU	ND Desired Access: R
11:38	calc.exe	4612 🔐 CreateFile	C:\Users\	\Desktop\WINMM.dll	NAME NOT FOU	ND Desired Access: R
11:38	calc.exe	4612 📻 CreateFile	C:\Users\	Desktop\VERSION.dll	NAME NOT FOU	ND Desired Access: R
11:38	calc exe	4612 TreateFile	C:\Users\	Desktop\WindowsCodecs.dll	NAME NOT FOU	ND Desired Access: R
* 0					<ul> <li>Calc.exe searching for Window desktop path where it is palced the calc.exe shown above but p directory.</li> </ul>	l. This is a copy of

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.

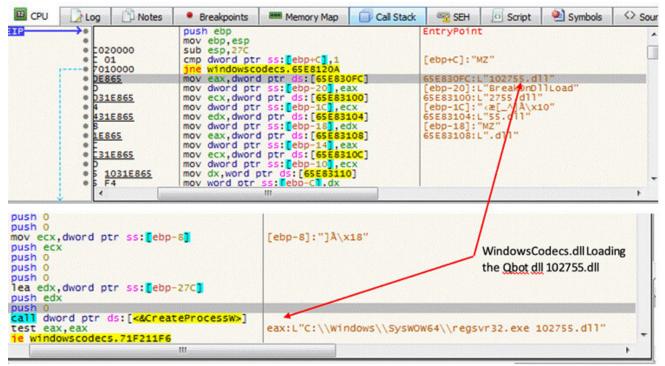


Figure 14: Windows codecs leveraging regsvr32 via CreateProcessW to load the Qbot loader DLL

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

Import /C:/Users/		2518/TXRTN_2636021/10	27 🗪
Format:	Portable Executable (PE)		- 0
Language:	x86:LE:32:default:borlanddelphi		
Destination Folder:	auto:/		
Program Name:	102755.dll		
		Opti	ons

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.

Exeinfo PE - ver	.0.0.5.3 by A.S	L - 1031+71 si	ign 2018.	09.25		
	bin				<i>"</i> Р <u>н</u>	
Entry Point :	0000651C	00 < EP	Section :	.text		
File Offset :	0000591C	Firs	t Bytes :	55.8B.EC.51.83		Plu
Linker Info :	14.29	Sub	System :	Windows GUI	PE	
File Size :	0001F800h	< N Ove	erlay :	NO 00000000	0	2
DLL 32bit-Lib	orary image	RE	5/OVL : 0	/ 0 % 2022	而	
(1) Antonio and a support of the		and a second a second	015 [Win	Vista ] [ Debug:	Scan / t	Rig
Lamer Info - I	Help Hint - Unpa	ck info		No. 330 ms; 54		
	try OllyDbg v2	- www.ollydbg.c	de or IDA	7 www.hex-ray		≥>

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".

FF	cmp eax, FFFFFFF	eax:L"C:\\INTERNAL\\empty"	
0C	<pre>lea eax,dword ptr ss:[ebp+C]</pre>		
	push eax	<pre>eax:L"C:\\INTERNAL\\empty"</pre>	
	je 1.10006587		
290000	call 1.1000SEFB		
	pop ecx xor eax,eax	eax:L"C:\\INTERNAL\\empty"	
	imp 1.10006604	carre c. ((interiore (()))	
290000	call 1.10008EFB		
4 CA010000	mov dword ptr ss:[esp],1CA	[esp]:L"C:\\INTERNAL\\empty"	
ABFFFF	call 1.1000109A		
	pop ecx		-
4	III		+

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.

8BC8	mov ecx, eax	eax:L"SELF_TEST_1"
E8 6C3A0000 8945 0C 85C0 74 37 68 D90E0000 6A 54 5A	<pre>call 1.1000A00C mov dword ptr ss:[ebp+C],eax test eax,eax je 1.100065DE push ED9 push 54 pop edx</pre>	eax:L"SELF_TEST_1"

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.

A 74 82	<pre>call dword ptr ds: [eax+74] mov dword ptr ds: [1001F89C],eax test eax,eax je 1.10006583 xor eax,eax inc eax leave</pre>	-
•	CreateThread>]= <kernel32.createthread></kernel32.createthread>	•

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:

- 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

• 词 🤇	Open	Include in library      Share with	New folder		# .	6
wnloads	^	Name	Date modified	Туре	Size	
ent Places		Ja AddIns	10/5/2021 7:25 AM	File folder		
ities		CLR Security Config	4/11/2021 12:46 PM	File folder		
		鷆 Credentials	4/11/2021 2:21 AM	File folder		
ries		3 Crypto	8/9/2022 8:43 AM	File folder		
uments		🎉 Document Building Blocks	10/13/2021 12:44	File folder		
sic		🔐 Excel	10/5/2021 7:25 AM	File folder		
tures		📕 Fhdvy 🔪	8/9/2022 8:36 AM	File folder		
eos		)) Installer	8/9/2022 9:59 AM	File folder		
egroup	ш	MMC create a co	ated to drop its dll opy of itself. These d upon execution			

Folder Creations:

Figure 20: Folder created in AppData Roaming, dropping DLL's Dropped DLLs are loaded via regsvr32.exe as shown in below:

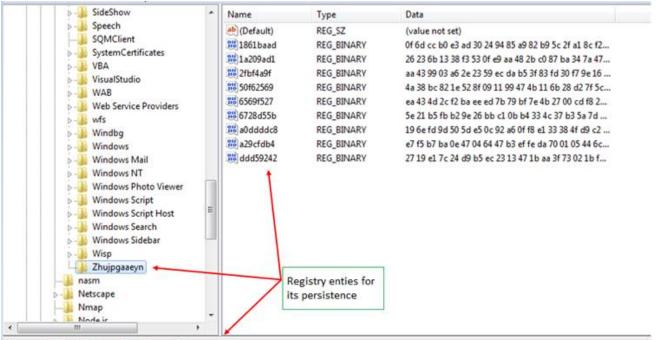


Figure 21: Executing Dropped DLLs via RegSvr32

# **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.



Computer\HKEY\_CURRENT\_USER\Software\Microsoft\Zhujpgaaeyn

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.

C:\Users\keydcoder>python_keydcoder.py -k_HKEY_CURRENT_USER\Softw Password: 'keydcoder>python_keydcoder.py -k_HKEY_CURRENT_USER\Softw Password_Hash:	are\Microsoft\Zhujpgaaeyn
Registry key path: HKEY_CURRENT_USER\Software\Microsoft\Zhujpgaaeyn\2fbf4a9f RC4 key: a1 21 2d e0 81 fc c8 08 02 b0 2e 46 83 38 09 90 d6 47 54 f2	
Decrypted value: 000000000: 03 01 1F 33 3B 31 3B 31 36 36 30 30 31 35 33 343;1;166001534 00000010: 35 7C 33 3B 32 31 3B 31 36 36 30 30 31 35 33 34 5:3;21;166001534 00000020: 35 FB AB DF 83 BC 59 C1 90 86 5Y	
Registry key path: HKEY_CURRENT_USER\Software\Microsoft\Zhujpgaaeyn\1a209ad1 RC4 key: 00 c9 4c 40 72 48 ed d5 ef ef 34 f3 2f f4 41 fb 4e 05 4b 31	$\backslash$
Decrypted value: 000000000: 04 01 80 43 3A 5C 55 73 65 72 73 5C 53 61 6E 64 000000000: 74 66 C 61 72 65 5C 41 70 70 44 61 74 61 5C 52 000000000: 5C 66 86 46 76 79 5C 7A 68 75 6A 70 67 61 2E 64 000000000: 5C 66 86 46 76 79 5C 7A 68 75 6A 70 67 61 2E 64 0000000000: 5C 66 19 52 64 41 C6 D5 7B 93 35 42 8F 64 6D FB 0000000060: DB 88 CD CE 0C 9A F7 C2 000000060: DB 88 CD CE 0C 9A F7 C2	Decrypted Registry Entries Showing dropped DLL and its campaign tag
Registry key path: HKEY_CURRENT_USER\Software\Microsoft\Zhujpgaaeyn\baad RC4 key: cd ad 7f b4 64 f5 2a 7b f9 70 a4 b9 0d 75 0a a1 2f 63 ac 2f	obama200
Decrypted value: 00000000: 02 01 08 02 4B 4C EC 6E 1A 65 CB 65 A2 83 A6 BEKL.n.e.e 00000010: 9D .	
Registry key path: HKEY_CURRENT_USER\Software\Microsoft\Zhujpgaaeyn\a@ddddc8 RC4 key: dc c2 85 50 2d 05 8d f3 79 3b 54 aa cc 55 78 6e 6d ca 3d 5c	
Decrypted value: 000000008: 03 01 09 6F 62 61 6D 61 32 30 30 C3 FD 87 58 E3obana200X. 000000018: AE 25 83 85 26 AA C7 DA 8B C8 4C 4B 6D 64 .+.X.&LKAd 000000028: 02 3D 6D E3 1E 69 BC C1 FA 4D .=niM	

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:

Input	length: lines:		+		Ð	Î	-
cgBlaccacuBlautathinyAcAAZQBAACUATAAAAEMAAgBaAEWAcuBla	HIACHBCAFMAYQBUAGQA	eQBG	AGWAY	ByAG	UAXAB	BEAG8	AdwBu
AGwAbwBhAGQAcwBcAFEAYQBrAGIAbwB0AFwAZgBpAGwAZQAyAFwAM XAAxADAAMgA3ADUANQAuAGQAbABsACIA	IgA1ADEAOABcAFQAWABS	AFQA	TgBfAD	)IANg	AzADY	(AMAA)	YADEA
Output	time: length: lines:	1ms 87 1	8	ē	(†)	5	53
regsvr32.exe "C:\Users\	\TXRTN_2636	021\	102755	5.dll			

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.

werm	ngr.exe:4388 P	Properties							-	(
mage	Performance	Performance Graph	GPU Graph	Threads	TCP/IP	Security	Environment	Strings		
∠] Res	olve addresses	5								
Proto	~	s ocal Address		Ren	note Addr	ess		State		

Figure 26: wermgr process initiated a C2 connection

* < *** [	в.)	1	,	1: [esp+4] 00000001 2: [esp+6] 0000000 3: [esp+C] 75E8420 "ÉÅ\x10" 4: [esp+10] 0000000 5: [esp+14] 0000000
UNICODE (UNICODE (1) (1) (1) (1) (1) (1) (1) (1)	https://i87.172.164.102/ https://i87.172.164.102/ https://i87.172.164.102/ https://i87.172.164.102/ https://i84.65.3.41:2222 https://i84.65.3.41:2222 https://i84.65.3.41:2222 https://i84.65.3.41:2222 https://i82.20.104.235/t https://i82.20.104.235/t https://i82.20.104.235/t https://i82.20.104.235/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.23.155.120/t https://i82.135.129.249/t https://i82.135.129.249/t https://i82.1351.259.549/t https://i82.139.125.1595 https://i82.139.125.121/t4 https://i82.139.125.121/t4 https://i82.1351.257.14	Communication with its C2 IP's leveraging wininet and dnsapi	<pre>4"1/4%x01" return to wininet.7488C18A fro dnsap1.740A4358 dnsap1.740B0A80 "https://108.56.213.160:995/t4 dnsap1.740A36D0 wininet.74874104 return to wininet.748749CF fro "https://108.56.213.160:995/t4</pre>	n wininet.74874664

#### 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:

Headers	TextView	SyntaxView	WebForms	HexView	Auth	Cookies	Raw	JSON	XML	
Accept: Content User-Ag Host: 1 Content Cache-C	applicati -Type: application ent: Mozi 08.56.213 -Length: 5 ontrol: no	.172:995 74	vave-flash, www-form-u npatible; M	image/gi rlencoded SIE 8.0;	1					C2; .NET CLR 2.
<	in an									,
Find (p	ress Ctrl+Ent	er to highlight all	)							View in Notepad

Figure 28: POST requests to its C2 IP

#### Attack flow

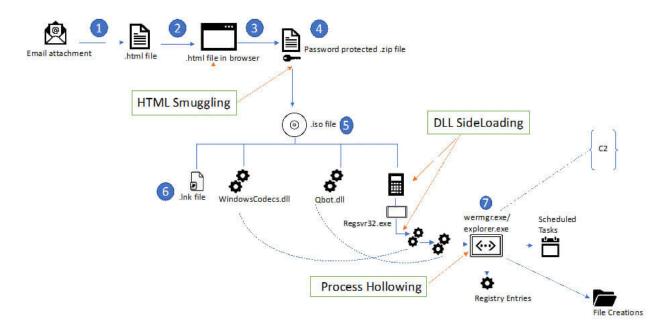


Figure 29: Infection flow overview

# 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	Static:					
Trellix File Protect Trellix Malware Analysis	FE_Trojan_HTM_Phish_209					
Trellix SmartVision Trellix Email Security	FE_Trojan_HTML_Phish_347					
Trellix Detection as Service	FEC_Dropper_HTML_Generic_11					
Service	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 DIlloaded Regsvr on Dropped DII					
	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-securityauditing` 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 <u>Vulnerability Reasonable Disclosure Policy | Trellix</u>. 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.