

Lab 3: Network Sniffing

Procedure

Part 1, first screenshot:

- Downloaded and installed Wireshark and rebooted my computer once Wireshark was installed.
- Opened Wireshark application and put it into capture mode on my Wi-Fi.
- Went to my browser (Edge) and typed a random domain name that doesn't exist and tried to connect <http://www.jahfjrigjfdivobofkjhkd.com>.
- Opened a terminal.
- Then I typed: ping www.google.com and then enter via PowerShell.
- Visited www.amazon.com and click on "sign in" (but did not sign in).
- Stopped packet capture.
- Applied a simple filter, *dns*, to display DNS packets, took a screenshot of the results, and pasted into my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide one*).

Part 2, second screenshot:

Found the DNS packets for the website that doesn't exist (<http://www.jahfjrigjfdivobofkjhkd.com>) and located in the Domain Name System (response) by applying *dns.qry.name contains www.jahfjrigjfdivobofkjhkd.com* filter. Here I observed the first packet and the sixth packet. I know the first packet is the request and sixth packet is the response based on their transaction IDs, *0x3011* (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide two*).

Part 3, third screenshot:

- Removed the DNS packet filter.
- Applied a filter, *icmpv6*, to display ICMP packets, took a screenshot of the filtered results, and pasted in my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide three*).

Part 4, fourth screenshot:

Found the ICMP packets that contained the ping data I sent, took a screen capture and pasted into my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide four*). I analyzed and compared the Source and Destination contents, specifically the IPv6 addresses, via Wireshark to the IPv6 address in the terminal. I noticed the same IPv6 address in both the packet list pane and in the terminal.

Part 5, fifth screenshot:

- Found the first TLS x.x "Client Hello" packet associated with the IP address for www.amazon.com.
- Expanded packet contents and under "Transport Layer Security," drilled down to find the list of cipher suites supported and took a screen capture and pasted it into my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide five*).

Part 6, sixth screenshot:

- Opened the "Lab 3 Capture" pcap file, provided by instructor, in Wireshark
- In Wireshark, selected "Edit," then "Find Packet," then pasted the hex values for a PDF file in the search bar and selected "Hex value" from the drop-down display then clicked "Find."

- Right clicked on the packet that contains the hex values for a .pdf file header, selected “Follow,” then “TCP Stream.”
- In the “Show data as” window, selected “Raw.” Then selected “Save as” and saved file to my desktop with filename “capture” and with a .pdf extension.
- Took a screenshot of the saved file and pasted it into my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide six*).

Part 7, seventh screenshot:

Opened the file as a .pdf, took a screenshot of it opened and pasted it into my PowerPoint presentation (*reference separate file, ng_raymond_IS3513_Lab 3.pptx, slide six*).

Questions for this Lab Exercise

1. What is the Domain Name System (DNS)? What would happen if the DNS didn't function properly?

DNS is an important part of the internet infrastructure. Think of it as sort of like a phone book of the internet (*GoDaddy, 2023*). DNS translates domain names into numerical IP addresses that computers use to identify each other on a network. DNS works through a hierarchy of nameservers, from root nameservers, through top-level domain nameservers, down to authoritative nameservers for specific domains. When you type a URL into your web browser, it asks your internet service provider's DNS server to find the corresponding IP address. If your ISP's DNS server doesn't know the address, it will query other DNS servers up the hierarchy until it finds one that does.

If the DNS didn't function properly, it would essentially disrupt the internet. When a user enters a URL into a browser, a non-functioning DNS would not be able to translate that URL into an IP address. This means that the browser wouldn't know where to find the website the user is attempting to visit. Essentially, without DNS, every user would need to remember and enter the numerical IP addresses for each website, which is not practical or user-friendly.

2. What is the purpose of ICMP?

The Internet Control Message Protocol (ICMP) is a supporting protocol in the Internet protocol suite. It is primarily used by network devices, like routers, to send error messages and operational information indicating whether a requested service is available, or a requested route is accessible (*Lutkevich, 2023*). ICMP can be utilized to relay query messages, for instance, to check if a host is up or network-available (known as a “ping” like how I pinged for www.google.com in this lab). Although it's not used for sending application data, ICMP provides necessary functionality for managing and troubleshooting an IP network.

3. What is TLS? Which version did you observe in your packet capture?

Transport Layer Security (TLS) is a cryptographic protocol used to secure communications over a network (*Cloudflare, n.d.*). It provides privacy and data integrity between two communicating applications, and it's widely used for internet communications across web browsers and servers, email, instant messaging, and voice-over IP (VoIP).

In this lab I observed TLSv1.2. TLSv1.2 refers to version 1.2 of the TLS protocol. TLSv1.2 introduces some improved features compared to its predecessors, including the expansion of support for cryptographic algorithms and enhanced security (*Keycdn*, 2022). For example, it deprecated the use of MD5 and SHA-224 cryptographic hash functions due to identified vulnerabilities (*Keycdn*, 2022). Also, TLSv1.2 made the use of extensions more flexible.

4. Is there a Wireshark filter that could have saved you time in finding the “Client Hello” packet and cipher suites? What is it?

Yes, `tls.handshake.type == 1` (*W3schools*, 2023).

5. What is FTP? How are you able to capture the contents of a .pdf using Wireshark in this example?

File Transfer Protocol (FTP) is a standard network protocol used to transfer files from one host to another over the internet (*Kerner & Burke*, 2021). FTP is built on a client-server model architecture using separate control and data connections between the client and the server. It uses clear-text—unencrypted—transmission, which means that all the data including the usernames, passwords, and content are sent in a form that is easily readable (*JavaTPoint*, n.d.).

Wireshark is a network protocol analyzer that allows users to inspect data packets transferred over a network. When a file is transferred using FTP—with the understanding FTP is unencrypted—Wireshark captures the individual packets containing the file data. For the PDF files, I assess Wireshark users can find the relevant packets for the file transfer by looking for the FTP RETR (retrieve) or STOR (store) command in the packet details, which includes the filename. By following the TCP stream, I isolated the data packets specific to the PDF file. Wireshark provided an option to view this stream as 'Raw' data, and this raw data can then be saved as a PDF file. This may not work for encrypted files.

References

Cloudflare. (n.d.). *What is TLS (Transport Layer Security)?* Retrieved from Clouflare: <https://www.cloudflare.com/learning/ssl/transport-layer-security-tls/>

GoDaddy. (2023). *What is DNS?* Retrieved from GoDaddy: <https://www.godaddy.com/help/what-is-dns-665>

JavaTPoint. (n.d.). *FTP*. Retrieved from JavaTPoint: <https://www.javatpoint.com/computer-network-ftp>

Kerner, S. M., & Burke, J. (2021, May). *FTP (File Transfer Protocol)*. Retrieved from TechTarget: https://www.techtarget.com/searchnetworking/definition/File-Transfer-Protocol-FTP?Offer=abMeterCharCount_ctrl

Keycdn. (2022, February 22). *TLS 1.2 vs TLS 1.1*. Retrieved from <https://www.keycdn.com/support/tls-1-2-vs-tls-1-1>

Lutkevich, B. (2023). *ICMP (Internet Control Message Protocol)*. Retrieved from TechTarget: <https://www.techtarget.com/searchnetworking/definition/ICMP>

W3schools. (2023). *Wireshark Tls Client Hello Filter*. Retrieved from W3schools: <https://www.w3schools.blog/wireshark-tls-client-hello-filter>

DNS query via Wireshark

*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Answer RRs	Respc	Info
1	11:33:24.357898	192.168.1.254	192.168.7.78	DNS	166	0		Standard query response 0xfd71 HTTPS completion.amazon.com SOA ns-179.awst
4	11:33:24.362702	192.168.1.254	192.168.7.78	DNS	410	11		Standard query response 0x1ef0 AAAA images-na.ssl-images-amazon.com CNAME
6	11:33:24.362702	192.168.1.254	192.168.7.78	DNS	140	2		Standard query response 0xfee8 A m.media-amazon.com CNAME c.media-amazon.
10	11:33:24.365631	192.168.1.254	192.168.7.78	DNS	95	0		Standard query response 0x047b AAAA completion.amazon.com
15	11:33:24.876903	192.168.1.254	192.168.7.78	DNS	241	5		Standard query response 0xb79f A p11.techlab-cdn.com CNAME p11.techlab-cd
21	11:33:25.398098	192.168.1.254	192.168.7.78	DNS	270	3		Standard query response 0x79bf HTTPS p11.techlab-cdn.com CNAME p11.techla
32	11:33:26.261418	192.168.7.78	192.168.1.254	DNS	87	0		Standard query 0x814a AAAA browser.events.data.msn.com
33	11:33:26.261635	192.168.7.78	192.168.1.254	DNS	87	0		Standard query 0xe5aa A browser.events.data.msn.com
34	11:33:26.261735	192.168.7.78	192.168.1.254	DNS	87	0		Standard query 0xb7d2 HTTPS browser.events.data.msn.com
41	11:33:26.274567	192.168.1.254	192.168.7.78	DNS	269	2		Standard query response 0x814a AAAA browser.events.data.msn.com CNAME glo
42	11:33:26.274567	192.168.1.254	192.168.7.78	DNS	268	2		Standard query response 0xb7d2 HTTPS browser.events.data.msn.com CNAME glo
43	11:33:26.274567	192.168.1.254	192.168.7.78	DNS	226	3		Standard query response 0xe5aa A browser.events.data.msn.com CNAME glo

> Frame 32: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface \Device
> Ethernet II, Src: IntelCor_13:62:67 (bc:17:b8:13:62:67), Dst: eero_36:d1:d2 (9c:57:bc:36
> Internet Protocol Version 4, Src: 192.168.7.78, Dst: 192.168.1.254
> User Datagram Protocol, Src Port: 50379, Dst Port: 53
▼ Domain Name System (query)
 Transaction ID: 0x814a
 Flags: 0x0100 Standard query
 Questions: 1
 Answer RRs: 0
 Authority RRs: 0
 Additional RRs: 0
 Queries
 [Response In: 41]

0000	9c 57 bc 36 d1 d2 bc 17 b8 13 62 67 08 00 45 00	·W·6..... ·bg··E·
0010	00 49 c5 0c 00 00 80 11 00 00 c0 a8 07 4e c0 a8	·I..... ···N··
0020	01 fe c4 cb 00 35 00 35 8a e3 81 4a 01 00 00 01	···5·5 ···J···
0030	00 00 00 00 00 00 07 62 72 6f 77 73 65 72 06 65	·····b rowser·e
0040	76 65 6e 74 73 04 64 61 74 61 03 6d 73 6e 03 63	vents·da ta·msn·c
0050	6f 6d 00 00 1c 00 01	om.....

Frame (frame), 87 bytes

Packets: 9421 · Displayed: 448 (4.8%) · Dropped: 0 (0.0%)

Profile: Default

Part 1, first screenshot



dnsqry.name contains www.jahfjrigjfjdivobofkjhkd.com

X → +

No.	Time	Source	Destination	Protocol	Length	Answer RRs	Respc	Info
1098	11:33:30.885585	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x3011 AAAA www.jahfjrigjfjdivobofkjhkd.com
1099	11:33:30.885768	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x278c A www.jahfjrigjfjdivobofkjhkd.com
1100	11:33:30.903318	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x7995 AAAA www.jahfjrigjfjdivobofkjhkd.com
1101	11:33:30.903482	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x1868 A www.jahfjrigjfjdivobofkjhkd.com
1102	11:33:30.903557	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x6853 HTTPS www.jahfjrigjfjdivobofkjhkd.com
1105	11:33:30.910980	192.168.1.254	192.168.7.78	DNS	91	0		Standard query response 0x3011 AAAA www.jahfjrigjfjdivobofkjhkd.com
1106	11:33:30.910980	192.168.1.254	192.168.7.78	DNS	107	1		Standard query response 0x278c A www.jahfjrigjfjdivobofkjhkd.com A 143.244.220.150
1109	11:33:30.916396	192.168.1.254	192.168.7.78	DNS	107	1		Standard query response 0x1868 A www.jahfjrigjfjdivobofkjhkd.com A 143.244.220.150
1110	11:33:30.925329	192.168.1.254	192.168.7.78	DNS	91	0		Standard query response 0x7995 AAAA www.jahfjrigjfjdivobofkjhkd.com
1113	11:33:30.934451	192.168.1.254	192.168.7.78	DNS	91	0		Standard query response 0x6853 HTTPS www.jahfjrigjfjdivobofkjhkd.com
1153	11:33:32.189311	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0xe9cb AAAA www.jahfjrigjfjdivobofkjhkd.com
1154	11:33:32.189410	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0x5ff4 A www.jahfjrigjfjdivobofkjhkd.com
1165	11:33:32.209709	192.168.1.254	192.168.7.78	DNS	107	1		Standard query response 0x5ff4 A www.jahfjrigjfjdivobofkjhkd.com A 143.244.220.150
1184	11:33:32.258697	192.168.7.78	192.168.1.254	DNS	91	0		Standard query 0xf5e7 AAAA www.jahfjrigjfjdivobofkjhkd.com
1188	11:33:32.274803	192.168.1.254	192.168.7.78	DNS	91	0		Standard query response 0xe9cb AAAA www.jahfjrigjfjdivobofkjhkd.com
1225	11:33:32.351688	192.168.1.254	192.168.7.78	DNS	91	0		Standard query response 0xf5e7 AAAA www.jahfjrigjfjdivobofkjhkd.com

Frame 1105: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface \Device\NPF_{2684D732-66A2-456B-947A-FD28095E0172}
 Section number: 1
 ▾ Interface id: 0 (\Device\NPF_{2684D732-66A2-456B-947A-FD28095E0172})
 Interface name: \Device\NPF_{2684D732-66A2-456B-947A-FD28095E0172}
 Interface description: Wi-Fi
 Encapsulation type: Ethernet (1)
 Arrival Time: Jul 10, 2023 11:33:30.910980000 Central Daylight Time
 [Time shift for this packet: 0.000000000 seconds]
 Epoch Time: 1689006810.910980000 seconds
 [Time delta from previous captured frame: 0.005251000 seconds]
 [Time delta from previous displayed frame: 0.007423000 seconds]
 [Time since reference or first frame: 6.553082000 seconds]
 Frame Number: 1105
 Frame Length: 91 bytes (728 bits)
 Capture Length: 91 bytes (728 bits)
 [Frame is marked: False]
 [Frame is ignored: False]
 [Protocols in frame: eth:ethertype:ip:udp:dns]
 [Coloring Rule Name: UDP]
 [Coloring Rule String: udp]
 ▾ Ethernet II, Src: eero_36:d1:d2 (9c:57:bc:36:d1:d2), Dst: IntelCor_13:62:67 (bc:17:b8:13:62:
 ▾ Destination: IntelCor_13:62:67 (bc:17:b8:13:62:67)
 Address: IntelCor_13:62:67 (bc:17:b8:13:62:67)
0. = LG bit: Globally unique address (factory default)

0000	bc 17 b8 13 62 67 9c 57	bc 36 d1 d2 08 00 45 28	...bg·W ·6···E
0010	00 4d 74 48 00 00 3f 11	7c 93 c0 a8 01 fe c0 a8	·Mth··? ·
0020	07 4e 00 35 fc 84 00 39	22 d8 30 11 81 80 00 01	·N·5···9 "·0.....
0030	00 00 00 00 00 00 03 77	77 77 17 6a 61 68 66 6aw ww:jahfj
0040	72 69 67 6a 66 6a 64 69	76 6f 62 6f 66 6b 6a 68	rigjfjdi vobofkjh
0050	6b 64 03 63 6f 6d 00 00	1c 00 01	kd·com·

*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



ICMP Query via Wireshark

No.	Time	Source	Destination	Protocol	Length	Answer RRs	Respc	Info
4458	13:30:23.967785	fe80::9e57:bcff:fe36:d1d2	2600:1700:501:338f:f0ed:732a:e7c4:9899	ICMPv6	86			Neighbor Solicitation for 2600:1700:501:338f:f0ed:732a:e7c4:9899 from 9
4459	13:30:23.967818	2600:1700:501:338f:f0ed:732a:e7c4:9899	fe80::9e57:bcff:fe3...	ICMPv6	86			Neighbor Advertisement 2600:1700:501:338f:f0ed:732a:e7c4:9899 (sol, ovr)
4505	13:30:27.598809	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09...	ICMPv6	94			Echo (ping) request id=0x0001, seq=25, hop limit=128 (reply in 4506)
4506	13:30:27.624078	2607:f8b0:4002:c09::93	2600:1700:501:338f:f...	ICMPv6	94			Echo (ping) reply id=0x0001, seq=25, hop limit=56 (request in 4505)
4527	13:30:28.603481	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09...	ICMPv6	94			Echo (ping) request id=0x0001, seq=26, hop limit=128 (reply in 4528)
4528	13:30:28.626282	2607:f8b0:4002:c09::93	2600:1700:501:338f:f...	ICMPv6	94			Echo (ping) reply id=0x0001, seq=26, hop limit=56 (request in 4527)
4569	13:30:29.619621	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09...	ICMPv6	94			Echo (ping) request id=0x0001, seq=27, hop limit=128 (reply in 4570)
4570	13:30:29.647082	2607:f8b0:4002:c09::93	2600:1700:501:338f:f...	ICMPv6	94			Echo (ping) reply id=0x0001, seq=27, hop limit=56 (request in 4569)
4579	13:30:30.623553	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09...	ICMPv6	94			Echo (ping) request id=0x0001, seq=28, hop limit=128 (reply in 4580)
4580	13:30:30.651178	2607:f8b0:4002:c09::93	2600:1700:501:338f:f...	ICMPv6	94			Echo (ping) reply id=0x0001, seq=28, hop limit=56 (request in 4579)
8186	13:30:52.897989	fe80::ed88:583b:1487:a79	ff02::1	ICMPv6	86			Router Advertisement

> Frame 4527: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF
 > Ethernet II, Src: IntelCor_13:62:67 (bc:17:b8:13:62:67), Dst: eero_36:d1:d2 (9c:57:bc:36:d1:d2)
 > Internet Protocol Version 6, Src: 2600:1700:501:338f:f0ed:732a:e7c4:9899, Dst: 2607:f8b0:4002:
 > Internet Control Message Protocol v6

0000	9c 57 bc 36 d1 d2 bc 17 b8 13 62 67 86 dd 60 00	W 6 bg . . .
0010	00 00 00 28 3a 80 26 00 17 00 05 01 33 8f f0 ed	. . . (: & 3 . . .
0020	73 2a e7 c4 98 99 26 07 f8 b0 40 02 0c 09 00 00	s* . . . & . . @
0030	00 00 00 00 00 93 80 00 0f 81 00 01 00 1a 61 62 ab
0040	63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72	cdefghij klmnopqr
0050	73 74 75 76 77 61 62 63 64 65 66 67 68 69	stuvwxyz abc defghi

ICMP Packets Containing Ping Data

```
Command Prompt + ^

Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rayng>ping www.google.com

Pinging www.google.com [2607:f8b0:4002:c09::93] with 32 bytes of data:
Reply from 2607:f8b0:4002:c09::93: time=25ms
Reply from 2607:f8b0:4002:c09::93: time=22ms
Reply from 2607:f8b0:4002:c09::93: time=27ms
Reply from 2607:f8b0:4002:c09::93: time=27ms

Ping statistics for 2607:f8b0:4002:c09::93:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 22ms, Maximum = 27ms, Average = 25ms

C:\Users\rayng>
```

4505 13:30:27.598809	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09::93 ICMPv6	94	Echo (ping) request id=0x0001, seq=25, hop limit=128 (reply in 4506)
4506 13:30:27.624078	2607:f8b0:4002:c09::93	2600:1700:501:338f:: ICMPv6	94	Echo (ping) reply id=0x0001, seq=25, hop limit=56 (request in 4505)
4527 13:30:28.603481	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09::93 ICMPv6	94	Echo (ping) request id=0x0001, seq=26, hop limit=128 (reply in 4528)
4528 13:30:28.626282	2607:f8b0:4002:c09::93	2600:1700:501:338f:: ICMPv6	94	Echo (ping) reply id=0x0001, seq=26, hop limit=56 (request in 4527)
4569 13:30:29.619621	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09::93 ICMPv6	94	Echo (ping) request id=0x0001, seq=27, hop limit=128 (reply in 4570)
4570 13:30:29.647082	2607:f8b0:4002:c09::93	2600:1700:501:338f:: ICMPv6	94	Echo (ping) reply id=0x0001, seq=27, hop limit=56 (request in 4569)
4579 13:30:30.623553	2600:1700:501:338f:f0ed:732a:e7c4:9899	2607:f8b0:4002:c09::93 ICMPv6	94	Echo (ping) request id=0x0001, seq=28, hop limit=128 (reply in 4580)
4580 13:30:30.651178	2607:f8b0:4002:c09::93	2600:1700:501:338f:: ICMPv6	94	Echo (ping) reply id=0x0001, seq=28, hop limit=56 (request in 4579)



TLS 1.2 "Client Hello" Packet



No.	Time	Source	Destination	Protocol	Length	Answer RRs	Respc	Info
523	13:30:15.053722	192.168.7.78	13.107.5.80	TLSv1.2	571			Client Hello
535	13:30:15.066889	192.168.7.78	13.107.5.80	TLSv1.2	212			Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
536	13:30:15.067275	192.168.7.78	13.107.5.80	TLSv1.2	159			Application Data
537	13:30:15.067498	192.168.7.78	13.107.5.80	TLSv1.2	432			Application Data
551	13:30:15.079414	192.168.7.78	13.107.5.80	TLSv1.2	92			Application Data
585	13:30:15.129712	192.168.7.78	13.107.5.80	TLSv1.2	168			Application Data
6020	13:30:40.095638	192.168.7.78	13.226.204.21	TLSv1.3	571			Client Hello
6032	13:30:40.108495	192.168.7.78	13.226.204.21	TLSv1.3	118			Change Cipher Spec, Application Data
6033	13:30:40.108624	192.168.7.78	13.226.204.21	TLSv1.3	152			Application Data
6034	13:30:40.108702	192.168.7.78	13.226.204.21	TLSv1.3	459			Application Data
6042	13:30:40.113371	192.168.7.78	13.226.204.21	TLSv1.3	85			Application Data
6178	13:30:40.208973	192.168.7.78	13.86.58.123	TLSv1.3	571			Client Hello
6244	13:30:40.231814	192.168.7.78	13.86.58.123	TLSv1.3	134			Change Cipher Spec, Application Data
6245	13:30:40.232030	192.168.7.78	13.86.58.123	TLSv1.3	152			Application Data

Cipher Suites Length: 32

- ✓ Cipher Suites (16 suites)
 - Cipher Suite: Reserved (GREASE) (0xbaba)
 - Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
 - Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
 - Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
 - Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xccaa9)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xccaa8)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
 - Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
 - Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
 - Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
 - Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
 - Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)

Compression Methods Length: 1

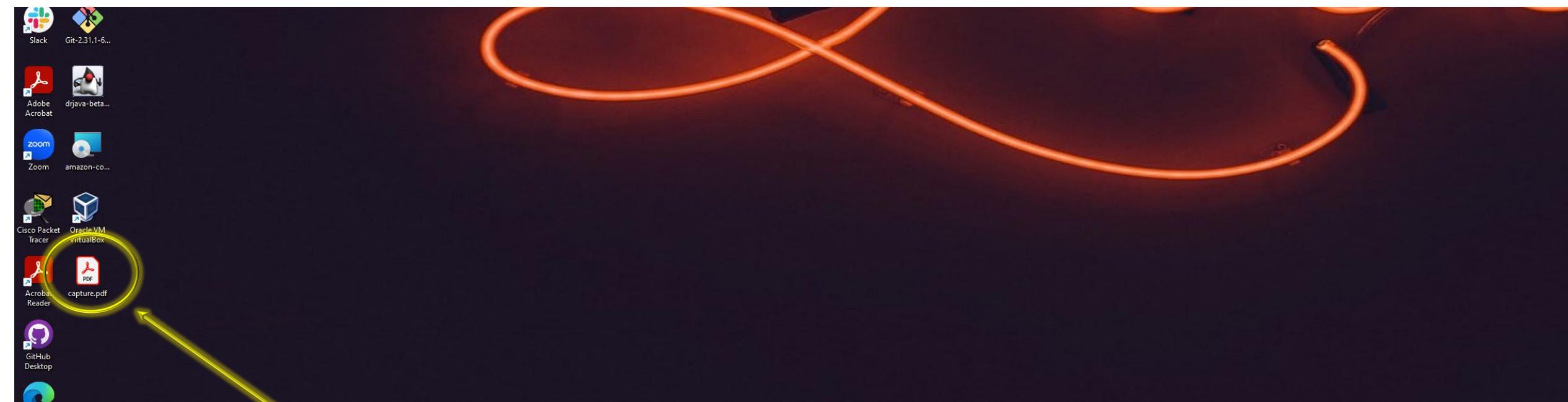
- > Compression Methods (1 method)

Extensions Length: 403

0000	9c 57 bc 36 d1 d2 bc 17 b8 13 62 67 08 00 45 00	·W·.....bg·E·
0010	02 2d 73 a2 40 00 80 06 00 00 c0 a8 07 4e 0d 6b	·-s·@.....N·k
0020	05 50 ef 3f 01 bb 06 03 6f 74 02 82 e4 22 50 18	·P·?.....ot··P·
0030	02 05 dc d0 00 00 16 03 01 02 00 01 00 01 fc 03
0040	03 d3 91 1e 0c 27 d7 00 65 25 0c ed 06 f5 d4 8f'·e%.....
0050	f6 7d be ea f8 0e 3b 58 e8 94 04 c1 fa 62 b9 49	·}....;X.....b·I
0060	65 20 ca 41 24 9d b7 1c 44 d9 e9 7d 64 83 9b c5	e·A\$.....D·}d.....
0070	3f 82 2c 1f 5e b2 c8 44 55 4e 61 ba cf 5a d5 74	?·,·^·D UNa·Z·t
0080	39 d5 00 20 ba ba 13 01 13 02 13 03 c0 2b c0 2f	9.....+·/.....
0090	c0 2c c0 30 cc a9 cc a8 c0 13 c0 14 00 9c 00 9d	·,·0.....
00a0	00 2f 00 35 01 00 01 93 8a 8a 00 00 00 17 00 00	·/5.....
00b0	00 10 00 0e 00 0c 02 68 32 08 68 74 74 70 2f 31h 2·http/1
00c0	2e 31 ff 01 00 01 00 00 0d 00 12 00 10 04 03 08	.1.....
00d0	04 04 01 05 03 08 05 05 01 08 06 06 01 00 0a 00
00e0	0a 00 08 da da 00 1d 00 17 00 18 00 05 00 05 01
00f0	00 00 00 00 00 23 00 00 00 12 00 00 00 2b 00 07#.....+
0100	06 3a 3a 03 04 03 03 44 69 00 05 00 03 02 68 32	:::·D i.....h2
0110	00 33 00 2b 00 29 da da 00 01 00 00 1d 00 20 9e	·3·+·).....
0120	75 b5 82 cb 9c 64 32 59 d8 76 9f 0a 1c 4d a5 97	u.....d2Y·v·M·
0130	ba 08 a0 be 2e 22 0f b2 e1 2a 77 60 46 8e 5d 00"·-*w`F·]
0140	2d 00 02 01 01 00 0b 00 02 01 00 00 00 00 15 00
0150	13 00 00 10 77 77 77 2e 62 69 6e 67 61 70 69 73www. bingapis
0160	2e 63 6f 6d 00 1b 00 03 02 00 02 1a 1a 00 01 00	.com.....
0170	00 15 00 c7 00 00 00 00 00 00 00 00 00 00 00 00 00



capture.pdf Saved on Desktop



“capture” file saved as .pdf
on my desktop



STOP THAT LEAKY TOILET!

Did you know a leaky toilet can waste up to **200 gallons** of water a day? That's a lot of wasted water and money flushed down the drain.

Some leaks are silent, some produce a running water sound and others may be visible as a small trickle running from the rim to the water in the bowl.

Simple steps to check for a toilet leak:

1: Start with a clean toilet free of any cleaning agents. Remove the lid to the tank.



2: Add a dye tablet, leak detector fluid or a few drops of food coloring to the tank.

3: Wait 10-15 minutes. Do not flush the toilet.

4: Look in the toilet bowl. If color has appeared in the bowl, you have a leak.

5: Flush as soon as the test is complete. Find and fix the leak or call a professional for assistance.



Southwest Florida
Water Management District