

Write-up CTF
SANS - Holiday Hack Challenge 2022
Challenge: Wireshark-Practice
Difficulty: Very Easy
<https://github.com/brunopelegreni>

First of all, open the Objectives session in the left menu.



For this challenge, we have the suspicious PCAP file to analyze. Download this file, then open it in the Wireshark.

Wireshark screenshot showing the 'suspicious(1).pcap' file. The packet list pane shows a large number of DNS requests from source IP 10.9.24.101 to destination IP 10.9.24.1. The details and bytes panes provide a detailed view of one specific DNS request, showing fields like 'Time', 'Source', 'Destination', 'Protocol', 'Length', and 'Info'. The packet list includes entries such as:

- 1 0.000000 10.9.24.101 > 10.9.24.1 DNS [REDACTED] Standard query 0x191e A adv.eposttoday.uk
- 2 0.364222 10.9.24.101 > 10.9.24.101 DNS [REDACTED] Standard query response 0x191e A adv.eposttoday.uk A 192.185.57.242
- 3 0.366183 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 4 0.366183 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 5 0.366061 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 6 0.397627 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 7 0.397704 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 8 0.868778 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 9 0.855969 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 10 1.025771 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 11 0.862854 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 12 1.568191 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 13 1.568574 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 14 1.568576 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 15 1.568893 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 16 2.258248 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 17 2.274269 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 18 2.514395 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 19 2.668439 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 20 3.068848 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
- 21 3.024821 10.9.24.101 > 10.9.24.101 TCP [REDACTED] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

@badposeidon1
<https://github.com/brunopelegreni>

In the Kringlecon game, go to Tokien Ring tunnels and search for Wireshark Phishing Terminal, then connect.



The Terminal will show two sessions. In the above session is where you'll insert your answers.
Then type “Yes” to continue.

```
This all started when I clicked on a link in my email.  
Can you help me? yes
```

Now the first question about challenge is shown, it refers to what type of object could be exported by Wireshark PCAP.

```
1. There are objects in the PCAP file that can be exported by Wireshark and/or Tshark. What type of objects can be exported from this PCAP?  
:
```

Task: Analyze the Wireshark file and Answer the Elf's Questions!
To complete your task, download the file from your badge or use this command line to answer the questions.

Tips:

1. Each question may have hints. If you want another hint from the elf, just type `hint` in the upper pane.
2. If you need help with Wireshark filters you can go here: <https://wiki.wireshark.org/DisplayFilters>
3. If you need help with tshark filters, try this cheat sheet: <https://cheatography.com/mwalke/r/cheat-sheets/tshark-wireshark-command-line/>
4. Of course, SANS has lots of cheat sheets that can help: <https://www.sans.org/blog/the-ultimate-list-of-sans-cheat-sheets/>
5. And remember, you can use Wireshark filters in tshark.

Tmux orientation:

For this terminal, you can use the mouse to switch or resize panes.

For clipboard use, you can `shift-click` and `drag`, then `Ctrl+c` to copy.

Use `Ctrl+Shift+v` to paste.

Normal tmux shortcuts (`Ctrl+b+t` or `!`) work as well.

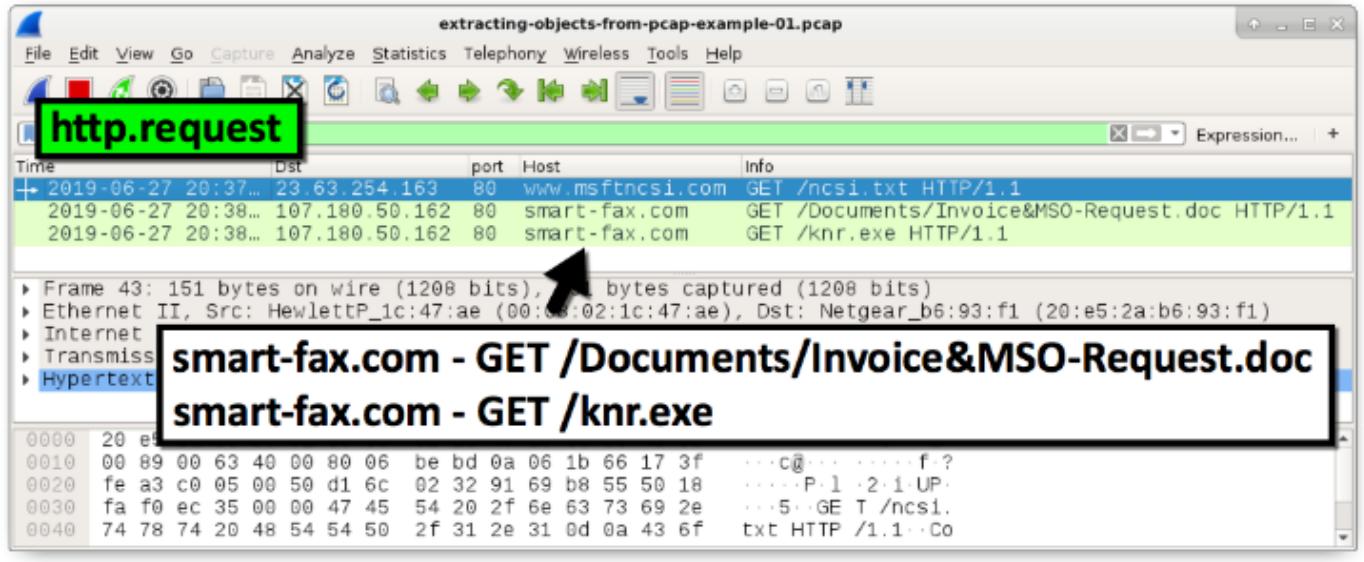
```
elf@3ff9e4eca436:~$
```

In this step, I searched in Google about “Extraction files in Wireshark”, how I can't find it, so I decided to ask for a hint for the terminal.

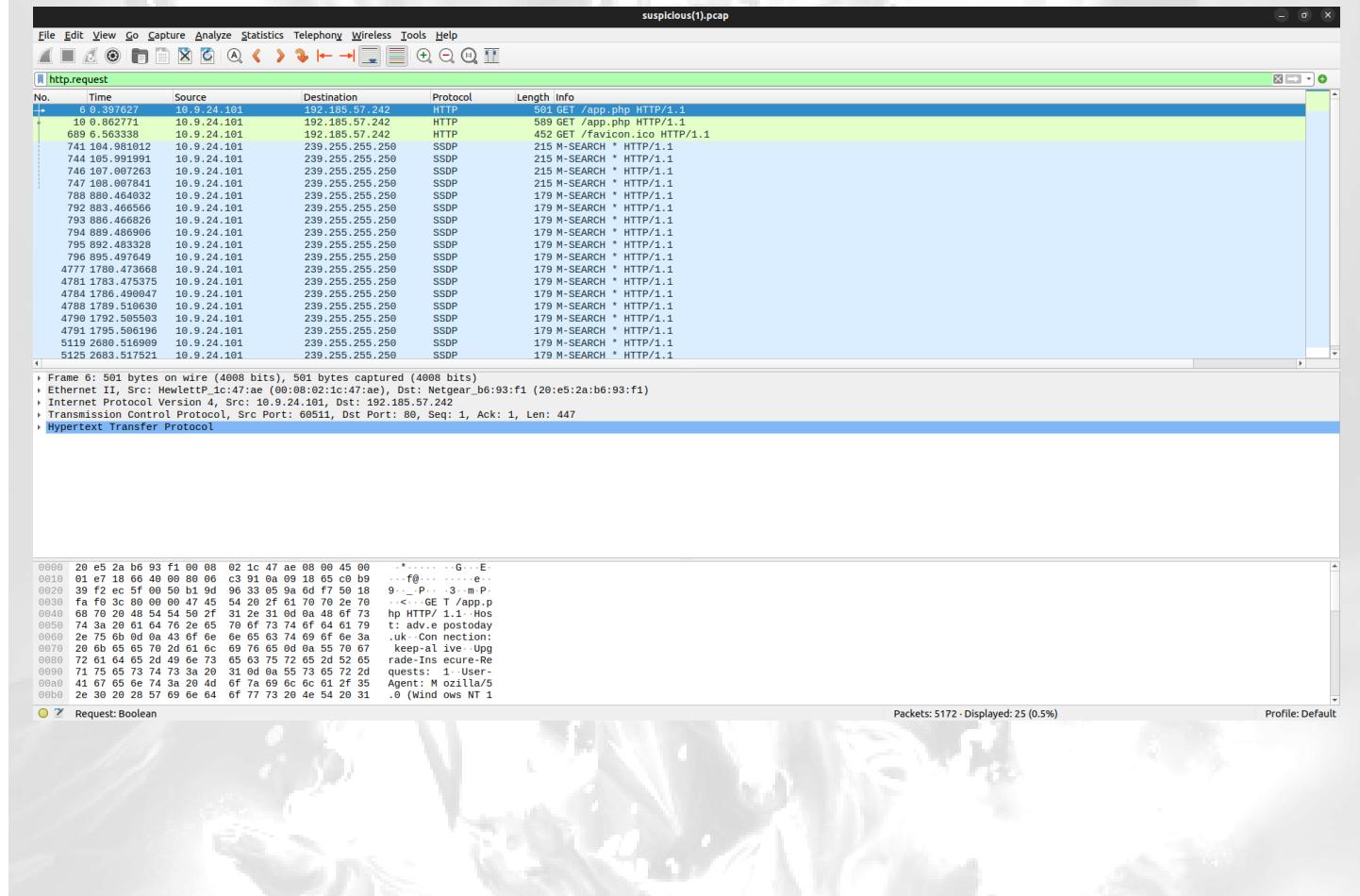
```
Hint 1: https://unit42.paloaltonetworks.com/using-wireshark-exporting-objects-from-a-pcap/
1. There are objects in the PCAP file that can be exported by Wireshark and/or Tshark. What type of objects can be exported from this PCAP?
: |
```

Task: Analyze the Wireshark file and Answer the Elf's Questions!
To complete your task, download the file from your badge or use this command line to answer the questions.
Tips:
1. Each question may have hints. If you want another hint from the elf, just type `hint` in the upper pane.
2. If you need help with Wireshark filters you can go here: <https://wiki.wireshark.org/DisplayFilters>
3. If you need help with tshark filters, try this cheat sheet: <https://cheatography.com/mbwaller/cheat-sheets/tshark-wireshark-command-line/>
4. Of course, SANS has lots of cheat sheets that can help: <https://www.sans.org/blog/the-ultimate-list-of-sans-cheat-sheets/>
5. And remember, you can use Wireshark filters in tshark.
Tmux orientation:
For this terminal, you can use the mouse to switch or resize panes.
For clipboard use, you can `shift+click` and `drag`, then `Ctrl+c` to copy.
Use `Ctrl+Shift+v` to paste.
Normal tmux shortcuts (`Ctrl+b+t` or `!`) work as well.
elf@3ff9e4eca436:~\$

It shows a [link](#) for reference. In this reference we have a specific session about HTTP requests, then I decided to try it!



Now, I filtered by HTTP requests, and now I have only 3 lines about HTTP requests.



Following the reference available, I tried exporting the HTTP request.

Notice, in the window opened we have two files, the app.php and favicon.ico, but the object showing in the bar is HTTP object list. I saved the first file in my PC to after analyzing.

The screenshot shows the 'Wireshark · Export · HTTP object list' dialog. At the top, there is a 'Text Filter:' input field and a 'Content Type:' dropdown set to 'All Content-Types'. Below is a table with the following data:

Packet	Hostname	Content Type	Size	Filename
8	adv.epostoday.uk	text/html	754 bytes	app.php
687	adv.epostoday.uk	text/html	808 kB	app.php
692	adv.epostoday.uk	text/html	1.130 bytes	favicon.ico

At the bottom, there are buttons for '? Help', 'Preview', 'Save All', 'Close', and 'Save'.

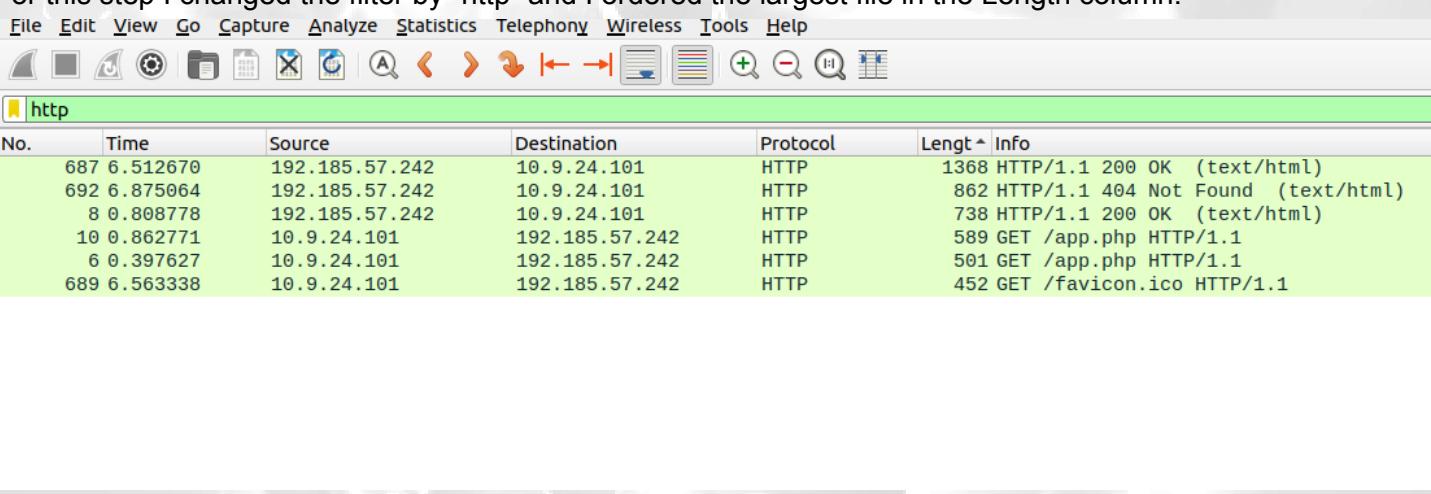
In Terminal, I tried to type “HTTP”, because the object exported for me it's of type HTTP.

```
Hint 1: https://unit42.paloaltonetworks.com/using-wireshark-exporting-objects-from-a-pcap/
1. There are objects in the PCAP file that can be exported by Wireshark and/or Tshark. What
type of objects can be exported from this PCAP?
: HTTP
```

Now the terminal is asking for the largest file we can export.

```
2. What is the file name of the largest file we can export?  
: app.php
```

For this step I changed the filter by "http" and I ordered the largest file in the Length column.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

HTTP

No.	Time	Source	Destination	Protocol	Length	Info
687	6.512670	192.185.57.242	10.9.24.101	HTTP	1368	HTTP/1.1 200 OK (text/html)
692	6.875064	192.185.57.242	10.9.24.101	HTTP	862	HTTP/1.1 404 Not Found (text/html)
8	0.808778	192.185.57.242	10.9.24.101	HTTP	738	HTTP/1.1 200 OK (text/html)
10	0.862771	10.9.24.101	192.185.57.242	HTTP	589	GET /app.php HTTP/1.1
6	0.397627	10.9.24.101	192.185.57.242	HTTP	501	GET /app.php HTTP/1.1
689	6.563338	10.9.24.101	192.185.57.242	HTTP	452	GET /favicon.ico HTTP/1.1

In the terminal I type app.php, which references the largest package.

```
2. What is the file name of the largest file we can export?  
: app.php
```

Now, we have question number 3, what number packet it refers to.

```
3. What packet number starts that app.php file?
```

```
: 687
```

Look for the first packet ordained.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



http

No.	Time	Source	Destination	Protocol	Length	Info
687	6.512670	192.185.57.242	10.9.24.101	HTTP	1368	HTTP/1.1 200 OK (text/html)
692	6.875064	192.185.57.242	10.9.24.101	HTTP	862	HTTP/1.1 404 Not Found (text/html)
8	0.808778	192.185.57.242	10.9.24.101	HTTP	738	HTTP/1.1 200 OK (text/html)
10	0.862771	10.9.24.101	192.185.57.242	HTTP	589	GET /app.php HTTP/1.1
6	0.397627	10.9.24.101	192.185.57.242	HTTP	501	GET /app.php HTTP/1.1
689	6.563338	10.9.24.101	192.185.57.242	HTTP	452	GET /favicon.ico HTTP/1.1

In the terminal type 687 and press enter.

```
3. What packet number starts that app.php file?
```

```
: 687
```

Now, we have question number 4.

```
4. What is the IP of the Apache server?  
: [REDACTED]
```

Look for the same line, the IP Source to find the required IP.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



No.	Time	Source	Destination	Protocol	Length	Info
687	6.512670	192.185.57.242	10.9.24.101	HTTP	1368	HTTP/1.1 200 OK (text/html)
692	6.875064	192.185.57.242	10.9.24.101	HTTP	862	HTTP/1.1 404 Not Found (text/html)
8	0.808778	192.185.57.242	10.9.24.101	HTTP	738	HTTP/1.1 200 OK (text/html)
10	0.862771	10.9.24.101	192.185.57.242	HTTP	589	GET /app.php HTTP/1.1
6	0.397627	10.9.24.101	192.185.57.242	HTTP	501	GET /app.php HTTP/1.1
689	6.563338	10.9.24.101	192.185.57.242	HTTP	452	GET /favicon.ico HTTP/1.1

So, type 192.185.57.242 and press Enter.

```
4. What is the IP of the Apache server?  
: 192.185.57.242 [REDACTED]
```

We have question number 5.

5. What file is saved to the infected host?

:

Select the line refer to app.php request, then right click, click Follow and HTTP Stream.

The screenshot shows the Wireshark interface with a list of network packets. Packet 6, which is the GET /app.php request, is selected. A context menu is open over this packet, with the 'Follow' submenu expanded. The 'HTTP Stream' option in this submenu is highlighted. The packet details and bytes panes are visible at the bottom.

No.	Time	Source	Destination	Protocol	Length	Info
687	6.512670	192.185.57.242	10.9.24.101	HTTP	1368	HTTP/1.1 200 OK (text/html)
692	6.875064	192.185.57.242	10.9.24.101	HTTP	862	HTTP/1.1 404 Not Found (text/html)
8	0.808778	192.185.57.242	10.9.24.101	HTTP	738	HTTP/1.1 200 OK (text/html)
10	0.862771	10.9.24.101	192.185.57.242	HTTP	589	GET /app.php HTTP/1.1
6	0.397627	10.9.24.101	192.185.57.242	HTTP	1	app.php HTTP/1.1
689	6.563338	10.9.24.101	192.185.57.242	HTTP	1	favicon.ico HTTP/1.1

[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80]
Ethernet II, Src: Hewlett_P_1c:47:ae (00:08:02:1c:47:ae), Dst: 00:0c:29:00:00:00 (00:0c:29:00:00:00)
Internet Protocol Version 4, Src: 10.9.24.101, Dst: 192.185.57.242
Transmission Control Protocol, Src Port: 60511, Dst Port: 80
Source Port: 60511
Destination Port: 80
[Stream index: 0]
[Conversation completeness: Complete, WITH_DATA]
[TCP Segment Len: 447]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2979894835
[Next Sequence Number: 448 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 94006775
0101 = Header Length: 20 bytes (5)
Flags: 0x0018 (PSH, ACK)
Window: 64240
[Calculated window size: 64240]
[Window size scaling factor: -2 (no window scaling used)]
Checksum: 0x3c80 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (447 bytes)
Hypertext Transfer Protocol
GET /app.php HTTP/1.1\r\n

Hex	Dec	Text
0000	20 e5 2a b6 93 f1 00 08 02 1c 47 ae 08 00 45 00	.*..... .G...E.
0010	01 e7 18 66 40 00 80 06 c3 91 0a 09 18 65 c0 b9	...f@....e..
0020	39 f2 ec 5f 00 50 b1 9d 96 33 05 9a 6d f7 50 18	9..._P... 3..m.P.
0030	fa f0 3c 80 00 00 47 45 54 20 2f 61 70 70 2e 70	..<...GE T /app.p
0040	68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73	hp HTTP/ 1.1. Hos
0050	74 3a 20 61 64 76 2e 65 70 6f 73 74 6f 64 61 79	t: adv.e postday
0060	2e 75 6b 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a	.uk. Con nection:
0070	20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 55 70 67	keep-alive. Upg
0080	72 61 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65	rade-Ins ecure-Re
0090	71 75 65 73 74 73 3a 20 31 0d 0a 55 73 65 72 2d	quests: 1. User-
00a0	41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35	Agent: Mozilla/5

The window Follow is opened, searching anything related to this, I found this stranger block of code.

Looking a little bit, I found the file .zip.

```
Wireshark · Follow HTTP Stream (tcp.stream eq 0) · suspicious(1).pcap
```

```
gFdKZ/
tnkqbSciK7cB8gsioLTGoNhmlvlBzPfy0FFoh0kB6ftxJnjquej3zWNFEkDziSjVz36fwHNAnejegA7MM89+a8hBPYJyT9WNFepGy
Iib0cznafjZC79BQdFRj5rZ8LEqUeXWRcvPMm4cJf4332JjvCDRGjOU90ADx60QTeN1UxbQzJhvVgHeHYUMBtQHJV8A7t6bMx7gM
i/
rrPqxIhBSJHsFHli1PrzwT7nc9SAyefwlFYkDs9ISdEZSrpw33bsoSvWmu0UnK8heFcNrwB3brroqD9CEUNBkt+WHYB5RejwHqdk
b+I+j/zoezIXFcq9FKnGjMTeeCX7KZzIYsCFBMr6IXmtESXN40z8P/EbyvEydsJgNazDAlsR5LFCx9rJGLPzB/
3KmTPYAoBH8BD0FW/dld5u3cCEztORTse0m02PEPASTNQUpTk2N1helqjPwg/
dmObbHwec30ILK7qwDTAtLrgg9T0ICYpLT5zE74ukSxcEJafhYm3YFN6lULGo/
qRekA5sRapi2HyPvDTUKzmuejfZZZz7ud87EPJcL1epbVC9z2jEUokrTMRej5owSi5K2PukUAabGZiQ4jXmj67vTxGy7L6AdlyQbv
ujWFaL/wnCBfV/zvIlXpQem70lR8KiXy3w0bc/hVFAbkjy/
aY1TC8dWE9qBSGR+jktokyYRJ9agFBE2jOrWUjIYgDWi4cvQBqjce4+lAWRwmQ4BJkN7cAzzJlBl2ot9AZwen8GZ/
tGgbhWhKN984dqf5fg7W3/Y6hGeZ2MyeQ1qhopyYOC2a0J5NclRIxsg6NMnBq85hjZwkw0fpmaUUuUWQ9J3KJP/
f110cFdL6xfWsUHT60TdEW/XJkts0n9bweQLOTIUohtgDVUpVvhphRcAzdGLOKny7d7X30eucJw3ZRpezS9jzEA5dcHeX6k3b/
RgtWzXCNk2A23NvvNtW8P0AAAAAAAAAAHV432BVddro/2vtRn59+i/
EPxsAQAcAUeSBAh8AFAAAAAgAUKM3UUirsfh70wkAg3oLABMAJAAAAAAAAAAgAAAAAAAAAFJlZl9TZXB0MjQtMjAyMC5zY3IKACAA
AAAAAAEAGABgQXRy35HWAQVnJmJUktYBBWcmYlSS1gFQSwUGAAAAAEAAQBlAAAArDsJAAA'');

let byteNumbers = new Array(byteCharacters.length);
for (let i = 0; i < byteCharacters.length; i++) {
    byteNumbers[i] = byteCharacters.charCodeAt(i);
}
let byteArray = new Uint8Array(byteNumbers);

// now that we have the byte array, construct the blob from it
let blob1 = new Blob([byteArray], {type: 'application/octet-stream'});

saveAs(blob1, 'Ref_Sept24-2020.zip');

})();

</script>
</body>GET /favicon.ico HTTP/1.1
Host: adv.eposttoday.uk
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36 Edg/85.0.564.63
Accept: image/webp,image/apng,image/*,*/*;q=0.8
Referer: http://adv.eposttoday.uk/app.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: d=0; n=UTFC

HTTP/1.1 404 Not Found
Date: Sat, 24 Dec 2022 17:31:12 GMT
Server: Apache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 566
Keep-Alive: timeout=5, max=73
Connection: Keep-Alive
Packet 687. 3 client pkts, 3 server pkts, 5 turns. Click to select.
```

Entire conversation (812 kB) Show data as ASCII

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

Type Ref_Sept24-2020.zip in the terminal and press Enter.

```
5. What file is saved to the infected host?  
: Ref_Sept24-2020.zip
```

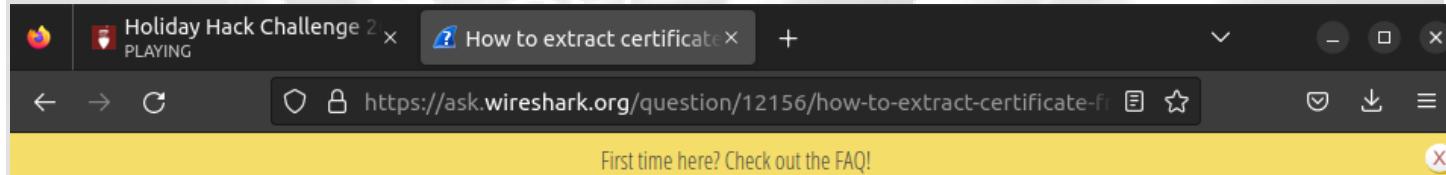
Task: Analyze the Wireshark file and Answer the Elf's Questions!

We have question number 6.

```
6. Attackers used bad TLS certificates in this traffic. Which countries were they registered to? Submit the names of the countries in alphabetical order separated by a commas (Ex: Norway, South Korea).  
:
```

Task: Analyze the Wireshark file and Answer the Elf's Questions!

I made a search in Google about "How to extract bad certificates Wireshark", then I found the filter below.



2 Answers

Sort by » | oldest | newest | most voted

You're looking at the wrong TLS record.

0

You need to look at the TLS handshake record that sends the server certificate.

Use the display filter `tls.handshake.type == 11` to find certificate records.

answered Oct 9 '19
grahamb
23665 4 888 227
<https://www.wireshark.org>

Note that 3.0.5 is the current stable release version of Wireshark.

Comments

link

Using the filter, I found some companies that use bad TLS certificates. But I have two that get my attention, because they use a atypical location.

- Nagoya - IL (Israel)
- Aquarelle - SS (South Sudan)

The screenshot shows two certificate details in Wireshark. The first is for Nagoya, Israel, with a length of 1068 bytes. It includes fields like signedCertificate, serialNumber, signature, issuer, rdnSequence, and RDNSequence items. The second is for Aquarelle, South Sudan, with a length of 1068 bytes, also showing similar structure with fields like signedCertificate, serialNumber, signature, issuer, rdnSequence, and RDNSequence items.

```
Handshake Type: Certificate (11)
Length: 1071
Certificates Length: 1068
  Certificates (1068 bytes)
    Certificate Length: 1065
    Certificate: 308204253082030da003020102020900c498012488156a13300d06092a864886f70d0101... (id-at-commonName=heardbellith.Icanwepeh.nagoya,id-at-organizationalUnitName=moasn@emanc,id-at-organizationName=)
      signedCertificate
        version: v3 (2)
        serialNumber: 0x00c498012488156a13
      signature (sha256WithRSAEncryption)
        Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 6 items (id-at-commonName=heardbellith.Icanwepeh.nagoya,id-at-organizationalUnitName=moasn@emanc,id-at-organizationName=Wemadd Hixchac GmbH,id-at-localityName=Tel Aviv,id-at-stateOrProvinceName=Anourd Thiolaved Thersile5 Fteda8)
          RDNSequence item: 1 item (id-at-countryName=IL)
            RelativeDistinguishedName item (id-at-countryName=IL)
              Id: 2.5.4.6 (id-at-countryName)
              CountryName: IL
          RDNSequence item: 1 item (id-at-stateOrProvinceName=Anourd Thiolaved Thersile5 Fteda8)
            RelativeDistinguishedName item (id-at-stateOrProvinceName)
              Id: 2.5.4.8 (id-at-stateOrProvinceName)
            DirectoryString: UTF8String (4)
              UTF8String: Anourd Thiolaved Thersile5 Fteda8
          RDNSequence item: 1 item (id-at-localityName=Tel Aviv)
        RDNSequence item: 1 item (id-at-localityName=Tel Aviv)

  Handshake Type: Certificate (11)
Length: 1071
Certificates Length: 1068
  Certificates (1068 bytes)
    Certificate Length: 1065
    Certificate: 308204253082030da003020102020900c498012488156a13300d06092a864886f70d0101... (id-at-commonName=psprponoum.aquarelle,id-at-organizationName=Hedanpr S.p.a.,id-at-localityName=Khartoum,id-at-countryName=SS)
      signedCertificate
        version: v3 (2)
        serialNumber: 0x00c498012488156a13
      signature (sha256WithRSAEncryption)
        Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
      issuer: rdnSequence (0)
        rdnSequence: 4 items (id-at-commonName=psprponoum.aquarelle,id-at-organizationName=Hedanpr S.p.a.,id-at-localityName=Khartoum,id-at-countryName=SS)
          RDNSequence item: 1 item (id-at-countryName=SS)
            RelativeDistinguishedName item (id-at-countryName=SS)
              Id: 2.5.4.6 (id-at-countryName)
              CountryName: SS
          RDNSequence item: 1 item (id-at-localityName=Khartoum)
            RelativeDistinguishedName item (id-at-localityName=Khartoum)
```

So, I put it in alphabetic order and send it to the terminal. Type “Israel, South Sudan”.

```
6. Attackers used bad TLS certificates in this traffic. Which countries were they registered to? Submit the names of the countries in alphabetical order separated by a commas (Ex: Norway, South Korea).
: Israel, South Sudan
```

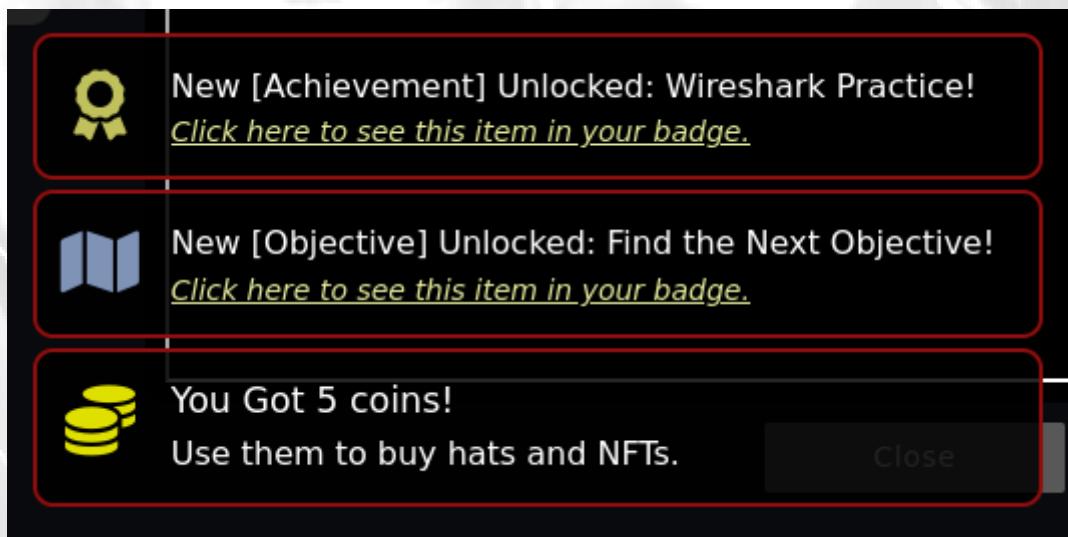
Now the terminal asks me if the endpoint is compromised.

```
7. Is the host infected (Yes/No) ?
: 
```

In my opinion, yes! So I type it the terminal!

```
7. Is the host infected (Yes/No) ?  
: yes
```

The challenge is done!



Now share it with your friends!

Congratulations! You have completed
the Wireshark Practice challenge!  [Tweet This!](#)