

# Day 7 - The Grinch Really Did Steal Christmas

## Scenario

### Story

It's 6 AM and Elf McSkidy is clocking-in to The Best Festival Company's SOC headquarters to begin his watch over TBFC's infrastructure. After logging in, Elf McEager proceeds to read through emails left by Elf McSkidy during the nightshift.

More automatic scanning alerts, oh look, another APT group. It feels like it's going to be a long, but easy start to the week for Elf McEager.

Whilst clearing the backlog of emails, Elf McEager reads the following: "**URGENT:** Data exfiltration detected on TBFC-WEB-01". "*Uh oh*" goes Elf McEager. "*TBFC-WEB-01? That's Santa's webserver! Who has the motive to steal data from there?!*". It's time for the ever-vigilant Elf McEager to prove his salt and find out exactly what happened.

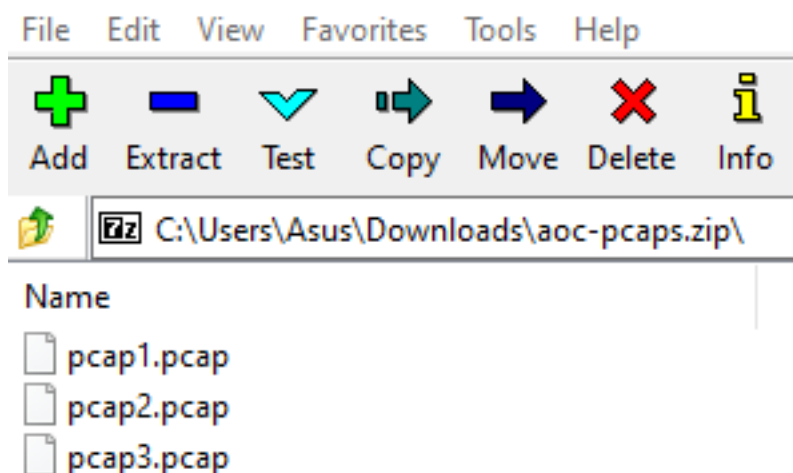
Unknowingly to Elf McEager, Elf McSkidy made this all up! Fortunately, this isn't a real attack - but a training exercise created ahead of Elf McEager's performance review.

### Challenge

Download the ZIP file "aocpcaps.zip" that is attached to this task, use a combination of the filters and features of Wireshark we've covered to answer the questions below:

download the pcap zip file, there's 3 pcap files in it

 C:\Users\Asus\Downloads\aoc-pcaps.zip\



now let's open pcap1 file using wireshark

pcap1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination
1	0.000000	10.10.15.52	10.11.3.2
2	0.000082	10.10.15.52	10.11.3.2
3	0.000155	10.10.15.52	10.11.3.2
4	0.033155	10.11.3.2	10.10.15.52
5	0.033167	10.11.3.2	10.10.15.52
6	2.507709	10.10.15.52	91.189.88.184
7	2.507792	10.10.15.52	91.189.88.185

let's search for the IP that initiate ICMP/ping

//here we can see that 10.11.3.2 are the IP that initiate ICMP/ping to the destination 10.10.15.52

icmp

No.	Time	Source	Destination	Protocol	L
17	10.430447	10.11.3.2	10.10.15.52	ICMP	
18	10.430472	10.10.15.52	10.11.3.2	ICMP	
19	11.428953	10.11.3.2	10.10.15.52	ICMP	
20	11.428977	10.10.15.52	10.11.3.2	ICMP	
21	12.432844	10.11.3.2	10.10.15.52	ICMP	
22	12.432870	10.10.15.52	10.11.3.2	ICMP	
23	13.433469	10.11.3.2	10.10.15.52	ICMP	
24	13.433495	10.10.15.52	10.11.3.2	ICMP	

Question: Open "pcap1.pcap" in Wireshark. What is the IP address that initiates an ICMP/ping?  
-10.11.3.2

now let's check for any GET request made in http protocol

http.request.method == GET						
No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP/1.1
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.min.js HTTP/1.1
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	GET /fonts/ noto-sans-jp-v25-japanese_latin-regula
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.
202	62.708297	10.10.67.199	10.10.15.52	HTTP	315	GET /favicon.ico HTTP/1.1
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP/1.1
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1

Question: If we only wanted to see HTTP GET requests in our "pcap1.pcap" file, what filter would we use?  
 -http.request.method == GET

now we need to find the article that IP '10.10.67.199' visited, we can append this command into it

http.request.method == GET && ip.addr == 10.10.67.199						
No.	Time	Source	Destination	Protocol	Length	Info
67	62.185886	10.10.67.199	10.10.15.52	HTTP	394	GET / HTTP/1.1
71	62.478663	10.10.67.199	10.10.15.52	HTTP	363	GET /fontawesome/css/all.min.css HTTP/1.1
75	62.479630	10.10.67.199	10.10.15.52	HTTP	348	GET /css/dark.css HTTP/1.1
83	62.480991	10.10.67.199	10.10.15.52	HTTP	333	GET /js/bundle.js HTTP/1.1
85	62.481045	10.10.67.199	10.10.15.52	HTTP	342	GET /js/instantpage.min.js HTTP/1.1
95	62.487106	10.10.67.199	10.10.15.52	HTTP	347	GET /images/icon.png HTTP/1.1
105	62.516878	10.10.67.199	10.10.15.52	HTTP	336	GET /post/index.json HTTP/1.1
107	62.530696	10.10.67.199	10.10.15.52	HTTP	430	GET /fonts/ noto-sans-jp-v25-japanese_latin-regular.wof
108	62.532591	10.10.67.199	10.10.15.52	HTTP	445	GET /fontawesome/webfonts/fa-solid-900.woff2 HTTP/1.1
117	62.540748	10.10.67.199	10.10.15.52	HTTP	415	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
202	62.708297	10.10.67.199	10.10.15.52	HTTP	315	GET /favicon.ico HTTP/1.1
295	63.665611	10.10.67.199	10.10.15.52	HTTP	445	GET / HTTP/1.1
299	63.694780	10.10.67.199	10.10.15.52	HTTP	414	GET /fontawesome/css/all.min.css HTTP/1.1
303	63.695898	10.10.67.199	10.10.15.52	HTTP	399	GET /css/dark.css HTTP/1.1
315	63.697840	10.10.67.199	10.10.15.52	HTTP	384	GET /js/bundle.js HTTP/1.1

> Frame 67: 394 bytes on wire (3152 bits), 394 bytes captured (3152 bits)

so this is the article that the IP 10.10.67.199 visited

467	64.028410	10.10.67.199	10.10.15.52	HTTP	466	GET /fonts/roboto-v20-latin-regular.woff2 HTTP/1.1
471	64.222360	10.10.67.199	10.10.15.52	HTTP	365	GET /posts/reindeer-of-the-week/ HTTP/1.1
475	66.239846	10.10.67.199	10.10.15.52	HTTP	369	GET /posts/post/index.json HTTP/1.1

Question: Now apply this filter to "pcap1.pcap" in Wireshark, what is the name of the article that the IP address "10.10.67.199" visited?  
 -reindeer-of-the-week

now let's analyze the pcap2 file using wireshark, we need to look for the FTP traffic & find the plaintext passphrase



Apply a display filter ... &lt;Ctrl-/&gt;

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.122.128	10.11.3.2	SSH	102	Server:
2	0.000084	10.10.122.128	10.11.3.2	SSH	150	Server:
3	0.060016	10.11.3.2	10.10.122.128	TCP	54	57748 →
4	0.101317	10.11.3.2	10.10.122.128	TCP	54	57748 →
5	1.127866	10.10.122.128	91.189.92.40	TCP	74	33400 →
6	2.549894	10.10.73.252	10.10.122.128	FTP	72	Request:
7	2.549999	10.10.122.128	10.10.73.252	FTP	80	Response
8	2.550011	10.10.122.128	10.10.73.252	TCP	66	21 → 453
9	2.555520	10.10.73.252	10.10.122.128	TCP	66	45332 →
10	2.555529	10.10.73.252	10.10.122.128	TCP	66	45332 →
11	2.555534	10.10.122.128	10.10.73.252	TCP	66	21 → 453
12	3.175873	10.10.122.128	91.189.92.40	TCP	74	33402 →

to filter the FTP traffic & get the PASS command only let's apply the following filter



tcp.port == 21 &amp;&amp; ftp.request.command == PASS

No.	Time	Source	Destination	Protocol	L
28	14.282063	10.10.73.252	10.10.122.128	FTP	

the passphrase that used would be

```

File Transfer Protocol (FTP)
  > PASS plaintext_password_fiasco\r\n
  [Current working directory: ]

```

Question: Let's begin analysing "pcap2.pcap". Look at the captured FTP traffic; what password was leaked during the login process?

-plaintext\_password\_fiasco

now let's find all the services that have in this pcap file, we can list it by using 'statistic > protocol hierarchy' //so the encrypted protocol would be the SSH protocol in this case

Wireshark · Protocol Hierarchy Statistics · pcap2.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets
▼ Frame	100.0	239	100.0	31642	3697	0
▼ Ethernet	100.0	239	10.6	3346	391	0
▼ Internet Protocol Version 4	96.7	231	14.6	4620	539	0
▼ Transmission Control Protocol	93.3	223	72.9	23076	2696	119
SSH Protocol	26.8	64	48.4	15302	1788	64
▼ FTP Data	1.3	3	1.4	438	51	0
Line-based text data	1.3	3	1.4	438	51	3
File Transfer Protocol (FTP)	15.5	37	2.9	928	108	37
Internet Control Message Protocol	3.3	8	1.0	320	37	8
Address Resolution Protocol	3.3	8	0.9	280	32	8

Question: Continuing with our analysis of "pcap2.pcap", what is the name of the protocol that is encrypted?  
-SSH

now let's analyze the pcap3 using wireshark again

pcap3.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.53.219	10.11.3.2	SSH	166	Ser
2	0.056471	10.11.3.2	10.10.53.219	TCP	54	603
3	1.329098	10.11.3.2	10.10.53.219	SSH	118	Cli
4	1.329467	10.10.53.219	10.11.3.2	SSH	102	Ser
5	1.388402	10.11.3.2	10.10.53.219	TCP	54	603
6	1.491913	10.11.3.2	10.10.53.219	SSH	102	Cli
7	1.492209	10.10.53.219	10.11.3.2	SSH	102	Ser
8	1.548870	10.11.3.2	10.10.53.219	TCP	54	603
9	1.634277	10.11.3.2	10.10.53.219	SSH	102	Cli
10	1.634594	10.10.53.219	10.11.3.2	SSH	102	Ser
11	1.695854	10.11.3.2	10.10.53.219	TCP	54	603
12	1.748455	10.11.3.2	10.10.53.219	SSH	102	Cli
13	1.748788	10.10.53.219	10.11.3.2	SSH	102	Ser
14	1.804690	10.11.3.2	10.10.53.219	TCP	54	603

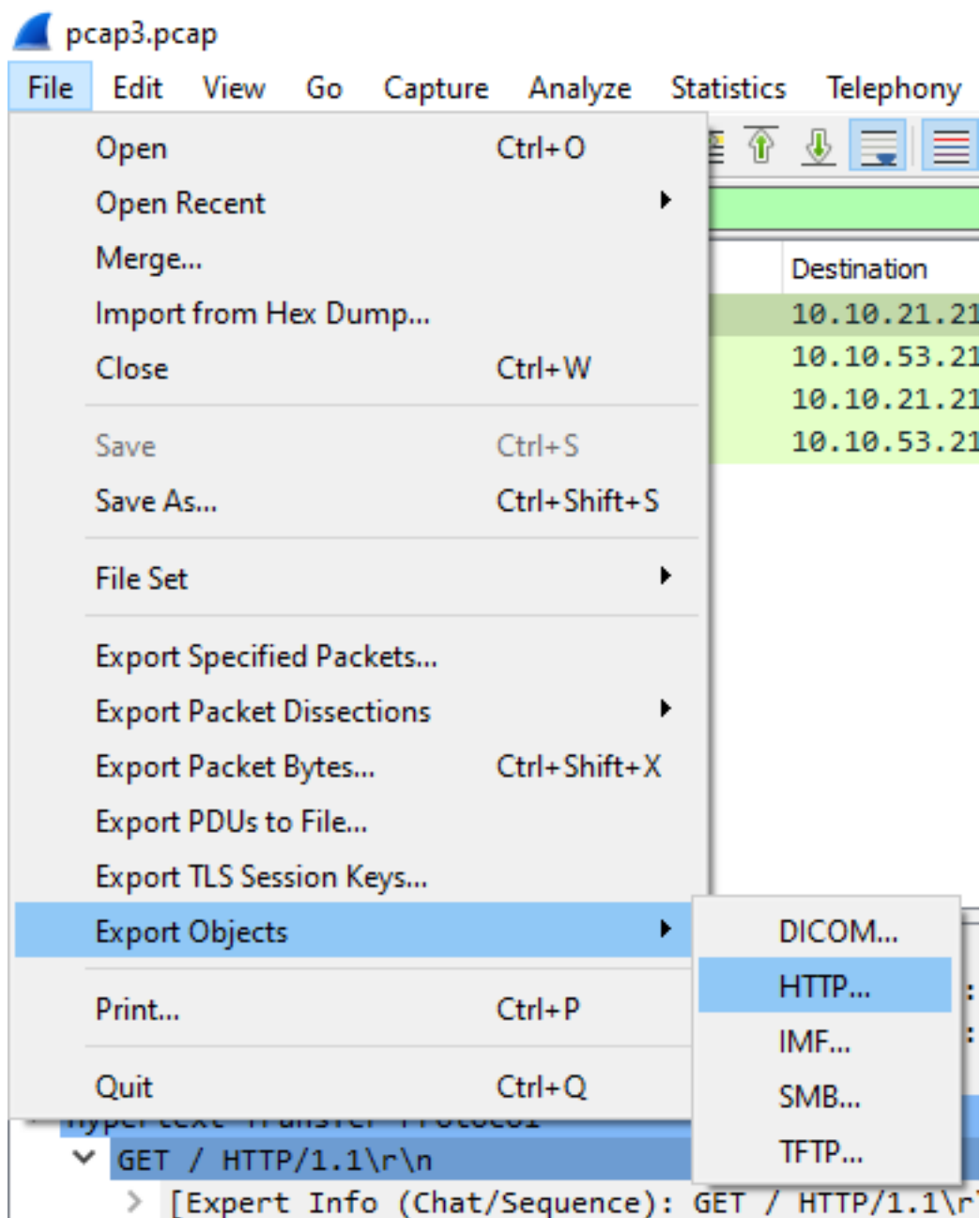
let's check the services that's available in this pcap file first, to find the interesting protocol that might shows Elf McSkidy's wishlist

Protocol	Percent Packets	P
▼ Frame	100.0	40
▼ Ethernet	100.0	40
▼ Internet Protocol Version 4	99.5	40
▼ Transmission Control Protocol	99.5	40
SSH Protocol	46.8	19
▼ Hypertext Transfer Protocol	1.0	4
Media Type	0.2	1
Line-based text data	0.2	1
Address Resolution Protocol	0.5	2

the HTTP protocol are not encrypted, so let's check out the HTTP protocol  
 //okay downloading christmas.zip that one looks interesting here

http						
No.	Time	Source	Destination	Protocol	Length	Info
→ 166	11.665107	10.10.53.219	10.10.21.210	HTTP	139	GET / HTTP/1.1
← 168	11.665723	10.10.21.210	10.10.53.219	HTTP	4852	HTTP/1.1 200 OK (text/html)
291	26.537049	10.10.53.219	10.10.21.210	HTTP	215	GET /christmas.zip HTTP/1.1
395	26.542475	10.10.21.210	10.10.53.219	HTTP	10388	HTTP/1.1 200 OK (application/zip)

let's export the christmas.zip file out from this pcap file

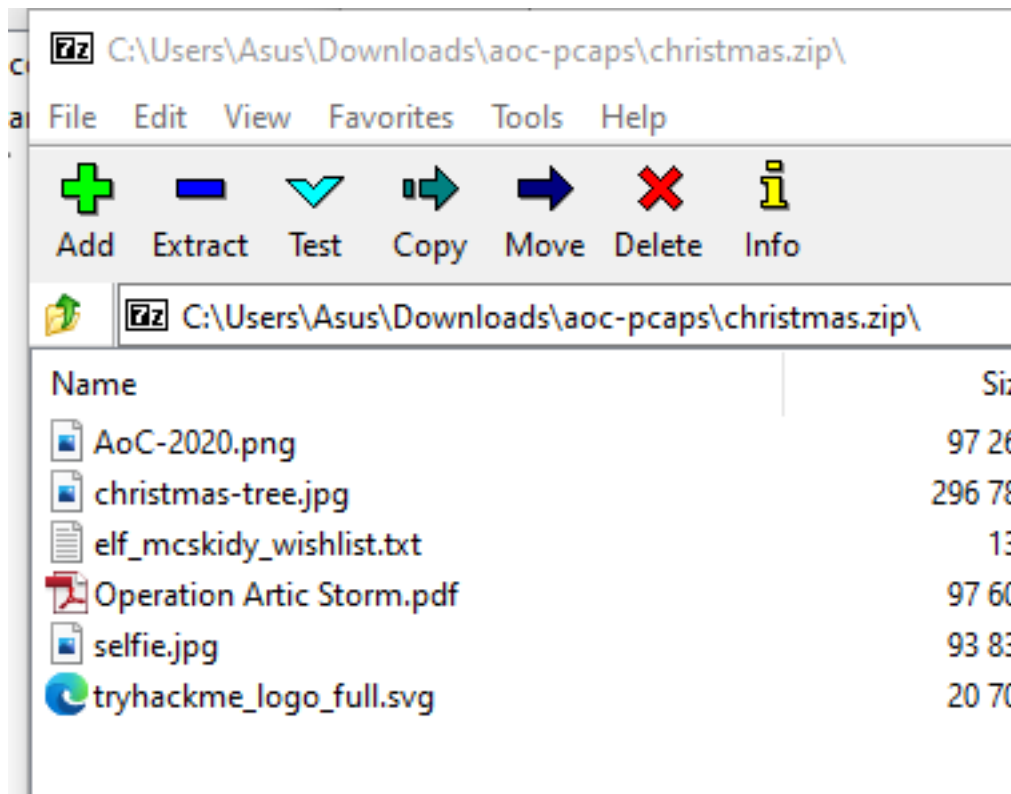


select save for the christmas.zip

Packet	Hostname	Content Type	Size	Filename
168	tbfc.blog	text/html	4532 bytes	\
395	tbfc.blog	application/zip	565kB	christmas.zip

opening the christmas.zip file & we found elf mcskidy wishlist text file





interesting stuff we found in there, seems like elf mcskidy wanna use rubber ducky to replace elf mceager

```
it x1 Rubber ducky (to replace Elf McEager)
```

Question: What is on Elf McSkidy's wishlist that will be used to replace Elf McEager?  
-rubber ducky