Challenge: PacketMaze Lab

Platform: CyberDefenders

Category: Network Forensics

Difficulty: Medium

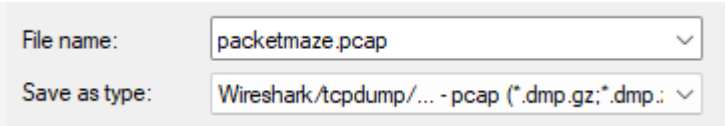
Tools Used: Wireshark, NetworkMiner

Summary: This challenge involved investigating a packet capture using Wireshark and NetworkMiner. I didn't find it that enjoyable, however, it is a good way to practice your Wireshark skills.

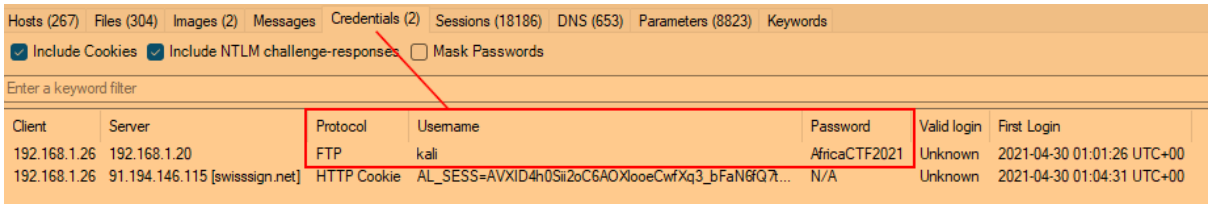
Scenario: A company's internal server has been flagged for unusual network activity, with multiple outbound connections to an unknown external IP. Initial analysis suggests possible data exfiltration. Investigate the provided network logs to determine the source and method of compromise.

What is the FTP password?

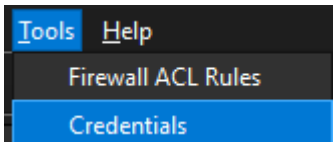
To find the FTP password, we can use NetworkMiner and look at the 'Credentials' tab. First, we need to convert the pcapng file into a pcap file that can work with the free version of NetworkMiner. To do so, open the pcapng with Wireshark, save the file and make sure to change the file type to pcap:



Within the credentials tab of NetworkMiner, we can find two credentials, and only one being for FTP:



Alternatively, if you navigate to Tools > Credentials in Wireshark:



You can see that Wireshark found three usernames:

Packet N	Protocol	Username	Additional Info
500	FTP	kali	Username in packet: 496
587	FTP	kali	Username in packet: 583
11816	FTP	kali	Username in packet: 11812

If you click on the first packet number (500), we can see the password in cleartext:

496	2021-04-30 01:01:26	192.168.1.26	192.168.1.20	21	FTP	Request: USER kali
497	2021-04-30 01:01:26	192.168.1.20	192.168.1.26	48794	TCP	21 → 48794 [ACK] Seq=113 Ac
498	2021-04-30 01:01:26	192.168.1.20	192.168.1.26	48794	FTP	Response: 331 Please specif
499	2021-04-30 01:01:26	192.168.1.26	192.168.1.20	21	TCP	48794 → 21 [ACK] Seq=32 Ack
500	2021-04-30 01:01:26	192.168.1.26	192.168.1.20	21	FTP	Request: PASS AfricaCTF2021

Answer: AfricaCTF2021

What is the IPv6 address of the DNS server used by 192.168.1.26?

I started by searching for all DNS requests originating from 192.168.1.26 using the following display filter:

- `ip.src == 192.168.1.26 and dns`

There are 6 results, all being sent to the destination host 192.168.1.10:

	Time	Source	Destination	Destination Port	Protocol	Info
51	2021-04-30 01:00:53	192.168.1.26	192.168.1.10	53	DNS	Standard query 0xa2ec A fp.msedge.net OPT
140	2021-04-30 01:00:56	192.168.1.26	192.168.1.10	53	DNS	Standard query 0x3b76 A l-ring.msedge.net OPT
171	2021-04-30 01:00:57	192.168.1.26	192.168.1.10	53	DNS	Standard query 0x303d A fp-vs-nocache.azureedge.net OPT
201	2021-04-30 01:00:57	192.168.1.26	192.168.1.10	53	DNS	Standard query 0xd289 A a-ring-fallback.msedge.net OPT
238	2021-04-30 01:00:59	192.168.1.26	192.168.1.10	53	DNS	Standard query 0x3800 A a-0001.a-afdentry.net.trafficmanager.net OPT
464	2021-04-30 01:01:11	192.168.1.26	192.168.1.10	53	DNS	Standard query 0x6820 A t-ring.msedge.net OPT

If you take note of the destination hosts' MAC address and filter for it, we can navigate to Statistics > Conversations > IPv6:

- `eth.addr==c8:09:a8:57:47:93`

2600:380:a85d:d1bd:ccf7:e5bf:2d97:ffe4	fe80::c80b:adff:feaa:1db7
fe80::8000:ffff:ffff:fffe	ff02::2
fe80::b011:ed39:8665:3b0a	fe80::c80b:adff:feaa:1db7
fe80::ffff:ffff:fffe	ff02::2

Here you can find the host in question sending multiple requests to fe80::c80b:adff:feaa:1db7. Upon using the following filter:

- `ipv6.addr==fe80::b011:ed39:8665:3b0a && ipv6.addr==fe80::c80b:adff:feaa:1db7`

I came across DNS requests:

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	198	100.0	41597	751	0	0	0	198
Ethernet	100.0	198	6.7	2772	50	0	0	0	198
Internet Protocol Version 6	100.0	198	19.0	7920	143	0	0	0	198
User Datagram Protocol	76.8	152	2.9	1216	21	0	0	0	152
Domain Name System	76.8	152	68.2	28353	512	152	28353	512	152
Internet Control Message Protocol v6	23.2	46	3.2	1336	24	46	1336	24	46

This means that 76.8% of the traffic between 192.168.1.26 to fe80::c80b:adff:feaa:1db7 is DNS related.

Answer: fe80::c80b:adff:feaa:1db7

What domain is the user looking up in packet 15174?

To navigate to a specific packet number, you can use the following query:

- frame.number == 15174

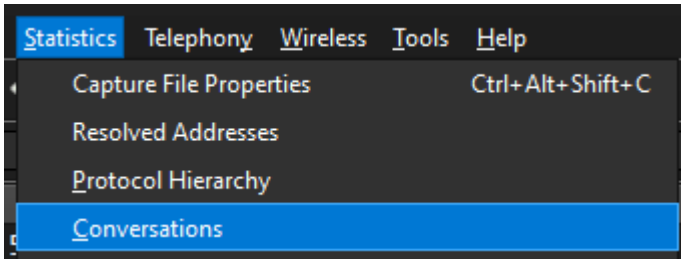
Info
Standard query 0x1ad5 A www.7-zip.org OPT

We can see that the user has made a DNS request for www.7-zip.org.

Answer: www.7-zip.org

How many UDP packets were sent from 192.168.1.26 to 24.39.217.246?

To find statistical data in Wireshark in relation to conversations, navigate to Statistics > Conversations:



If you navigate to the UDP tab, this is where you can find the number of UDP packets sent between two hosts:

Ethernet · 17 | IPv4 · 74 | IPv6 · 19 | TCP · 3236 | UDP · 127

Address A	Port A	Address B	Port B	Packets
192.168.1.26	53638	10.0.0.26	56004	1
192.168.1.26	53638	23.121.202.188	63286	1
192.168.1.26	60531	23.121.202.188	63286	1
192.168.1.26	53638	24.35.154.189	55038	80
192.168.1.26	57504	24.35.154.189	55038	33
192.168.1.26	51601	24.39.217.246	54150	1
192.168.1.26	53638	24.39.217.246	54150	9

Answer: 10

What is the MAC address of the system under investigation in the PCAP file?

If you look at the conversation statistics of this PCAP, you can see that most conversations are from 192.168.1.26:

Address A	Address B
0.0.0.0	224.0.0.1
173.223.18.66	192.168.1.26
192.168.1.26	10.0.0.26
192.168.1.26	13.107.18.254
192.168.1.26	13.107.21.200
192.168.1.26	13.107.42.254
192.168.1.26	13.107.246.51
192.168.1.26	13.107.246.254
192.168.1.26	20.54.89.15
192.168.1.26	20.190.151.67
192.168.1.26	23.51.191.35
192.168.1.26	23.63.78.40
192.168.1.26	23.121.202.188
192.168.1.26	24.35.154.189
192.168.1.26	24.39.217.246
192.168.1.26	34.122.121.32
192.168.1.26	35.186.220.63
192.168.1.26	35.232.111.17
192.168.1.26	40.65.246.52
192.168.1.26	40.70.229.150
192.168.1.26	44.237.173.75

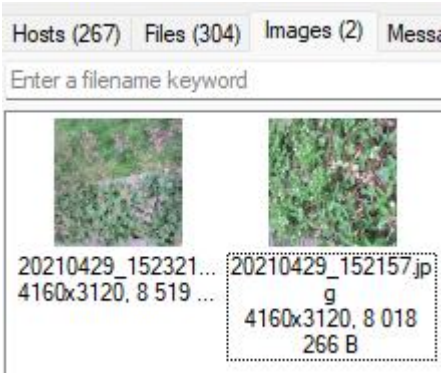
If you click on any packet with 192.168.1.26 as the source or destination, and expand the Ethernet section in the packet details pane, you can find the MAC address of this host:

```
▾ Ethernet II, Src: IntelCor_57:47:93 (c8:09:a8:57:47:93), Dst: ca:0b:ad:ad:20:ba (ca:0b:ad:ad:20:ba)
  ▾ Destination: ca:0b:ad:ad:20:ba (ca:0b:ad:ad:20:ba)
      Address: ca:0b:ad:ad:20:ba (ca:0b:ad:ad:20:ba)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory def…
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: IntelCor_57:47:93 (c8:09:a8:57:47:93)
      Address: IntelCor_57:47:93 (c8:09:a8:57:47:93)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
▸ Internet Protocol Version 4, Src: 192.168.1.26, Dst: 173.223.18.66
```

Answer: c8:09:a8:57:47:93

What was the camera model name used to take picture 20210429_152157.jpg?

Using NetworkMiner, we can click on the 'Images' tab to find the image in question:



If you right-click this file, and select open folder, we can run exiftool against it to extract the images metadata:

- `exiftool 20210429_152157.jpg | grep -i model`

```
Camera Model Name                : LM-Q725K
```

Answer: LM-Q725K

What is the ephemeral public key provided by the server during the TLS handshake in the session with the session ID: da4a0000342e4b73459d7360b4bea971cc303ac18d29b99067e46d16cc07f4ff?

To filter for the session ID in question, we can use the following display filter:

- `tls.handshake.session_id == da4a0000342e4b73459d7360b4bea971cc303ac18d29b99067e46d16cc07f4ff`

If you click on the only packet from the result and expand the TLS section in the packet details pane, you can find the public key provided by the server during the TLS handshake:

```
TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 7108
  Handshake Protocol: Server Hello
  Handshake Protocol: Certificate
  Handshake Protocol: Certificate Status
  Handshake Protocol: Server Key Exchange
    Handshake Type: Server Key Exchange (12)
    Length: 361
    EC Diffie-Hellman Server Params
      Curve Type: named_curve (0x03)
      Named Curve: secp384r1 (0x0018)
      Pubkey Length: 97
      Pubkey: 04edcc123af7b13e90ce101a31c2f996f471a7c8f48a1b81d765085f548059a550f3f4f62ca1f0e8f74…
```

Answer:
04edcc123af7b13e90ce101a31c2f996f471a7c8f48a1b81d765085f548059a550f3f4f62ca1f0e8f74d727053074a37bceb2cbdc7ce2a8994dcd76dd6834eefc5438c3b6da929321f3a1366bd14c877cc83e5d0731b7f80a6b80916efd4a23a4d

What is the first TLS 1.3 client random that was used to establish a connection with protonmail.com?

Using the following filter, I was able to find frames that contain protonmail.con:

- `frame contains "protonmail.com"`

Here we can find several packets related to the TLS handshake:

17992	2021-04-30 01:04:29	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
17997	2021-04-30 01:04:29	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
18000	2021-04-30 01:04:29	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
18144	2021-04-30 01:04:32	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
18145	2021-04-30 01:04:32	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
18146	2021-04-30 01:04:32	192.168.1.26	185.70.41.35	443	TLSv1.3	Client Hello (SNI=protonmail.com)
19069	2021-04-30 01:04:35	192.168.1.26	185.70.41.130	443	TLSv1.3	Client Hello (SNI=mail.protonmail.com)
19070	2021-04-30 01:04:35	192.168.1.26	185.70.41.130	443	TLSv1.3	Client Hello (SNI=mail.protonmail.com)
19093	2021-04-30 01:04:35	192.168.1.26	185.70.41.130	443	TLSv1.3	Client Hello (SNI=mail.protonmail.com)
20350	2021-04-30 01:04:42	192.168.1.26	185.70.41.130	443	TLSv1.3	Client Hello (SNI=mail.protonmail.com)

If you click on the first Client Hello message, you can find the client Random in the packet details pane under the TLS section:

```
▾ Transport Layer Security
  ▾ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 508
        Version: TLS 1.2 (0x0303)
        Random: 24e92513b97a0348f733d16996929a79be21b0b1400cd7e2862a732ce7775b70
```

Answer: 24e92513b97a0348f733d16996929a79be21b0b1400cd7e2862a732ce7775b70

Which country is the manufacturer of the FTP server's MAC address registered in?

If you filter for ftp in the display filter, we can see that the FTP server is running on 192.168.1.20:

ftp

No.	Time	Source	Destination	Destination Port	Protocol	Info
7049	2021-04-30 01:01:51	192.168.1.26	192.168.1.20	21	FTP	Request: LIST
7053	2021-04-30 01:01:51	192.168.1.20	192.168.1.26	48800	FTP	Response: 150 Here comes the directory listing.
7060	2021-04-30 01:01:51	192.168.1.20	192.168.1.26	48800	FTP	Response: 226 Directory send OK.
7067	2021-04-30 01:01:56	192.168.1.26	192.168.1.20	21	FTP	Request: PASV
7068	2021-04-30 01:01:56	192.168.1.20	192.168.1.26	48800	FTP	Response: 227 Entering Passive Mode (192,168,1,20,134,87).

We can retrieve the MAC address associated with the FTP server and search for it using a MAC lookup tool:

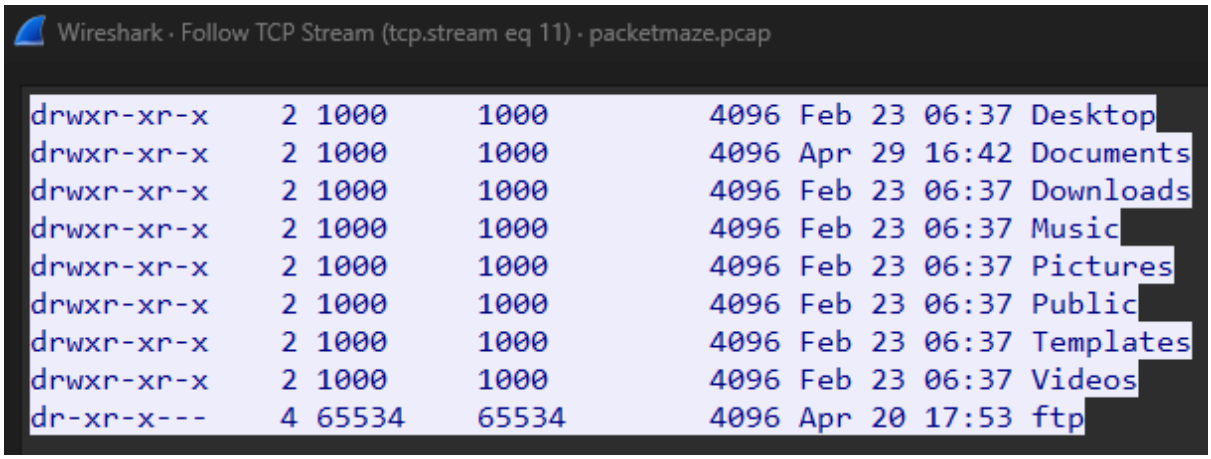
```
Ethernet II, Src: IntelCor_57:47:93 (c8:09:a8:57:47:93), Dst: PcsCompu_a6:1f:86 (08:00:27:a6:1f:86)
▾ Destination: PcsCompu_a6:1f:86 (08:00:27:a6:1f:86)
    Address: PcsCompu_a6:1f:86 (08:00:27:a6:1f:86)
```

We can see that the manufacturer of the FTP server's MAC address is registered in the United States.

Answer: United States

What time was a non-standard folder created on the FTP server on the 20th of April?

If you filter for ftp, we can see that on the 30 April 2021 at 01:01:26, the LIST command was requested by 192.168.1.26. If you follow the TCP stream for that packet, and click on the next stream, we can see the response from that command:



The non-standard folder is ftp and was created at 17:53 on April 20.

Answer: 17:53

What URL was visited by the user and connected to the IP address 104.21.89.171?

We can use the following filter to search for http traffic and the IP address in question:

- `http and ip.addr == 104.21.89.171`

This outputs two results, a GET request and its response. If you click on the GET request, you can find the host associated with this request:

```
Hypertext Transfer Protocol
  GET / HTTP/1.1\r\n
  Host: dfir.science\r\n
```

This means the user visited http://dfir.science/

Answer: http://dfir.science/