

Lab 5b: PCAP Analysis with Wireshark

Grade: 7% (100 Points)

Note: This assignment requires research and independent learning. You are encouraged to use web resources, official documentation, and tutorials to understand the functionality of each tool and protocols. The goal is to analyze the packet and find information about vulnerability.

What is PCAP?

PCAP stands for Packet Capture, which is a file format used to store network packet data captured from a network interface. It is commonly associated with network analysis and troubleshooting activities.

PCAP files contain the raw data of network packets, including the headers and payloads of each packet. These files can be generated by packet capture tools such as Wireshark, tcpdump, or other network monitoring software.

PCAP files are widely used in network analysis and security tasks. They enable network administrators, analysts, and researchers to inspect and analyze network traffic for various purposes, including:

1. Network troubleshooting: PCAP files can help diagnose network issues by examining packet-level details such as source and destination addresses, protocols, and error messages.
2. Network security: PCAP files are valuable for detecting and investigating network security incidents. They allow security professionals to analyze packet payloads, identify malicious activity, and track network intrusions.
3. Protocol analysis: PCAP files provide a wealth of information about network protocols. By analyzing the captured packets, researchers can gain insights into the behavior of network protocols, identify vulnerabilities, and develop mitigation strategies.
4. Performance monitoring: PCAP files can be used to measure network performance, identify bottlenecks, and optimize network configurations. They provide a detailed view of network traffic, allowing administrators to analyze latency, throughput, and other performance metrics.

To capture PCAP files you need to use a packet sniffer. A packet sniffer captures packets and presents them in a way that's easy to understand. When using a PCAP sniffer the first thing you need to do is identify what interface you want to sniff on.

Using Wireshark for PCAP file capture and analysis

Wireshark is the most popular traffic analyzer in the world. Wireshark uses .pcap files to record packet data that has been pulled from a network scan. Packet data is recorded in files with the .pcap file extension and can be used to find performance problems and cyberattacks on the network.

In other words, the PCAP file creates a record of network data that you can view through Wireshark. You can then assess the status of the network and identify if there are any service issues that you need to respond to.

It is important to note that Wireshark isn't the only tool that can open .pcap files. Other widely used alternatives include tcpdump and WinDump, network monitoring tools that also use PCAP to take a magnifying glass to network performance.

[Download the PCAP file provided and use your analysis tools to examine the provided PCAP file.](#)

Scenario:

You, as a SOC analyst, belong to a company specializing in hosting web applications through KVM-based Virtual Machines. Over the weekend, one VM went down, and the site administrators fear this might be the result of malicious activity. They extracted a few logs from the environment in hopes that you might be able to determine what happened.

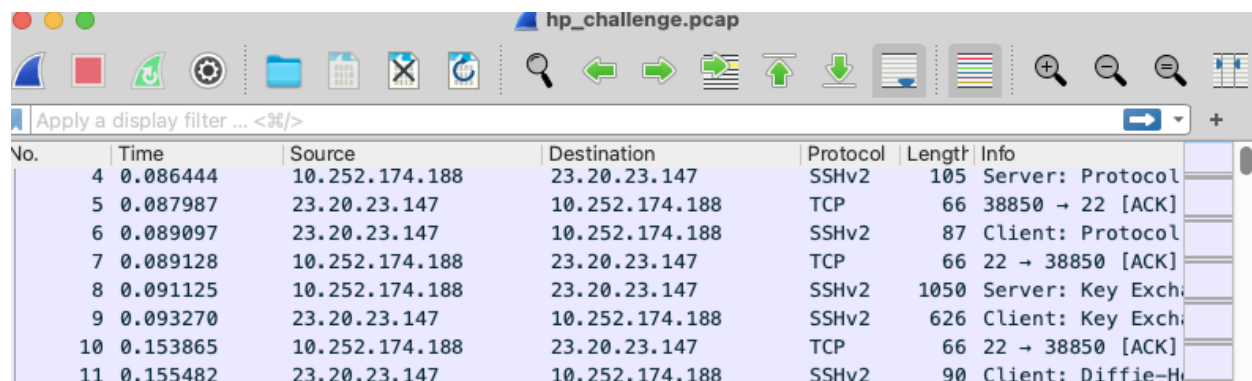
This challenge is a combination of several entry to intermediate-level tasks of increasing difficulty focusing on authentication, information hiding, and cryptography. Participants will benefit from entry-level knowledge in these fields, as well as knowledge of general Linux operations, kernel modules, a scripting language, and reverse engineering. Not everything may be as it seems. Innocuous files may turn out to be malicious so take precautions when dealing with any files from this challenge. In this lab, you will analyze pcap files from the scenario, whereas in the upcoming labs, you will analyze the log files.

Download the pcap file and use your analysis tools to examine provided PCAPs and log files.

[Challenge Questions](#)

Open hp_challen.pcap in Wireshark provided by the CTF challenge.

Next, use Wireshark to open the PCAP file and see if there is **SSH** traffic in the network capture file.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.086444	10.252.174.188	23.20.23.147	SSHv2	105	Server: Protocol
5	0.087987	23.20.23.147	10.252.174.188	TCP	66	38850 → 22 [ACK]
6	0.089097	23.20.23.147	10.252.174.188	SSHv2	87	Client: Protocol
7	0.089128	10.252.174.188	23.20.23.147	TCP	66	22 → 38850 [ACK]
8	0.091125	10.252.174.188	23.20.23.147	SSHv2	1050	Server: Key Exch
9	0.093270	23.20.23.147	10.252.174.188	SSHv2	626	Client: Key Exch
10	0.153865	10.252.174.188	23.20.23.147	TCP	66	22 → 38850 [ACK]
11	0.155482	23.20.23.147	10.252.174.188	SSHv2	90	Client: Diffie-H

1. (5 Points) What is SSH Protocol? What is the function of this protocol?

SSH Stands for secure shell which provides secure, encrypted transmission and communication between two computers over an unsecured network. It is often used for remote connection and command execution.

2. (5 Points) What other protocols can you see? List all of them.

SSHv2, TCP, HTTP

3. (10 Points) Find out what type of attack was used to gain access to the system? Explain.

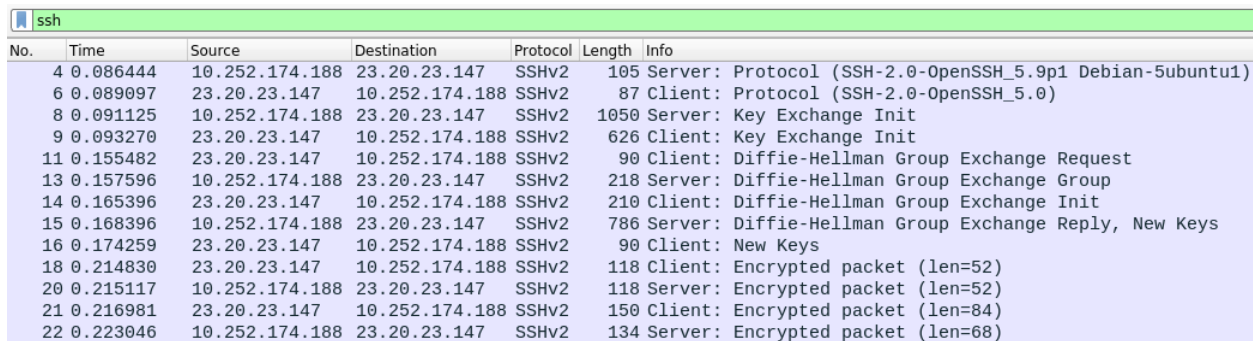
Hint: filter out all the ssh packet

Based on the timing of the ssh requests and the exchange of keys, whoever was sending the data was authorized so the likely reason for the type of attack would be a credential theft attack from either a brute force attack or stolen credentials of the username/ password. According to google, the attacker could have exploited a vulnerability in OpenSSH 5.9p1 (CVE-2014-2532) to bypass authentication. This would allow them to establish the SSH session without needing valid credentials



No.	Time	Source
-----	------	--------

You can see that the initial output from this filter shows multiple failed attempts to establish SSH sessions.



No.	Time	Source	Destination	Protocol	Length	Info
4	0.086444	10.252.174.188	23.20.23.147	SSHv2	105	Server: Protocol (SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1)
6	0.089097	23.20.23.147	10.252.174.188	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_5.0)
8	0.091125	10.252.174.188	23.20.23.147	SSHv2	1050	Server: Key Exchange Init
9	0.093270	23.20.23.147	10.252.174.188	SSHv2	626	Client: Key Exchange Init
11	0.155482	23.20.23.147	10.252.174.188	SSHv2	90	Client: Diffie-Hellman Group Exchange Request
13	0.157596	10.252.174.188	23.20.23.147	SSHv2	218	Server: Diffie-Hellman Group Exchange Group
14	0.165396	23.20.23.147	10.252.174.188	SSHv2	210	Client: Diffie-Hellman Group Exchange Init
15	0.168396	10.252.174.188	23.20.23.147	SSHv2	786	Server: Diffie-Hellman Group Exchange Reply, New Keys
16	0.174259	23.20.23.147	10.252.174.188	SSHv2	90	Client: New Keys
18	0.214830	23.20.23.147	10.252.174.188	SSHv2	118	Client: Encrypted packet (len=52)
20	0.215117	10.252.174.188	23.20.23.147	SSHv2	118	Server: Encrypted packet (len=52)
21	0.216981	23.20.23.147	10.252.174.188	SSHv2	150	Client: Encrypted packet (len=84)
22	0.223046	10.252.174.188	23.20.23.147	SSHv2	134	Server: Encrypted packet (len=68)

The image above represents the different steps that take place when attempting to establish and SSH session. These steps are briefly outlined below:

- The client and server **negotiate the SSH version** (i.e. packet no. 4 & 6).
- The client and server **exchanged public keys to generate secret key**. The server then issues a “New Keys” message and waits for the client to answer. (i.e. packet no. 8, 9, 11, 13, 14 and 15).
- The client **acknowledges the server’s “New Keys” message** (i.e. packet no. 16)
- We then see several **encrypted packets** before the SSH session is closed (i.e. packet no. 18, 20, 21 and 22).

Looking down through the SSH traffic, we see this process repeated multiple times until we near the end of the SSH filtered output. At packet number **1365**, we see an attempt to establish an SSH session, only this time we see far more encrypted packets than with previous attempts. If you look at the lower packets it shows guessing attacks. In cryptography, what is the attack that consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly.

Read more : https://resources.infosecinstitute.com/topics/incident-response-resources/network-traffic-analysis-for-ir-ssh-protocol-with-wireshark/?source=post_page-----ea7abcc68a18-----

4. (10 Points) What was the tool the attacker possibly used to perform this attack?

Hint: common linux bruteforce tool, Google it!

The common linux brute forcing tool which could be used for this attack would be the hydra ssh brute force or dictionary attack.

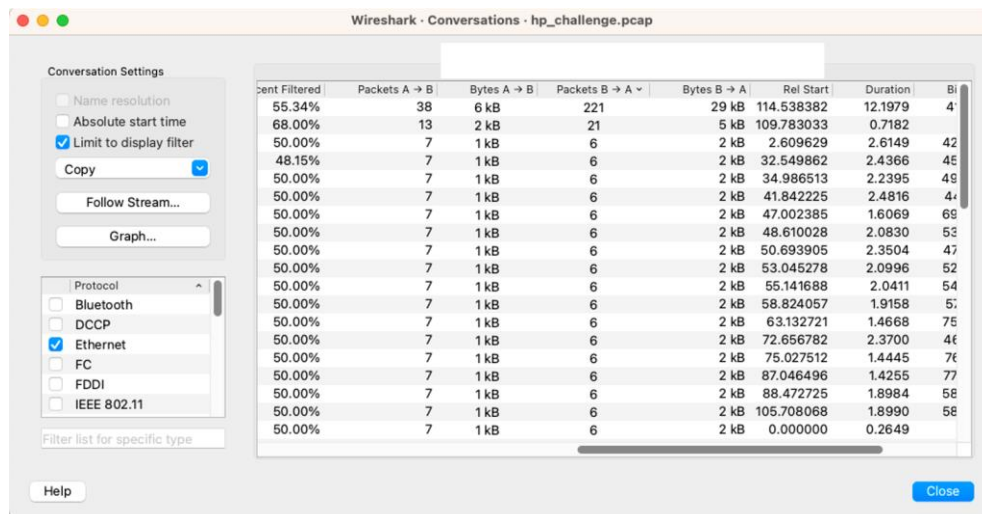
5. (10 Points) Now, find out how many failed attempts were there.

In the top left of my wireshark it says TCP * 54 meaning 54 total attempts were made. Looking at the photo, we know that the first row was a success since it's more packets than the rest since it completes the handshake, negotiates encryption, and exchanges encrypted data while the other rows don't have a large amount of data. We also see the highlighted blue row to have 468 packets sent which indicates a brute force attack over ssh probably using hydra. Since there are 54 rows, 2 successful, there are 52 failed attempts.

Ethernet · 1		TCP · 54								
Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes
23.20.23.147	33677	10.252.174.188	22	34	7 kB	52	50	68.00%	13	
23.20.23.147	34468	10.252.174.188	22	13	4 kB	36	28	46.43%	7	
23.20.23.147	35036	10.252.174.188	22	13	4 kB	26	26	50.00%	7	
23.20.23.147	35715	10.252.174.188	22	13	4 kB	17	26	50.00%	7	
23.20.23.147	36013	10.252.174.188	22	13	4 kB	6	27	48.15%	7	
23.20.23.147	36180	10.252.174.188	22	13	4 kB	20	26	50.00%	7	
23.20.23.147	36216	10.252.174.188	22	13	4 kB	18	26	50.00%	7	
23.20.23.147	36478	10.252.174.188	22	13	4 kB	19	26	50.00%	7	
23.20.23.147	37833	10.252.174.188	22	13	4 kB	43	26	50.00%	7	
23.20.23.147	37835	10.252.174.188	22	13	4 kB	21	26	50.00%	7	
23.20.23.147	38850	10.252.174.188	22	13	4 kB	0	26	50.00%	7	
23.20.23.147	39566	10.252.174.188	22	13	4 kB	31	26	50.00%	7	
23.20.23.147	40484	10.252.174.188	22	259	35 kB	53	468	55.34%	38	
23.20.23.147	41137	10.252.174.188	22	13	4 kB	34	26	50.00%	7	
23.20.23.147	41265	10.252.174.188	22	13	4 kB	10	27	48.15%	7	
23.20.23.147	41519	10.252.174.188	22	13	4 kB	5	26	50.00%	7	
23.20.23.147	41634	10.252.174.188	22	13	4 kB	9	26	50.00%	7	
23.20.23.147	42125	10.252.174.188	22	13	4 kB	48	26	50.00%	7	
23.20.23.147	42574	10.252.174.188	22	13	4 kB	1	26	50.00%	7	
23.20.23.147	42691	10.252.174.188	22	13	4 kB	37	26	50.00%	7	
23.20.23.147	43457	10.252.174.188	22	13	4 kB	13	26	50.00%	7	
23.20.23.147	43507	10.252.174.188	22	13	4 kB	16	26	50.00%	7	
23.20.23.147	43935	10.252.174.188	22	13	4 kB	40	26	50.00%	7	
23.20.23.147	44640	10.252.174.188	22	13	4 kB	41	26	50.00%	7	
23.20.23.147	44907	10.252.174.188	22	13	4 kB	50	26	50.00%	7	
23.20.23.147	45869	10.252.174.188	22	13	4 kB	11	28	46.43%	7	
23.20.23.147	47260	10.252.174.188	22	13	4 kB	29	26	50.00%	7	
23.20.23.147	47447	10.252.174.188	22	13	4 kB	7	26	50.00%	7	
23.20.23.147	47769	10.252.174.188	22	13	4 kB	25	26	50.00%	7	
23.20.23.147	47940	10.252.174.188	22	13	4 kB	51	26	50.00%	7	
23.20.23.147	49484	10.252.174.188	22	13	4 kB	4	26	50.00%	7	
23.20.23.147	49744	10.252.174.188	22	13	4 kB	33	26	50.00%	7	
23.20.23.147	51055	10.252.174.188	22	13	4 kB	49	26	50.00%	7	
23.20.23.147	51492	10.252.174.188	22	13	4 kB	2	26	50.00%	7	
23.20.23.147	51803	10.252.174.188	22	13	4 kB	14	28	46.43%	7	
23.20.23.147	52014	10.252.174.188	22	13	4 kB	24	26	50.00%	7	
23.20.23.147	52474	10.252.174.188	22	13	4 kB	28	26	50.00%	7	
23.20.23.147	52758	10.252.174.188	22	13	4 kB	42	26	50.00%	7	
23.20.23.147	53469	10.252.174.188	22	13	4 kB	30	26	50.00%	7	
23.20.23.147	53534	10.252.174.188	22	13	4 kB	45	26	50.00%	7	
23.20.23.147	55025	10.252.174.188	22	13	4 kB	3	26	50.00%	7	
23.20.23.147	57384	10.252.174.188	22	13	4 kB	23	26	50.00%	7	
23.20.23.147	57752	10.252.174.188	22	13	4 kB	12	26	50.00%	7	
23.20.23.147	57787	10.252.174.188	22	13	4 kB	47	26	50.00%	7	
23.20.23.147	58283	10.252.174.188	22	13	4 kB	46	26	50.00%	7	
23.20.23.147	58441	10.252.174.188	22	13	4 kB	39	26	50.00%	7	
23.20.23.147	58563	10.252.174.188	22	13	4 kB	32	26	50.00%	7	
23.20.23.147	58975	10.252.174.188	22	13	4 kB	8	26	50.00%	7	
23.20.23.147	59034	10.252.174.188	22	13	4 kB	27	26	50.00%	7	
23.20.23.147	59095	10.252.174.188	22	13	4 kB	22	26	50.00%	7	
23.20.23.147	59172	10.252.174.188	22	13	4 kB	15	27	48.15%	7	
23.20.23.147	59454	10.252.174.188	22	13	4 kB	35	26	50.00%	7	
23.20.23.147	60067	10.252.174.188	22	13	4 kB	44	26	50.00%	7	
23.20.23.147	60670	10.252.174.188	22	13	4 kB	38	26	50.00%	7	

To identify the number of failed attempts, make sure that you are still filtering for SSH traffic in the main Wireshark view.

Navigate to “*Statistics > Conversations > TCP tab*” in Wireshark. At the bottom of the conversations window, there is a checkbox option to limit what we see to our display filter only (i.e. SSH traffic). After enabling this option, we see only SSH traffic under the TCP tab. Check the number. See how many were successful and reduce it from total attempts the attacker made.



6. (10 Points) What is the tool used to download malicious files on the system?

The tool used to download the malicious files on the system can be seen in the screenshot below, it was using wget, which is a tool that allows user to download files on HTTP, HTTPS, and FTP.

Based on our earlier findings, we know that the only other protocol present, apart from SSH, is HTTP. Now, filter the HTTP traffic.

Select the first HTTP packet and follow its HTTP stream. If you look carefully at the request headers highlighted, you can see that the User-Agent request header has the value. It is a tool that retrieves content from web servers by downloading via HTTP, HTTPS, and FTP.

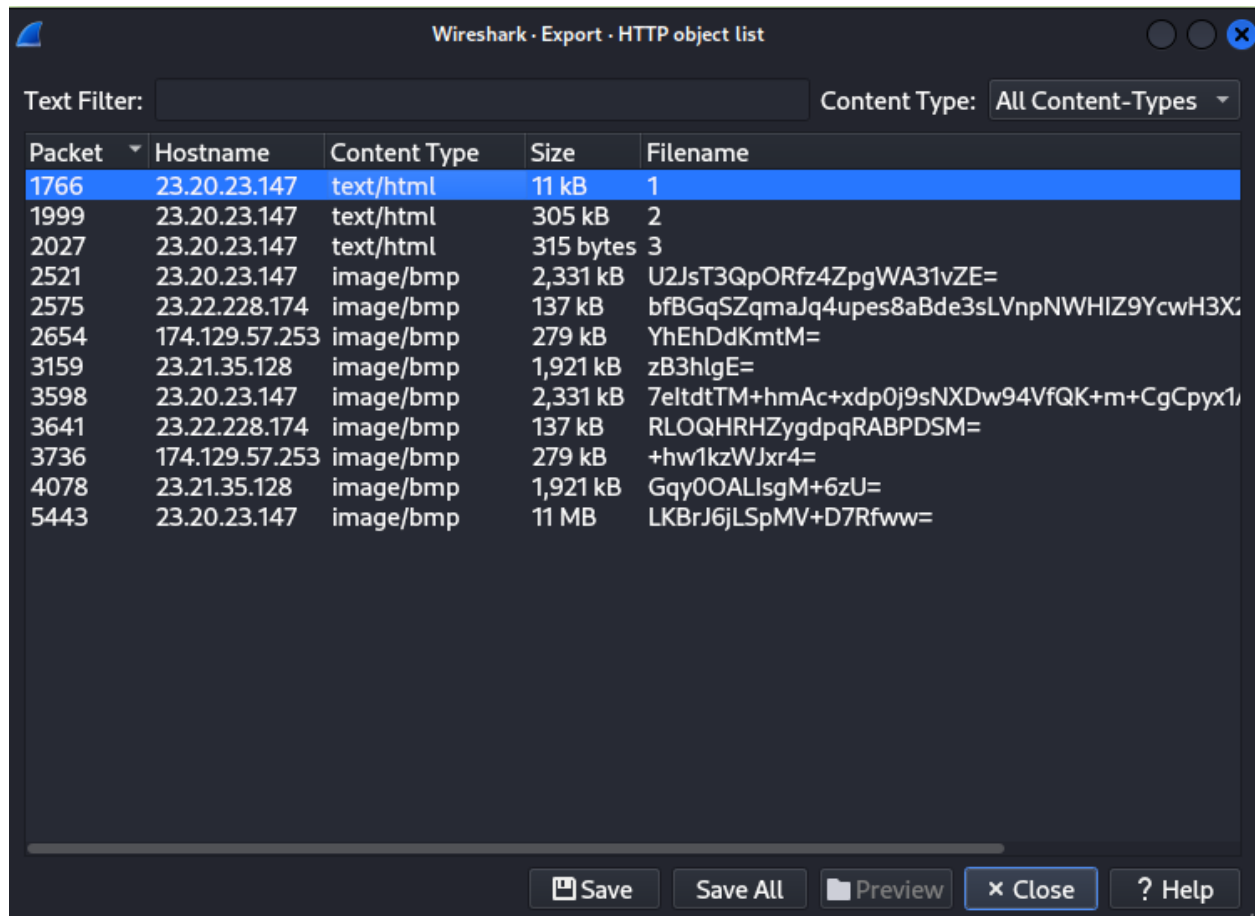
```

v Hypertext Transfer Protocol
  > GET /d/1 HTTP/1.1\r\n
  User-Agent: Wget/1.13.4 (linux-gnu)\r\n
  Accept: */*\r\n
  Host: 23.20.23.147\r\n
  Connection: Keep-Alive\r\n
  \r\n
  [Full request URI: http://23.20.23.147/d/1]
  [HTTP request 1/1]
  [Response in frame: 1766]

```

7. (10 Points) Find out how many files the attacker downloaded to perform malware installation. Attach screenshot.

Below is the screenshot from my Wireshark export. There are 12 files total including filename 1, 2, and 3 which could be part of the attack. According to google, typically malicious attackers install malware with base64 encoded names to obfuscate the file names of their attack which would be the bottom 9 files. The BMP files indicate the malware installation, so the attacker downloaded 9 files.



Packet	Hostname	Content Type	Size	Filename
1766	23.20.23.147	text/html	11 kB	1
1999	23.20.23.147	text/html	305 kB	2
2027	23.20.23.147	text/html	315 bytes	3
2521	23.20.23.147	image/bmp	2,331 kB	U2JsT3QpORfz4ZpgWA31vZE=
2575	23.22.228.174	image/bmp	137 kB	bFBGqSZqmaJq4upes8aBde3sLVnpNWHIZ9YcwH3X:
2654	174.129.57.253	image/bmp	279 kB	YhEhDdKmtM=
3159	23.21.35.128	image/bmp	1,921 kB	zB3hlgE=
3598	23.20.23.147	image/bmp	2,331 kB	7eltdtTM+hmAc+xdp0j9sNXDw94VfQK+m+CgCpyx1,
3641	23.22.228.174	image/bmp	137 kB	RLOQHRHZygdppqRABPDSM=
3736	174.129.57.253	image/bmp	279 kB	+hw1kzWJxr4=
4078	23.21.35.128	image/bmp	1,921 kB	Gqy0OALlsgM+6zU=
5443	23.20.23.147	image/bmp	11 MB	LKBrJ6jLSpMV+D7Rfww=

To answer this question, navigate to “*File > Export Objects > HTTP*” in Wireshark. In this window, you can see three files named 1, 2 and 3. There are also multiple BMP files with base64 encoded filenames:

Packet	Hostname	Content Type	Size	Filename
1766	23.20.23.147	text/html	11 kB	1
1999	23.20.23.147	text/html	305 kB	2
2027	23.20.23.147	text/html	315 bytes	3
2521	23.20.23.147	image/bmp	2331 kB	U2JsT3QpORfz4ZpgWA31vZE=
2575	23.22.228.174	image/bmp	137 kB	bFBGqSZqmaJq4upes8aBde3sLVnpNWHIZ9YcwH3X2hWRI
2654	174.129.57.253	image/bmp	279 kB	YhEhDdKmtM=
3159	23.21.35.128	image/bmp	1921 kB	zB3hlgE=
3598	23.20.23.147	image/bmp	2331 kB	7elttdTM+hmA+xdp0j9sNXDw94VfQK+m+CgCpyx1AVPFS
3641	23.22.228.174	image/bmp	137 kB	RLOQHRHZygdpgRABPDSM=
3736	174.129.57.253	image/bmp	279 kB	+hw1kzWJxr4=
4078	23.21.35.128	image/bmp	1921 kB	Gqy0OALIsGm+6zU=
5443	23.20.23.147	image/bmp	11 MB	LKBrJ6jLSpMV+D7Rfww=

HTTP Objects

Copy your screenshot here and label it.

8. (10 Points) One of the IP's the malware contacted starts with 17. Find and provide the full IP.

174.129.57.253

(screenshot below)

The image shows a Wireshark packet capture of an HTTP session. The packet list on the left shows several GET requests. Packet 3736 is selected, and its details are shown in the packet details pane. The details pane shows the HTTP request structure, including the URI and content type. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
1744	122.331491	10.252.174.188	23.20.23.147	HTTP	181	GET /d/1 HTTP/1.1
1766	122.571830	23.20.23.147	10.252.174.188	HTTP	71	HTTP/1.1 200 OK (text/html)
1795	122.580931	10.252.174.188	23.20.23.147	HTTP	181	GET /d/2 HTTP/1.1
1999	122.843290	23.20.23.147	10.252.174.188	HTTP	489	HTTP/1.1 200 OK (text/html)
2020	122.852487	10.252.174.188	23.20.23.147	HTTP	181	GET /d/3 HTTP/1.1
2027	122.855661	23.20.23.147	10.252.174.188	HTTP	71	HTTP/1.1 200 OK (text/html)
2106	318.638960	10.252.174.188	23.20.23.147	HTTP	352	GET /n/bsmHSxbNLOqx6jycBS677vZFxEVfJXhkn13GEwa7Ed8Klsp9Bewf
2521	320.671617	23.20.23.147	10.252.174.188	HTTP	736	HTTP/1.1 200 OK (image/bmp)
2529	442.247121	10.252.174.188	23.22.228.174	HTTP	353	GET /n/kPkC9Vhd2m5VasUQFzqvFKtQ4XMT86FmgLzIB5V9myrcSWsFJEGxP8c
2575	442.426525	23.22.228.174	10.252.174.188	HTTP	1447	HTTP/1.1 200 OK (image/bmp)
2583	884.485955	10.252.174.188	174.129.57.253	HTTP	354	GET /n/vtXrOV1HstqG120v/CwLeuJXLhQHiJJbRYFJOJlN3Bds1S4Gb3JbwG,
2654	884.743448	174.129.57.253	10.252.174.188	HTTP	1315	HTTP/1.1 200 OK (image/bmp)
2662	1652.107107	10.252.174.188	23.21.35.128	HTTP	352	GET /n/MzLN2Yya80gPhtA06fAsER18hvNWuysSFHu146f2df64K6IT8VYQ108;
3159	1653.455551	23.21.35.128	10.252.174.188	HTTP	1308	HTTP/1.1 200 OK (image/bmp)
3167	2309.679056	10.252.174.188	23.20.23.147	HTTP	352	GET /n/JURgH3Pd1rSdCYCuI3yWe1Zpir3DvDt1v4FBXmj6I+y1Brg3C9TgPV
3598	2311.847698	23.20.23.147	10.252.174.188	HTTP	736	HTTP/1.1 200 OK (image/bmp)
3606	3082.028785	10.252.174.188	23.22.228.174	HTTP	353	GET /n/d1fv+eJkdZulJPj8rquuxHFbI0h1Mu6LR65HGhkiX2dwsUSxwdo3he,
3641	3082.217002	23.22.228.174	10.252.174.188	HTTP	1447	HTTP/1.1 200 OK (image/bmp)
3649	4117.396949	10.252.174.188	174.129.57.253	HTTP	353	GET /n/bsmHSxbNLOqx6jycBS677vZFxEVfJXhkn13GEwa7Ed8Klsp9Bewf
3736	4117.671690	174.129.57.253	10.252.174.188	HTTP	279	GET /n/bsmHSxbNLOqx6jycBS677vZFxEVfJXhkn13GEwa7Ed8Klsp9Bewf
3744	4238.323755	10.252.174.188	23.21.35.128	HTTP	1308	HTTP/1.1 200 OK (image/bmp)
4078	4239.769725	23.21.35.128	10.252.174.188	HTTP	736	HTTP/1.1 200 OK (image/bmp)
4086	4334.784014	10.252.174.188	23.20.23.147	HTTP	353	GET /n/d1fv+eJkdZulJPj8rquuxHFbI0h1Mu6LR65HGhkiX2dwsUSxwdo3he,
5443	4335.511676	23.20.23.147	10.252.174.188	HTTP	1447	HTTP/1.1 200 OK (image/bmp)

Frame 3736: 5659 bytes on wire (45272 bits), 5659 bytes captured (45272 bits) on interface 0, 10.252.174.188 to 174.129.57.253, Ethernet II, Src: fe:ff:ff:ff:ff:ff (fe:ff:ff:ff:ff:ff), Dst: 08:00:27:1d:33:03 (08:00:27:1d:33:03), Internet Protocol Version 4, Src: 174.129.57.253, Dst: 10.252.174.188, Transmission Control Protocol, Src Port: 80, Dst Port: 443, [...] 47 Reassembled TCP Segments (279265 bytes): #3651(7700) ... #3736(279) ... #5443(1447) ... #5444(1447) ... #5445(1447) ... #5446(1447) ... #5447(1447) ... #5448(1447) ... #5449(1447) ... #5450(1447) ... #5451(1447) ... #5452(1447) ... #5453(1447) ... #5454(1447) ... #5455(1447) ... #5456(1447) ... #5457(1447) ... #5458(1447) ... #5459(1447) ... #5460(1447) ... #5461(1447) ... #5462(1447) ... #5463(1447) ... #5464(1447) ... #5465(1447) ... #5466(1447) ... #5467(1447) ... #5468(1447) ... #5469(1447) ... #5470(1447) ... #5471(1447) ... #5472(1447) ... #5473(1447) ... #5474(1447) ... #5475(1447) ... #5476(1447) ... #5477(1447) ... #5478(1447) ... #5479(1447) ... #5480(1447) ... #5481(1447) ... #5482(1447) ... #5483(1447) ... #5484(1447) ... #5485(1447) ... #5486(1447) ... #5487(1447) ... #5488(1447) ... #5489(1447) ... #5490(1447) ... #5491(1447) ... #5492(1447) ... #5493(1447) ... #5494(1447) ... #5495(1447) ... #5496(1447) ... #5497(1447) ... #5498(1447) ... #5499(1447) ... #5500(1447) ... #5501(1447) ... #5502(1447) ... #5503(1447) ... #5504(1447) ... #5505(1447) ... #5506(1447) ... #5507(1447) ... #5508(1447) ... #5509(1447) ... #5510(1447) ... #5511(1447) ... #5512(1447) ... #5513(1447) ... #5514(1447) ... #5515(1447) ... #5516(1447) ... #5517(1447) ... #5518(1447) ... #5519(1447) ... #5520(1447) ... #5521(1447) ... #5522(1447) ... #5523(1447) ... #5524(1447) ... #5525(1447) ... #5526(1447) ... #5527(1447) ... #5528(1447) ... #5529(1447) ... #5530(1447) ... #5531(1447) ... #5532(1447) ... #5533(1447) ... #5534(1447) ... #5535(1447) ... #5536(1447) ... #5537(1447) ... #5538(1447) ... #5539(1447) ... #5540(1447) ... #5541(1447) ... #5542(1447) ... #5543(1447) ... #5544(1447) ... #5545(1447) ... #5546(1447) ... #5547(1447) ... #5548(1447) ... #5549(1447) ... #5550(1447) ... #5551(1447) ... #5552(1447) ... #5553(1447) ... #5554(1447) ... #5555(1447) ... #5556(1447) ... #5557(1447) ... #5558(1447) ... #5559(1447) ... #5560(1447) ... #5561(1447) ... #5562(1447) ... #5563(1447) ... #5564(1447) ... #5565(1447) ... #5566(1447) ... #5567(1447) ... #5568(1447) ... #5569(1447) ... #5570(1447) ... #5571(1447) ... #5572(1447) ... #5573(1447) ... #5574(1447) ... #5575(1447) ... #5576(1447) ... #5577(1447) ... #5578(1447) ... #5579(1447) ... #5580(1447) ... #5581(1447) ... #5582(1447) ... #5583(1447) ... #5584(1447) ... #5585(1447) ... #5586(1447) ... #5587(1447) ... #5588(1447) ... #5589(1447) ... #5590(1447) ... #5591(1447) ... #5592(1447) ... #5593(1447) ... #5594(1447) ... #5595(1447) ... #5596(1447) ... #5597(1447) ... #5598(1447) ... #5599(1447) ... #5600(1447) ... #5601(1447) ... #5602(1447) ... #5603(1447) ... #5604(1447) ... #5605(1447) ... #5606(1447) ... #5607(1447) ... #5608(1447) ... #5609(1447) ... #5610(1447) ... #5611(1447) ... #5612(1447) ... #5613(1447) ... #5614(1447) ... #5615(1447) ... #5616(1447) ... #5617(1447) ... #5618(1447) ... #5619(1447) ... #5620(1447) ... #5621(1447) ... #5622(1447) ... #5623(1447) ... #5624(1447) ... #5625(1447) ... #5626(1447) ... #5627(1447) ... #5628(1447) ... #5629(1447) ... #5630(1447) ... #5631(1447) ... #5632(1447) ... #5633(1447) ... #5634(1447) ... #5635(1447) ... #5636(1447) ... #5637(1447) ... #5638(1447) ... #5639(1447) ... #5640(1447) ... #5641(1447) ... #5642(1447) ... #5643(1447) ... #5644(1447) ... #5645(1447) ... #5646(1447) ... #5647(1447) ... #5648(1447) ... #5649(1447) ... #5650(1447) ... #5651(1447) ... #5652(1447) ... #5653(1447) ... #5654(1447) ... #5655(1447) ... #5656(1447) ... #5657(1447) ... #5658(1447) ... #5659(1447) ... #5660(1447) ... #5661(1447) ... #5662(1447) ... #5663(1447) ... #5664(1447) ... #5665(1447) ... #5666(1447) ... #5667(1447) ... #5668(1447) ... #5669(1447) ... #5670(1447) ... #5671(1447) ... #5672(1447) ... #5673(1447) ... #5674(1447) ... #5675(1447) ... #5676(1447) ... #5677(1447) ... #5678(1447) ... #5679(1447) ... #5680(1447) ... #5681(1447) ... #5682(1447) ... #5683(1447) ... #5684(1447) ... #5685(1447) ... #5686(1447) ... #5687(1447) ... #5688(1447) ... #5689(1447) ... #5690(1447) ... #5691(1447) ... #5692(1447) ... #5693(1447) ... #5694(1447) ... #5695(1447) ... #5696(1447) ... #5697(1447) ... #5698(1447) ... #5699(1447) ... #5700(1447) ... #5701(1447) ... #5702(1447) ... #5703(1447) ... #5704(1447) ... #5705(1447) ... #5706(1447) ... #5707(1447) ... #5708(1447) ... #5709(1447) ... #5710(1447) ... #5711(1447) ... #5712(1447) ... #5713(1447) ... #5714(1447) ... #5715(1447) ... #5716(1447) ... #5717(1447) ... #5718(1447) ... #5719(1447) ... #5720(1447) ... #5721(1447) ... #5722(1447) ... #5723(1447) ... #5724(1447) ... #5725(1447) ... #5726(1447) ... #5727(1447) ... #5728(1447) ... #5729(1447) ... #5730(1447) ... #5731(1447) ... #5732(1447) ... #5733(1447) ... #5734(1447) ... #5735(1447) ... #5736(1447) ... #5737(1447) ... #5738(1447) ... #5739(1447) ... #5740(1447) ... #5741(1447) ... #5742(1447) ... #5743(1447) ... #5744(1447) ... #5745(1447) ... #5746(1447) ... #5747(1447) ... #5748(1447) ... #5749(1447) ... #5750(1447) ... #5751(1447) ... #5752(1447) ... #5753(1447) ... #5754(1447) ... #5755(1447) ... #5756(1447) ... #5757(1447) ... #5758(1447) ... #5759(1447) ... #5760(1447) ... #5761(1447) ... #5762(1447) ... #5763(1447) ... #5764(1447) ... #5765(1447) ... #5766(1447) ... #5767(1447) ... #5768(1447) ... #5769(1447) ... #5770(1447) ... #5771(1447) ... #5772(1447) ... #5773(1447) ... #5774(1447) ... #5775(1447) ... #5776(1447) ... #5777(1447) ... #5778(1447) ... #5779(1447) ... #5780(1447) ... #5781(1447) ... #5782(1447) ... #5783(1447) ... #5784(1447) ... #5785(1447) ... #5786(1447) ... #5787(1447) ... #5788(1447) ... #5789(1447) ... #5790(1447) ... #5791(1447) ... #5792(1447) ... #5793(1447) ... #5794(1447) ... #5795(1447) ... #5796(1447) ... #5797(1447) ... #5798(1447) ... #5799(1447) ... #5800(1447) ... #5801(1447) ... #5802(1447) ... #5803(1447) ... #5804(1447) ... #5805(1447) ... #5806(1447) ... #5807(1447) ... #5808(1447) ... #5809(1447) ... #5810(1447) ... #5811(1447) ... #5812(1447) ... #5813(1447) ... #5814(1447) ... #5815(1447) ... #5816(1447) ... #5817(1447) ... #5818(1447) ... #5819(1447) ... #5820(1447) ... #5821(1447) ... #5822(1447) ... #5823(1447) ... #5824(1447) ... #5825(1447) ... #5826(1447) ... #5827(1447) ... #5828(1447) ... #5829(1447) ... #5830(1447) ... #5831(1447) ... #5832(1447) ... #5833(1447) ... #5834(1447) ... #5835(1447) ... #5836(1447) ... #5837(1447) ... #5838(1447) ... #5839(1447) ... #5840(1447) ... #5841(1447) ... #5842(1447) ... #5843(1447) ... #5844(1447) ... #5845(1447) ... #5846(1447) ... #5847(1447) ... #5848(1447) ... #5849(1447) ... #5850(1447) ... #5851(1447) ... #5852(1447) ... #5853(1447) ... #5854(1447) ... #5855(1447) ... #5856(1447) ... #5857(1447) ... #5858(1447) ... #5859(1447) ... #5860(1447) ... #5861(1447) ... #5862(1447) ... #5863(1447) ... #5864(1447) ... #5865(1447) ... #5866(1447) ... #5867(1447) ... #5868(1447) ... #5869(1447) ... #5870(1447) ... #5871(1447) ... #5872(1447) ... #5873(1447) ... #5874(1447) ... #5875(1447) ... #5876(1447) ... #5877(1447) ... #5878(1447) ... #5879(1447) ... #5880(1447) ... #5881(1447) ... #5882(1447) ... #5883(1447) ... #5884(1447) ... #5885(1447) ... #5886(1447) ... #5887(1447) ... #5888(1447) ... #5889(1447) ... #5890(1447) ... #5891(1447) ... #5892(1447) ... #5893(1447) ... #5894(1447) ... #5895(1447) ... #5896(1447) ... #5897(1447) ... #5898(1447) ... #5899(1447) ... #5900(1447) ... #5901(1447) ... #5902(1447) ... #5903(1447) ... #5904(1447) ... #5905(1447) ... #5906(1447) ... #5907(1447) ... #5908(1447) ... #5909(1447) ... #5910(1447) ... #5911(1447) ... #5912(1447) ... #5913(1447) ... #5914(1447) ... #5915(1447) ... #5916(1447) ... #5917(1447) ... #5918(1447) ... #5919(1447) ... #5920(1447) ... #5921(1447) ... #5922(1447) ... #5923(1447) ... #5924(1447) ... #5925(1447) ... #5926(1447) ... #5927(1447) ... #5928(1447) ... #5929(1447) ... #5930(1447) ... #5931(1447) ... #5932(1447) ... #5933(1447) ... #5934(1447) ... #5935(1447) ... #5936(1447) ... #5937(1447) ... #5938(1447) ... #5939(1447) ... #5940(1447) ... #5941(1447) ... #5942(1447) ... #5943(1447) ... #5944(1447) ... #5945(1447) ... #5946(1447) ... #5947(1447) ... #5948(1447) ... #5949(1447) ... #5950(1447) ... #5951(1447) ... #5952(1447) ... #5953(1447) ... #5954(1447) ... #5955(1447) ... #5956(1447) ... #5957(1447) ... #5958(1447) ... #5959(1447) ... #5960(1447) ... #5961(1447) ... #5962(1447) ... #5963(1447) ... #5964(1447) ... #5965(1447) ... #5966(1447) ... #5967(1447) ... #5968(1447) ... #5969(1447) ... #5970(1447) ... #5971(1447) ... #5972(1447) ... #5973(1447) ... #5974(1447) ... #5975(1447) ... #5976(1447) ... #5977(1447) ... #5978(1447) ... #5979(1447) ... #5980(1447) ... #5981(1447) ... #5982(1447) ... #5983(1447) ... #5984(1447) ... #5985(1447) ... #5986(1447) ... #5987(1447) ... #5988(1447) ... #5989(1447) ... #5990(1447) ... #5991(1447) ... #5992(1447) ... #5993(1447) ... #5994(1447) ... #5995(1447) ... #5996(1447) ... #5997(1447) ... #5998(1447) ... #5999(1447) ... #6000(1447) ... #6001(1447) ... #6002(1447) ... #6003(1447) ... #6004(1447) ... #6005(1447) ... #6006(1447) ... #6007(1447) ... #6008(1447) ... #6009(1447) ... #6010(1447) ... #6011(1447) ... #6012(1447) ... #6013(1447) ... #6014(1447) ... #6015(1447) ... #6016(1447) ... #6017(1447) ... #6018(1447) ... #6019(1447) ... #6020(1447) ... #6021(1447) ... #6022(1447) ... #6023(1447) ... #6024(1447) ... #6025(1447) ... #6026(1447) ... #6027(1447) ... #6028(1447) ... #6029(1447) ... #6030(1447) ... #6031(1447) ... #6032(1447) ... #6033(1447) ... #6034(1447) ... #6035(1447) ... #6036(1447) ... #6037(1447) ... #6038(1447) ... #6039(1447) ... #6040(1447) ... #6041(1447) ... #6042(1447) ... #6043(1447) ... #6044(1447) ... #6045(1447) ... #6046(1447) ... #6047(1447) ... #6048(1447) ... #6049(1447) ... #6050(1447) ... #6051(1447) ... #6052(1447) ... #6053(1447) ... #6054(1447) ... #6055(1447) ... #6056(1447) ... #6057(1447) ... #6058(1447) ... #6059(1447) ... #6060(1447) ... #6061(1447) ... #6062(1447) ... #6063(1447) ... #6064(1447) ... #6065(1447) ... #6066(1447) ... #6067(1447) ... #6068(1447) ... #6069(1447) ... #6070(1447) ... #6071(1447) ... #6072(1447) ... #6073(1447) ... #6074(1447) ... #6075(1447) ... #6076(1447) ... #6077(1447) ... #6078(1447) ... #6079(1447) ... #6080(1447) ... #6081(1447) ... #6082(1447) ... #6083(1447) ... #6084(1447) ... #6085(1447) ... #6086(1447) ... #6087(1447) ... #6088(1447) ... #6089(1447) ... #6090(1447) ... #6091(1447) ... #6092(1447) ... #6093(1447) ... #6094(1447) ... #6095(1447) ... #6096(1447) ... #6097(1447) ... #6098(1447) ... #6099(1447) ... #6100(1447) ... #6101(1447) ... #6102(1447) ... #6103(1447) ... #6104(1447) ... #6105(1447) ... #6106(1447) ... #6107(1447) ... #6108(1447) ... #6109(1447) ... #6110(1447) ... #6111(1447) ... #6112(1447) ... #6113(1447) ... #6114(1447) ... #6115(1447) ... #6116(1447) ... #6117(1447) ... #6118(1447) ... #6119(1447) ... #6120(1447) ... #6121(1447) ... #6122(1447) ... #6123(1447) ... #6124(1447) ... #6125(1447) ... #6126(1447) ... #6127(1447) ... #6128(1447) ... #6129(1447) ... #6130(1447) ... #6131(1447) ... #6132(1447) ... #6133(1447) ... #6134(1447) ... #6135(1447) ... #6136(1447) ... #6137(1447) ... #6138(1447) ... #6139(1447) ... #6140(1447) ... #6141(1447) ... #6142(1447) ... #6143(1447) ... #6144(1447) ... #6145(1447) ... #6146(1447) ... #6147(1447) ... #6148(1447) ... #6149(1447) ... #6150(1447) ... #6151(1447) ... #6152(1447) ... #6153(1447) ... #6154(1447) ... #6155(1447) ... #6156(1447) ... #6157(1447) ... #6158(1447) ... #6159(1447) ... #6160(1447) ... #6161(1447) ... #6162(1447) ... #6163(1447) ... #6164(1447) ... #6165(1447) ... #6166(1447) ... #6167(1447) ... #6168(1447) ... #6169(1447) ... #6170(1447) ... #6171(1447) ... #6172(1447) ... #6173(1447) ... #6174(1447) ... #6175(1447) ... #6176(1447) ... #6177(1447) ... #6178(1447) ... #6179(1447) ... #6180(1447) ... #6181(1447) ... #6182(1447) ... #6183(1447) ... #6184(1447) ... #6185(1447) ... #6186(1447) ... #6187(1447) ... #6188(1447) ... #6189(1447) ... #6190(1447) ... #6191(1447) ... #6192(1447) ... #6193(1447) ... #6194

Unusual outbound traffic with large data transfers to unknown IP's That could be suspicious. The utilization of ports on the network that are uncommon, high volume of DNS requests, encrypted traffic spikes, and repetitive connection attempts

d. What is a brute force attack, and how does it work?

A brute force attack is a type of attack which utilizes trial and error method by attackers to gain unauthorized access to a system by try all possible combinations of credentials of usernames and passwords. It keeps trying until the correct one is found and typically sends login requests to a target service like SSH or RDP usually using dictionary attacks.

e. What are some common strategies to defend against brute force attacks based on the information obtained from packet captures?

Probably one of the easiest ways to counter brute force attack is by rate limiting the amount of successive login attempts. Forcing users to wait just up to 5 seconds between logins significantly slows down the process for brute force attacks. Lockout policies for unsuccessful logins as well would work too. MFA is drastically helps too.

f. How can Wireshark help in detecting and analyzing brute force attacks on network services like SSH or RDP?

Wireshark can detect brute force attacks on services like SSH or RDP by capturing the network traffic and analyzing it for signs of high volume repeated login attempts from a single IP. Wireshark can identify the attackers IP and filter packets for tcp (port 22) and RDP (port 3389) as well for failed authentication responses or unusual packet rates indicated an attack.

Submission Instructions:

- Submit the screenshots and filled pdf of this document (with all the answers). Do not delete the questions or change the order of the questions. You can download and edit this document.
- Submit electronically through Canvas.
- Email or hardcopy submissions will not be accepted.

References:

<https://www.comparitech.com/net-admin/pcap-guide/#:~:text=To%20capture%20PCAP%20files%20you,could%20be%20eth0%20or%20wlan0>
This lab is Inspired from EscapeRoom CTF created by The Honeynet Project on the CyberDefenders website:
<https://cyberdefenders.org/blueteam-ctf-challenges/18#nav-overview>