



ARP Poisoning/Spoofing (A.K.A. Man in the Middle Attack)

- ARP (Address Resolution Protocol) allows devices to identify themselves on a network.
- APR Poisoning/Spoofing (MitM) is an attack that associates with network jamming or manipulating by sending a malicious ARP packet to the default gateway.
- The ARP Poisoning/Spoofing objective is to manipulate the IP to MAC address table and sniff traffic of the target host.

How to detect ARP attacks in Wireshark

ARP analysis

- Works on the local network
- Enables the communication between MAC addresses
- Not a secure protocol
- Not a routable protocol
- Common patterns are *request & response*, *announcement* and *gratuitous* packets

Notes	Wireshark filter
Global search	<ul style="list-style-type: none">• arp
"ARP" options for grabbing the low-hanging fruits: <ul style="list-style-type: none">• Opcode 1: <u>ARP</u> requests.• Opcode 2: <u>ARP</u> responses.• Hunt:<u>Arp</u> scanning• Hunt:Possible <u>ARP</u> poisoning detection• Hunt:Possible <u>ARP</u> flooding from detection:	<ul style="list-style-type: none">• arp.opcode == 1• arp.opcode == 2• arp.dst.hw_mac==00:00:00:00:00:00• arp.duplicate-address-detected or arp.duplicate-address-frame• ((arp) && (arp.opcode == 1)) && (arp.src.hw_mac == target-mac-address)

ARP Request

The screenshot shows a Wireshark capture of an ARP request packet. The packet list at the top shows two packets: packet 1 is the ARP request, and packet 2 is the ARP reply. The packet details pane for packet 1 is expanded, showing the Ethernet II header, the ARP request details, and the raw packet data. The ARP request details show the sender's MAC address (00:0c:29:e2:18:b4), sender's IP address (192.168.1.25), target's MAC address (00:00:00:00:00:00), and target's IP address (192.168.1.1). The packet bytes pane shows the raw data of the packet, with the ARP request details highlighted in red.

ARP Reply

The screenshot shows a Wireshark capture of an ARP reply packet. The packet list at the top shows two packets: packet 1 is the ARP request, and packet 2 is the ARP reply. The packet details pane for packet 2 is expanded, showing the Ethernet II header, the ARP reply details, and the raw packet data. The ARP reply details show the sender's MAC address (50:78:b3:f3:cd:f4), sender's IP address (192.168.1.1), target's MAC address (00:0c:29:e2:18:b4), and target's IP address (192.168.1.25). The packet bytes pane shows the raw data of the packet, with the ARP reply details highlighted in red.

- what makes this suspicious is noticing two different responses for the same IP address

The screenshot shows a Wireshark capture of an ARP spoofing attack. The packet list at the top shows three packets: packet 1 is the ARP request, packet 2 is the ARP reply, and packet 3 is another ARP reply. The packet details pane for packet 3 is expanded, showing the Ethernet II header, the ARP reply details, and the raw packet data. The ARP reply details show the sender's MAC address (00:0c:29:e2:18:b4), sender's IP address (192.168.1.1), target's MAC address (00:0c:29:98:c7:a8), and target's IP address (192.168.1.25). The packet bytes pane shows the raw data of the packet, with the ARP reply details highlighted in red. A red box highlights the text: "[Duplicate IP address detected for 192.168.1.1 (00:0c:29:e2:18:b4) - also in use by 50:78:b3:f3:cd:f4 (frame 2)]".

- As an analyst, it is critical to notate your findings before further investigating

Notes	Detection Notes	Findings
Possible IP address match.	1 IP address announced from a MAC address.	<ul style="list-style-type: none"> • MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.25
Possible <u>ARP</u> spoofing attempt.	2 MAC addresses claimed the same IP address (192.168.1.1). The "192.168.1.1" IP address is a possible gateway address.	<ul style="list-style-type: none"> • MAC1: 50:78:b3:f3:cd:f4 • MAC 2: 00:0c:29:e2:18:b4
Possible <u>ARP</u> flooding attempt.	The MAC address that ends with "b4" claims to have a different/new IP address.	<ul style="list-style-type: none"> • MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.1

- Further inspecting the traffic, we can view additional findings of the adversary.

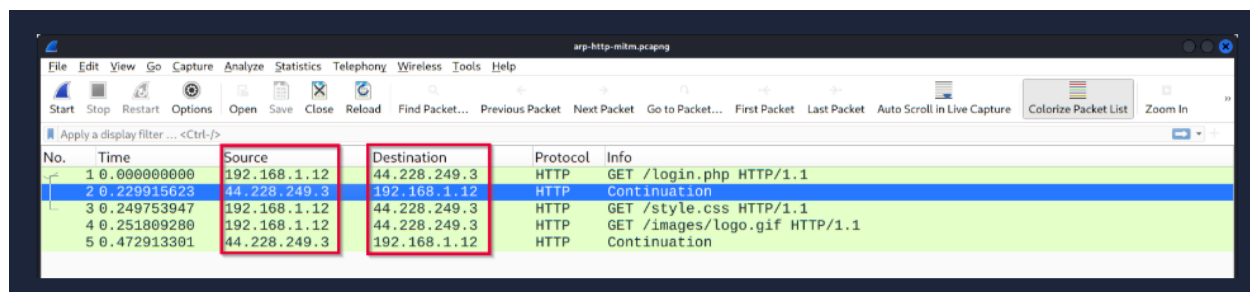
No.	Time	Source	Destination	Protocol	Info
1	0.000000000	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001059831	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4
3	0.010490253	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.37? Tell 192.168.1.25
4	0.020876839	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.158? Tell 192.168.1.25
5	0.031275021	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.212? Tell 192.168.1.25
6	0.041848453	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.176? Tell 192.168.1.25
7	0.052746298	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.73? Tell 192.168.1.25
8	0.063388601	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.216? Tell 192.168.1.25
9	0.073905794	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.181? Tell 192.168.1.25
10	0.084401792	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.217? Tell 192.168.1.25
11	0.095003040	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.173? Tell 192.168.1.25
12	0.105417559	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.136? Tell 192.168.1.25
13	0.115638938	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.132? Tell 192.168.1.25
14	0.125920898	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.130? Tell 192.168.1.25
15	0.136708415	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.254? Tell 192.168.1.25
16	0.147294383	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.232? Tell 192.168.1.25
17	0.157926474	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.162? Tell 192.168.1.25
18	0.168416850	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.109? Tell 192.168.1.25
19	0.178936116	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.253? Tell 192.168.1.25
20	0.189453050	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.169? Tell 192.168.1.25

- Notice there is a flood of request occurring from the same IP address
- Also notice the MAC address ending in “b4” is crafting multiple ARP request with the IP address “192.168.1.25”

Notes	Detection Notes	Findings
Possible IP address match.	1 IP address announced from a MAC address.	<ul style="list-style-type: none"> • MAC: 00:0c:29:e2:18:b4

		<ul style="list-style-type: none"> IP: 192.168.1.25
Possible <u>ARP</u> spoofing attempt.	<p>2 MAC addresses claimed the same IP address (192.168.1.1). The " 192.168.1.1" IP address is a possible gateway address.</p>	<ul style="list-style-type: none"> MAC1: 50:78:b3:f3:cd:f4 MAC 2: 00:0c:29:e2:18:b4
Possible <u>ARP</u> spoofing attempt.	The MAC address that ends with "b4" claims to have a different/new IP address.	<ul style="list-style-type: none"> MAC: 00:0c:29:e2:18:b4 IP: 192.168.1.1
Possible <u>ARP</u> flooding attempt.	The MAC address that ends with "b4" crafted multiple <u>ARP</u> requests against a range of IP addresses.	<ul style="list-style-type: none"> MAC: 00:0c:29:e2:18:b4 IP: 192.168.1.xxx

- From here, we can conclude that the MAC address owns the IP address “192.168.1.25” and initiated suspicious ARP requests against numerous IP addresses.
- In addition, this MAC address ending in “b4” also has a gateway address.



- As we see here, the traffic looks normal, but don't let the adversary fool the eyes. Wireshark detects everything that is occurring within the network
- Here we will add the mac address column to further investigate.

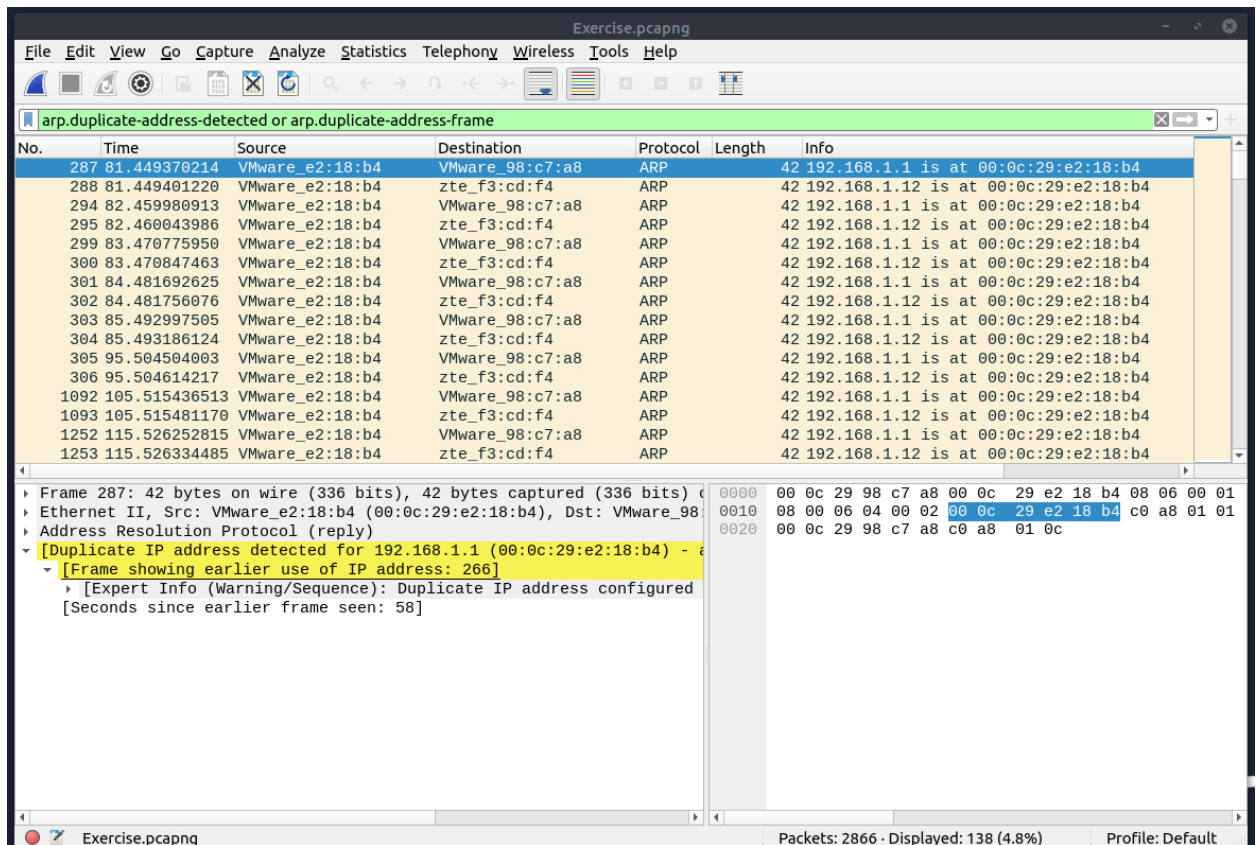
No.	Time	Source	Source	Destination	Destination	Protocol	Info
1	0.000000000	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /login.php HTTP/1.1
2	0.229915623	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP	Continuation
3	0.249753947	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /style.css HTTP/1.1
4	0.251809280	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /images/logo.gif HTTP/1.1
5	0.472913301	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP	Continuation

- Notice the same MAC address ending in “b4” is the destination of all the http packets.
- This assures us that there is a MitM attack occurring.
- The adversary is the host with the MAC address ending in “b4” and all the traffic linked to the IP address “192.168.1.12”.

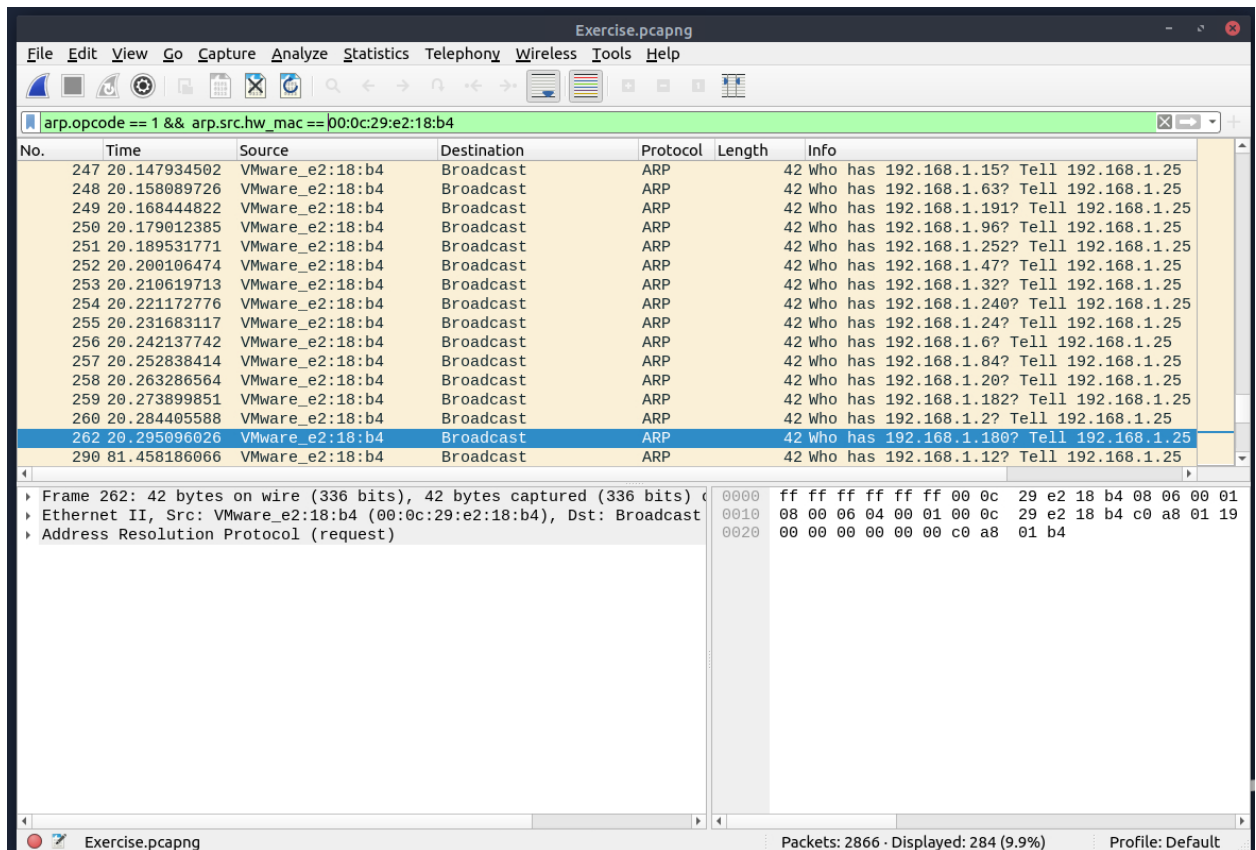
Notes	Detection Notes	Findings
IP to MAC matches.	3 IP to MAC address matches.	<ul style="list-style-type: none"> MAC: 00:0c:29:e2:18:b4 = IP: 192.168.1.25 MAC: 50:78:b3:f3:cd:f4 = IP: 192.1681.1 MAC: 00:0c:29:98:c7:a8 = IP: 192.168.1.12
Attacker	The attacker created noise with <u>ARP</u> packets.	<ul style="list-style-type: none"> MAC: 00:0c:29:e2:18:b4 = IP: 192.168.1.25
Router/gateway	Gateway address.	<ul style="list-style-type: none"> MAC: 50:78:b3:f3:cd:f4 = IP: 192.1681.1
Victim	The attacker sniffed all traffic of the victim.	<ul style="list-style-type: none"> MAC: 50:78:b3:f3:cd:f4 = IP: 192.1681.12

1. What is the number of ARP requests crafted by the attacker?

- The first step is to see who the attacker is. We can do this by inputting a filter:
- **arp.duplicate-address-detected or arp.duplicate-address-frame** <--- this filter is used to identify conflicting IP addresses. This is when Wireshark is scanning multiple devices that claim the same network.



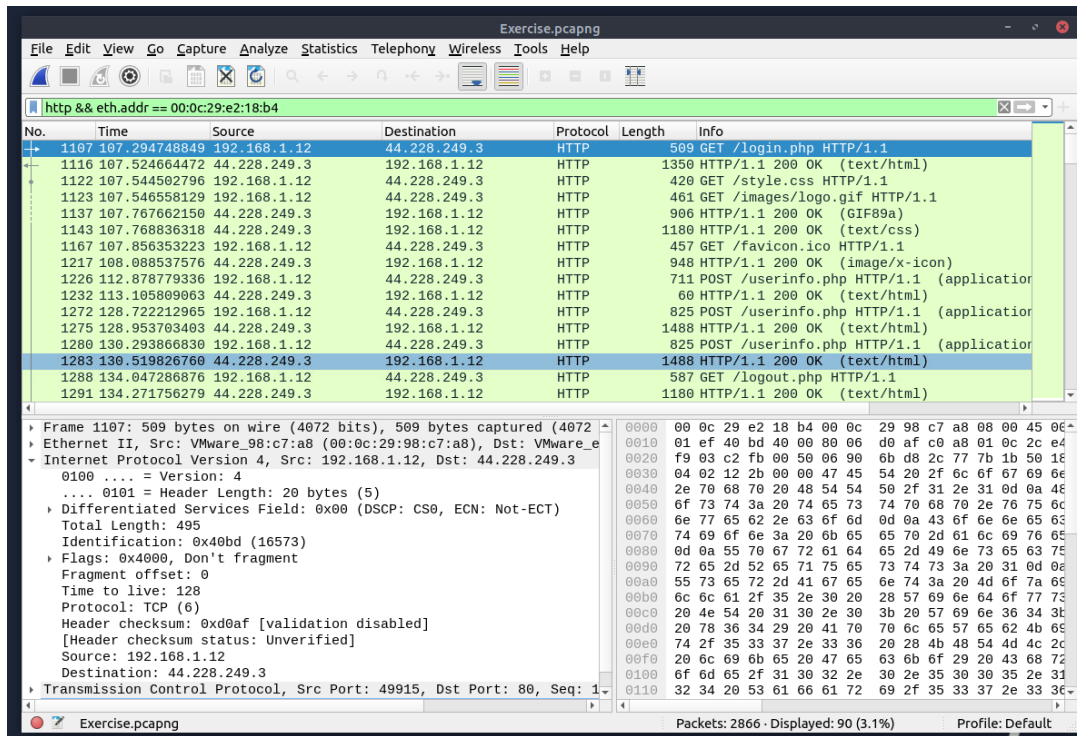
- Highlighted in yellow, there is a MAC address that is causing the duplication.
- From here we can now input a filter that can determine how many requests were made by that particular MAC address “00:0c:29:e2:18:b4”
- `arp.opcode == 1 && arp.src.hw_mac == 00:0c:29:e2:18:b4` <--- this filter is telling Wireshark to look for arp request “`arp.opcode == 1`” that is linked to the mac address “`arp.src.hw_mac == 00:0c:29:e2:18:b4`”



Answer: 284

2. What is the number of HTTP packets received by the attacker?

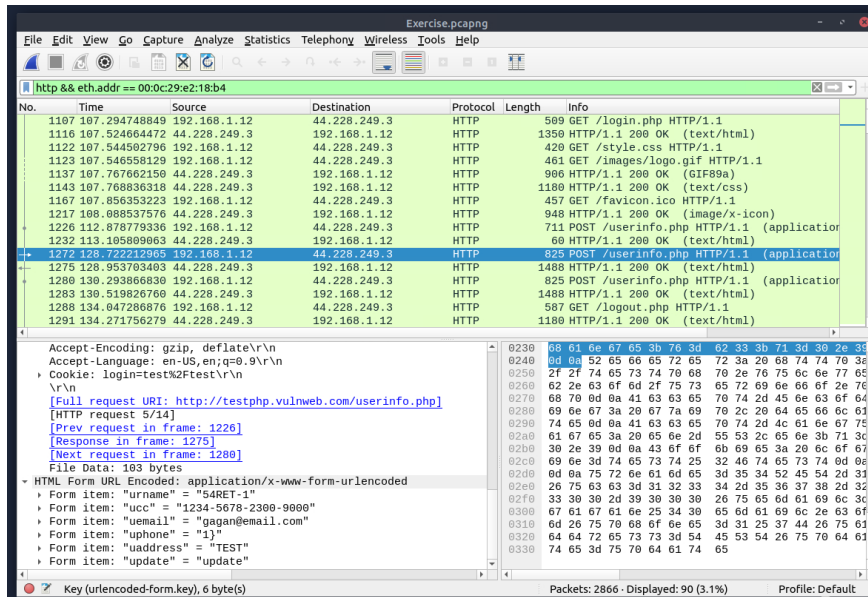
- We are looking for only http packets from the adversary with the MAC address **00:0c:29:e2:18:b4**
- By determining the number of http packets, we must use a filter input to retrieve all the http packets from the adversary MAC address.
- **http && eth.addr == 00:0c:29:e2:18:b4** <--- this is telling Wireshark to look for http packets that are generated from the MAC address **00:0c:29:e2:18:b4**



Answer: 90

3. What is the number of sniffed username & password entries?

- Notice in the screenshot above there are some POST info within the packet sniffer. Let's observe what we can find.

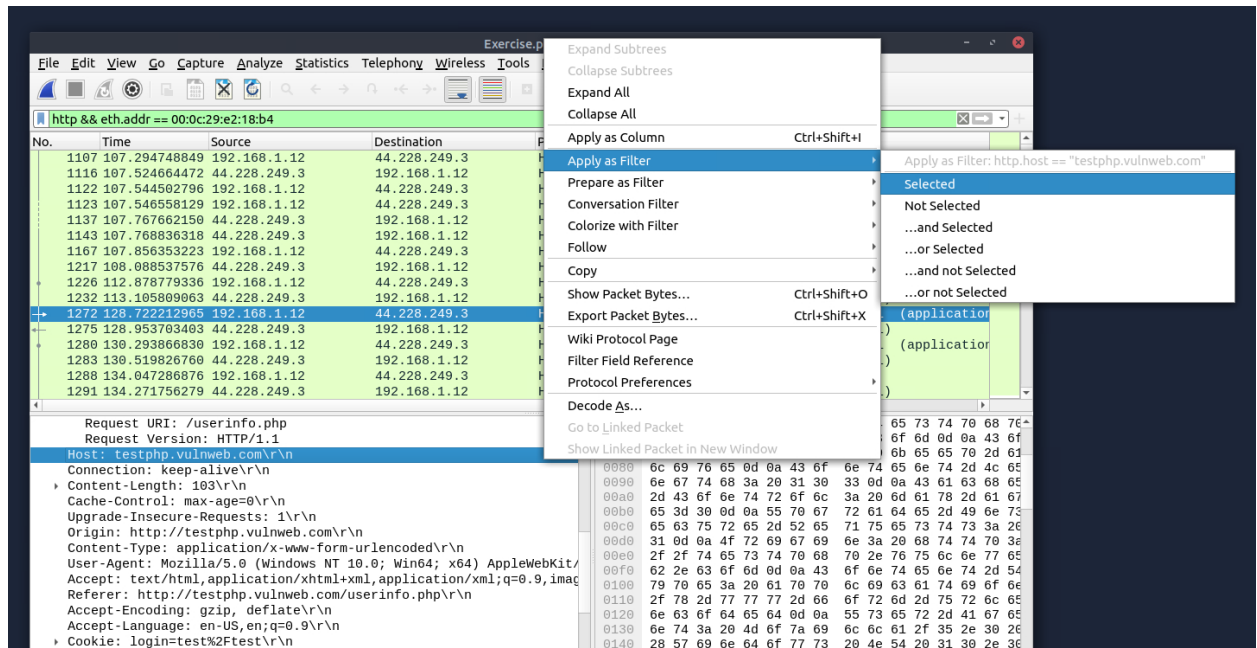


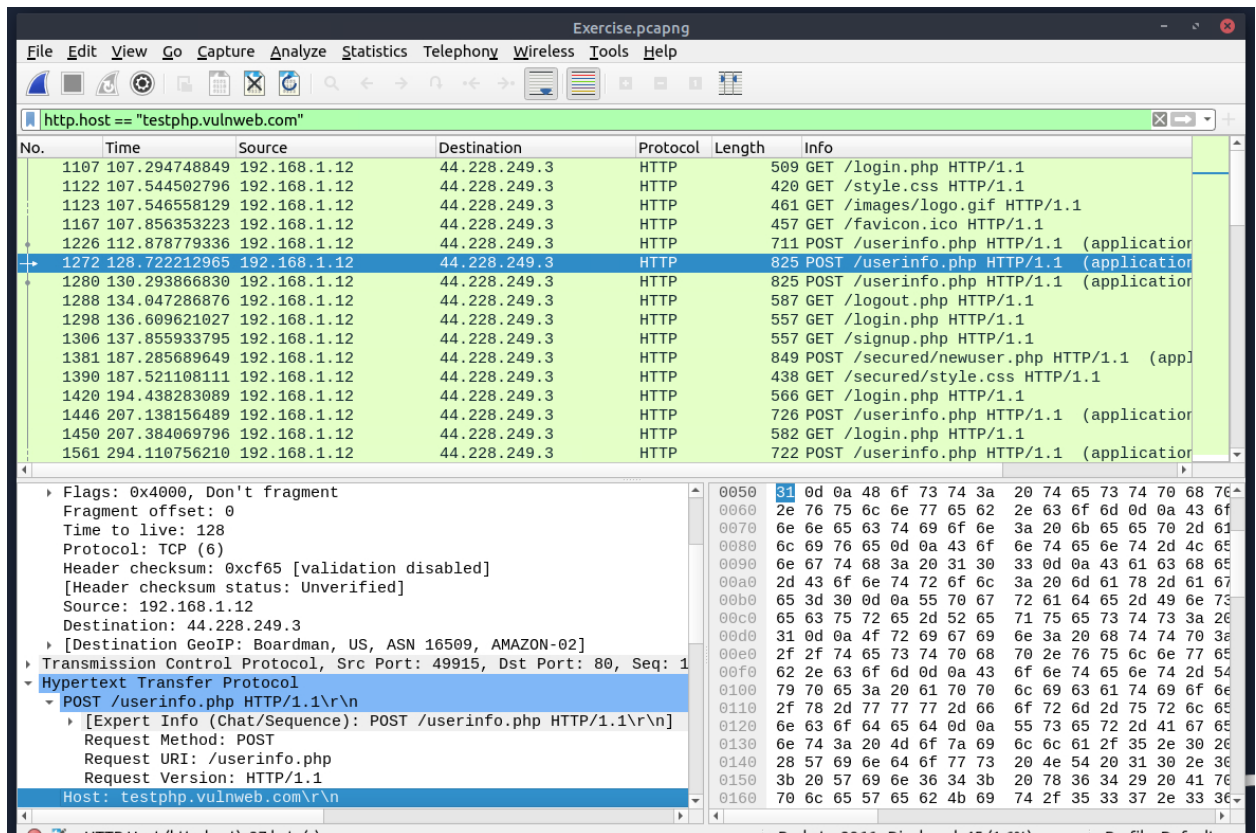
Answer: 6

- Looking at the “userinfo.php ” POST response, I was able to identify a few username and passwords.

4. What is the password of the “Client986”?

- This question consists of several steps. First, we need to filter the host name to be able to have Wireshark generate POST request for the host.





- Next, we want to generate all http “POST” requests.
- If we review the “HTTP” protocol. We can use the `http.request.method == POST`
- `http.host == testphp.vulnweb.com && http.request.method == POST` <--- This filter is telling Wireshark to look for only POST request for the host of “testphp.vulnweb.com”

Exercise.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.host == testphp.vulnweb.com && http.request.method == POST

No.	Time	Source	Destination	Protocol	Length	Info
1226	112.878779336	192.168.1.12	44.228.249.3	HTTP	711	POST /userinfo.php HTTP/1.1 (application/x-www
1272	128.722212965	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www
1280	130.293866830	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www
1381	187.285689649	192.168.1.12	44.228.249.3	HTTP	849	POST /secured/newuser.php HTTP/1.1 (applicati
1446	207.138156489	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www
1561	294.110756210	192.168.1.12	44.228.249.3	HTTP	722	POST /userinfo.php HTTP/1.1 (application/x-www
1599	334.960960385	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www
1668	354.682038726	192.168.1.12	44.228.249.3	HTTP	728	POST /userinfo.php HTTP/1.1 (application/x-www
1791	443.146852729	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www
2320	618.814163954	192.168.1.12	44.228.249.3	HTTP	787	POST /comment.php HTTP/1.1 (application/x-www

Flags: 0x4000, Don't fragment
Fragment offset: 0
Time to live: 128
Protocol: TCP (6)
Header checksum: 0xc65 [validation disabled]
[Header checksum status: Unverified]
Source: 192.168.1.12
Destination: 44.228.249.3
[Destination GeoIP: Boardman, US, ASN 16509, AMAZON-02]
Transmission Control Protocol, Src Port: 49915, Dst Port: 80, Seq: 1

Hypertext Transfer Protocol

POST /userinfo.php HTTP/1.1\r\n

[Expert Info (Chat/Sequence): POST /userinfo.php HTTP/1.1\r\n]
Request Method: POST
Request URI: /userinfo.php
Request Version: HTTP/1.1
Host: testphp.vulnweb.com\r\n

HTTP Host (http.host), 27 byte(s)

Packets: 2866 · Displayed: 10 (0.3%) Profile: Default

- We have 10 results.
- Lets comb through the results and see what we can find that associate with the user “Client986”

Exercise.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.host == testphp.vulnweb.com && http.request.method == POST

No.	Time	Source	Destination	Protocol	Length	Info
1226	112.878779336	192.168.1.12	44.228.249.3	HTTP	711	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1272	128.722212965	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1280	130.293866830	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1381	187.285689649	192.168.1.12	44.228.249.3	HTTP	849	POST /secured/newuser.php HTTP/1.1 (application/x-www-form-urlencoded)
1446	207.138156489	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1561	294.110756210	192.168.1.12	44.228.249.3	HTTP	722	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1599	334.960960385	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1668	354.682038726	192.168.1.12	44.228.249.3	HTTP	728	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1791	443.146852729	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
2320	618.814163954	192.168.1.12	44.228.249.3	HTTP	787	POST /comment.php HTTP/1.1 (application/x-www-form-urlencoded)

Origin: http://testphp.vulnweb.com\r\n
Content-Type: application/x-www-form-urlencoded\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.105 Safari/537.36\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\r\n
Referer: http://testphp.vulnweb.com/login.php\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/userinfo.php]
[HTTP request 5/6]
[Prev request in frame: 1605]
[Response in frame: 1670]
[Next request in frame: 1672]
File Data: 37 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

- Form item: "uname" = "client986"
- Form item: "pass" = "clientnothere!"

HTTP Host (http host) 27 bytes(s)

Packets: 2866. Displayed: 10 (0.3%) Profile: Default

Answer: clientnothere!

- Packet number 7 have a username "Client986"
- We have found our password

5. What is the comment provided by the "Client354"?

- Let's click on the comment to determine the output.

Exercise.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.host == testphp.vulnweb.com && http.request.method == POST

No.	Time	Source	Destination	Protocol	Length	Info
1226	112.878779336	192.168.1.12	44.228.249.3	HTTP	711	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1272	128.722212965	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1280	130.293866830	192.168.1.12	44.228.249.3	HTTP	825	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1381	187.285689649	192.168.1.12	44.228.249.3	HTTP	849	POST /secured/newuser.php HTTP/1.1 (application/x-www-form-urlencoded)
1446	207.138156489	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1561	294.110756210	192.168.1.12	44.228.249.3	HTTP	722	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1599	334.960960385	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1668	354.682038726	192.168.1.12	44.228.249.3	HTTP	728	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
1791	443.146852729	192.168.1.12	44.228.249.3	HTTP	726	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
2320	618.814163954	192.168.1.12	44.228.249.3	HTTP	787	POST /comment.php HTTP/1.1 (application/x-www-form-urlencoded)

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.149 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://testphp.vulnweb.com/comment.php?pid=7
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
[Full request URI: http://testphp.vulnweb.com/comment.php]
[HTTP request 3/8]
[Prev request in frame: 2259]
[Response in frame: 2323]
[Next request in frame: 2352]
File Data: 89 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "name" = "client354"
Form item: "comment" = "Nice work!"
Form item: "Submit" = "Submit"
Form item: "phpaction" = "echo \$_POST[comment];"

HTTP Host (http.host), 27 byte(s)

Packets: 2866 · Displayed: 10 (0.3%) Profile: Default

Answer: Nice work!