



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

## ARP Request

arp.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Start Stop Restart Options Open Save Close Reload Find Packet... Previous Packet Next Packet Go to Packet... First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In

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

No.	Time	Source	Destination	Protocol	Info
1	0.0000000000	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

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)  
Ethernet II, Src: VMware\_e2:18:b4 (00:0c:29:e2:18:b4), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Address Resolution Protocol (request)  
Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: request (1)  
Sender MAC address: 00:0c:29:e2:18:b4  
Sender IP address: 192.168.1.25  
Target MAC address: 00:00:00:00:00:00  
Target IP address: 192.168.1.1

## ARP Reply

arp.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Start Stop Restart Options Open Save Close Reload Find Packet... Previous Packet Next Packet Go to Packet... First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In

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

No.	Time	Source	Destination	Protocol	Info
1	0.0000000000	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

Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)  
Ethernet II, Src: 50:78:b3:f3:cd:f4, Dst: 00:0c:29:e2:18:b4  
Address Resolution Protocol (reply)  
Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: reply (2)  
Sender MAC address: 50:78:b3:f3:cd:f4  
Sender IP address: 192.168.1.1  
Target MAC address: 00:0c:29:e2:18:b4  
Target IP address: 192.168.1.25

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

arp-spoof.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Start Stop Restart Options Open Save Close Reload Find Packet... Previous Packet Next Packet Go to Packet... First Packet Last Packet Auto Scroll in Live Capture Colorize Packet List Zoom In

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

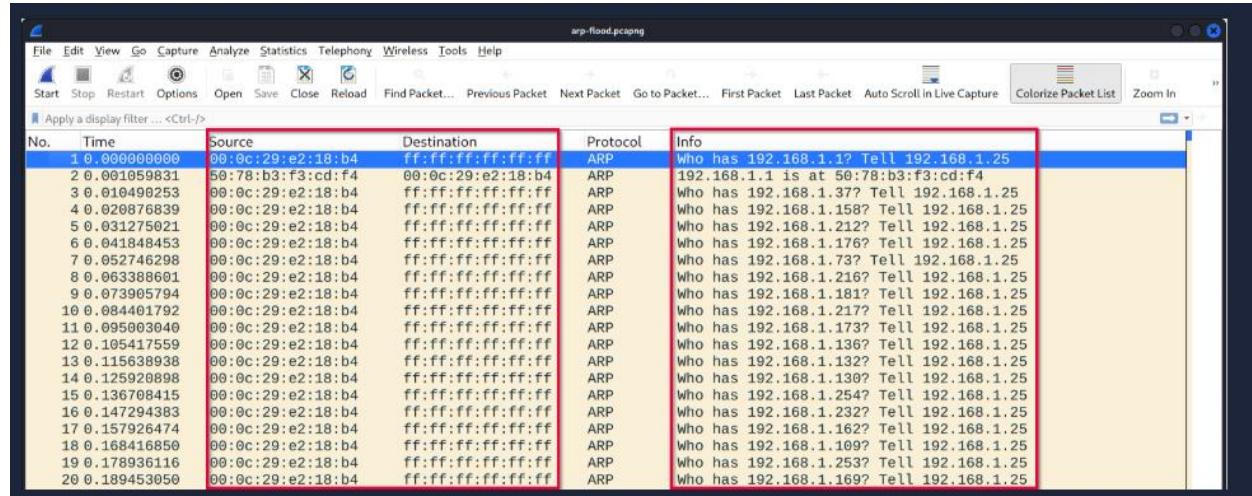
No.	Time	Source	Destination	Protocol	Info
1	0.0000000000	00:0c:29:e2:18:b4	50:78:b3:f3:cd:f4	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001271501	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4
3	0.393554684	00:0c:29:e2:18:b4	00:0c:29:98:c7:a8	ARP	192.168.1.1 is at 00:0c:29:e2:18:b4

Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0  
Ethernet II, Src: 00:0c:29:e2:18:b4, Dst: 00:0c:29:98:c7:a8  
Address Resolution Protocol (reply)  
Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: reply (2)  
Sender MAC address: 00:0c:29:e2:18:b4  
Sender IP address: 192.168.1.1  
Target MAC address: 00:0c:29:98:c7:a8  
Target IP address: 192.168.1.12  
[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 note 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"> <li>• MAC: 00:0c:29:e2:18:b4</li> <li>• IP: 192.168.1.25</li> </ul>
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"> <li>• MAC1: 50:78:b3:f3:cd:f4</li> <li>• MAC 2: 00:0c:29:e2:18:b4</li> </ul>
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"> <li>• MAC: 00:0c:29:e2:18:b4</li> <li>• IP: 192.168.1.1</li> </ul>

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

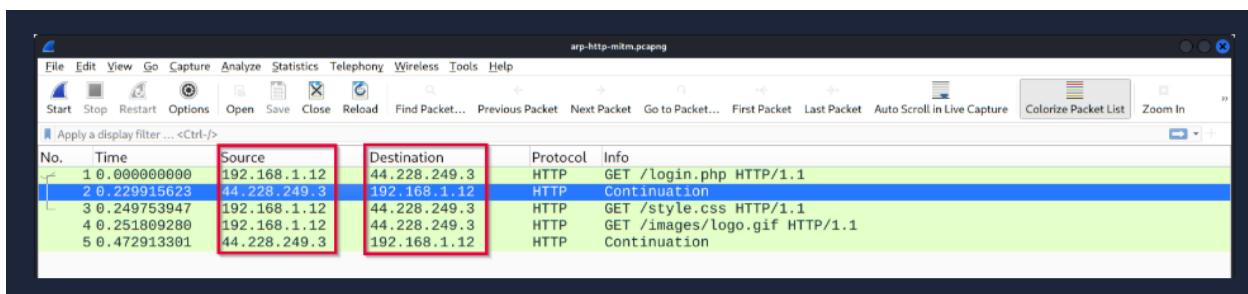


- Notice there is a flood of request occurring from the same IP address
- Also notice the MAC address ending in "b4" is crafting multiple ARP request with the IP address "192.168.1.25"

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

		<ul style="list-style-type: none"> <li>IP: 192.168.1.25</li> </ul>
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"> <li>MAC1: 50:78:b3:f3:cd:f4</li> <li>MAC 2: 00:0c:29:e2:18:b4</li> </ul>
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"> <li>MAC: 00:0c:29:e2:18:b4</li> <li>IP: 192.168.1.1</li> </ul>
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"> <li>MAC: 00:0c:29:e2:18:b4</li> <li>IP: 192.168.1.xxx</li> </ul>

- 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.

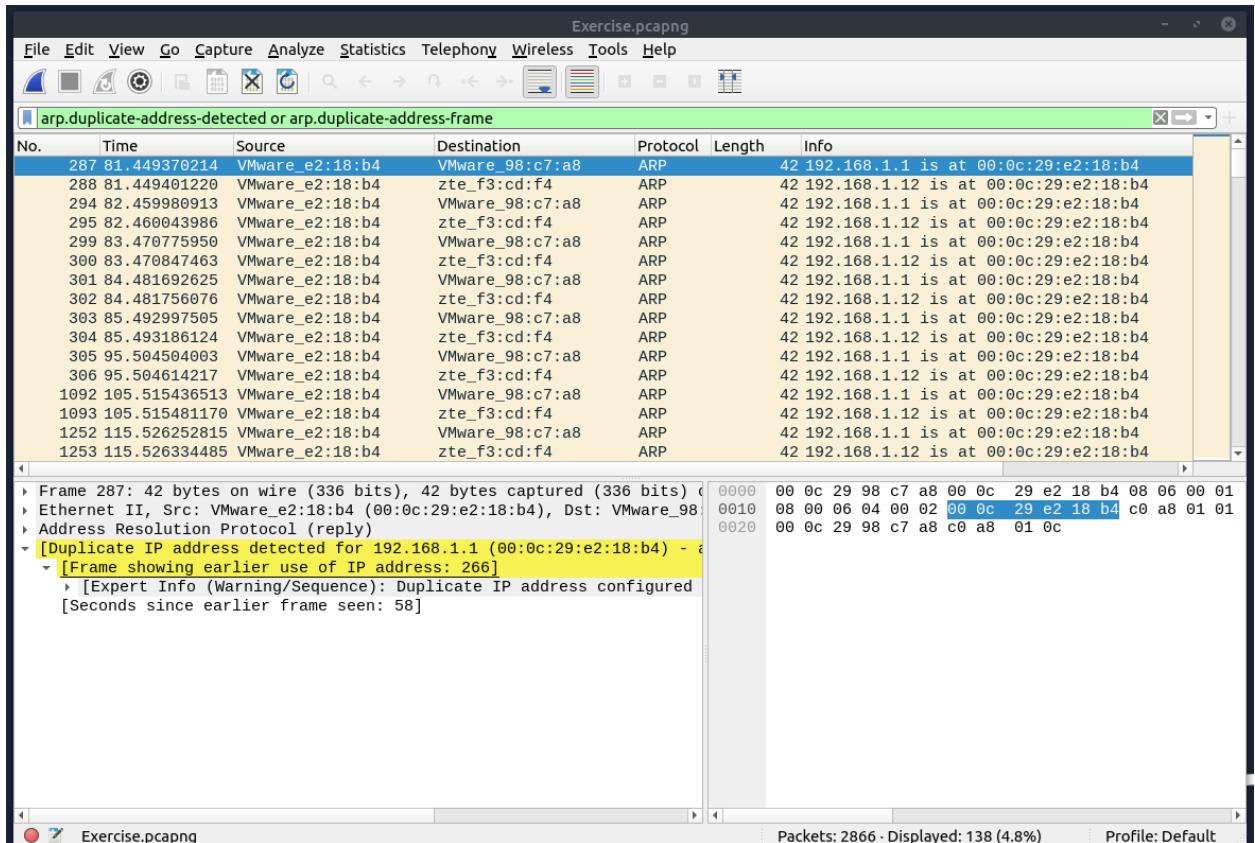
arp-http-mitm.pcapng						
No.	Time	Source	Source	Destination	Destination	Protocol
1	0.000000000	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP
2	0.229915623	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP
3	0.249753947	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP
4	0.251809280	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP
5	0.472913301	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP
				192.168.1.12	00:0c:29:e2:18:b4	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”.

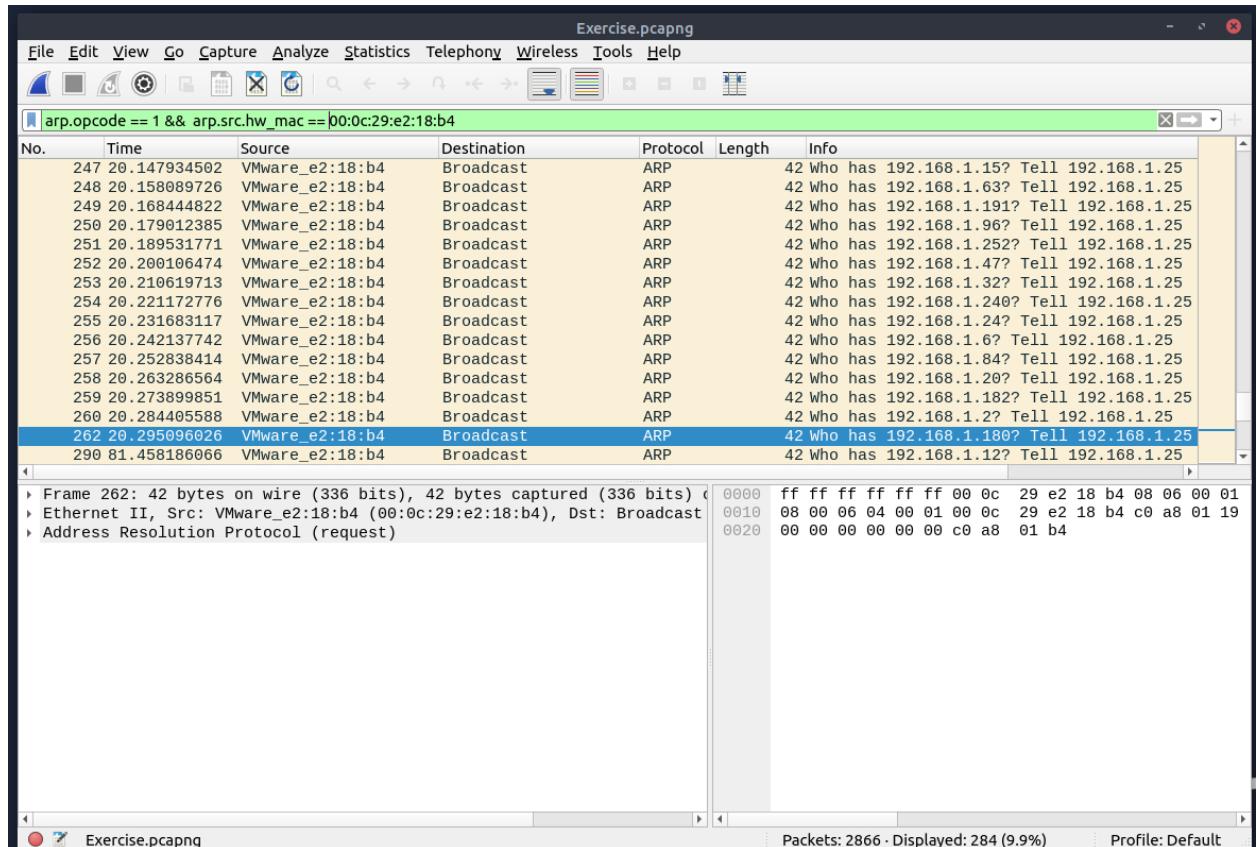
Notes	Detection Notes	Findings
IP to MAC matches.	3 IP to MAC address matches.	<ul style="list-style-type: none"> <li>• MAC: 00:0c:29:e2:18:b4 = IP: 192.168.1.25</li> <li>• MAC: 50:78:b3:f3:cd:f4 = IP: 192.168.1.1</li> <li>• MAC: 00:0c:29:98:c7:a8 = IP: 192.168.1.12</li> </ul>
Attacker	The attacker created noise with <u>ARP</u> packets.	<ul style="list-style-type: none"> <li>• MAC: 00:0c:29:e2:18:b4 = IP: 192.168.1.25</li> </ul>
Router/gateway	Gateway address.	<ul style="list-style-type: none"> <li>• MAC: 50:78:b3:f3:cd:f4 = IP: 192.168.1.1</li> </ul>
Victim	The attacker sniffed all traffic of the victim.	<ul style="list-style-type: none"> <li>• MAC: 50:78:b3:f3:cd:f4 = IP: 192.168.1.12</li> </ul>

### 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.



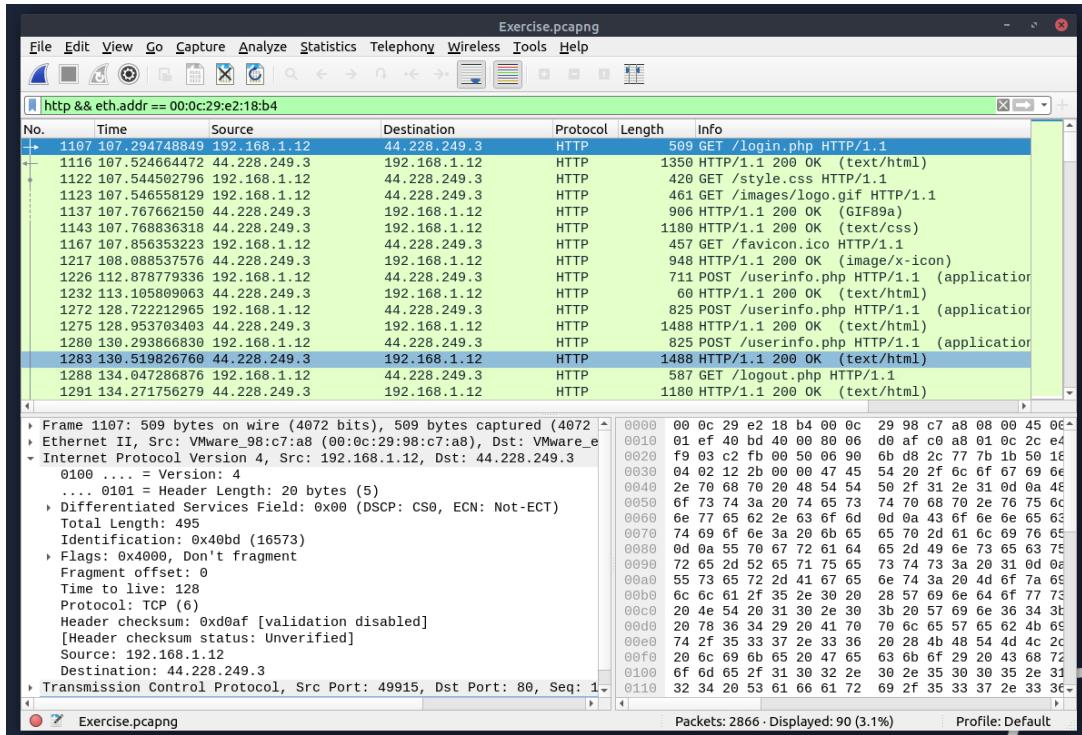
- Highlighted in yellow, there is a MAC address that is causing the duplication.
- From here we can now input a filter than can determine how many requests was made by that particular MAC address “00:0c:29:e2:18:b4”
- **arp.opcode ==1 && arp.src.hw\_mac == 00:0c:29:e2:18:b4** <--- this filter is telling 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?

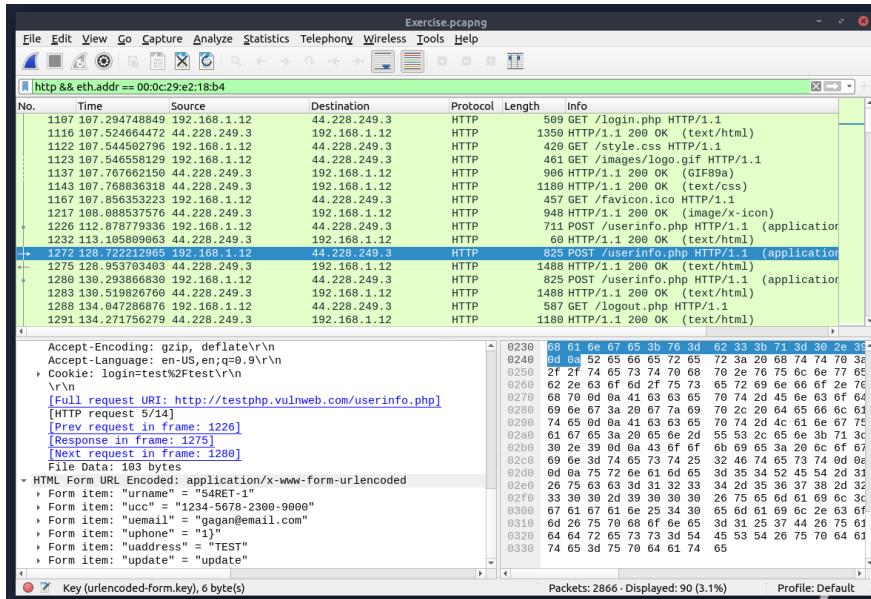
- We are looking for only http packets from the adversary with the MAC address **00:0c:29:e2:18:b4**
- By determining the number of http packets, we must use a filter input to retrieve all the http packets from the adversary MAC address.
- **http && eth.addr == 00:0c:29:e2:18:b4** <--- this is telling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

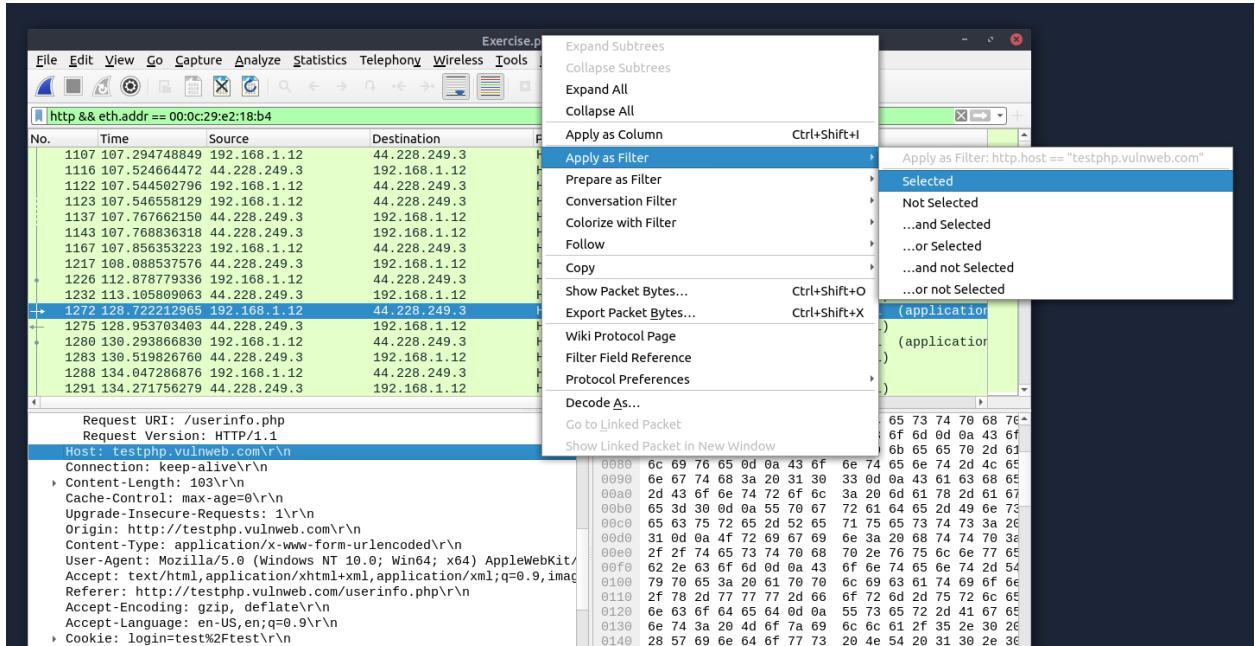


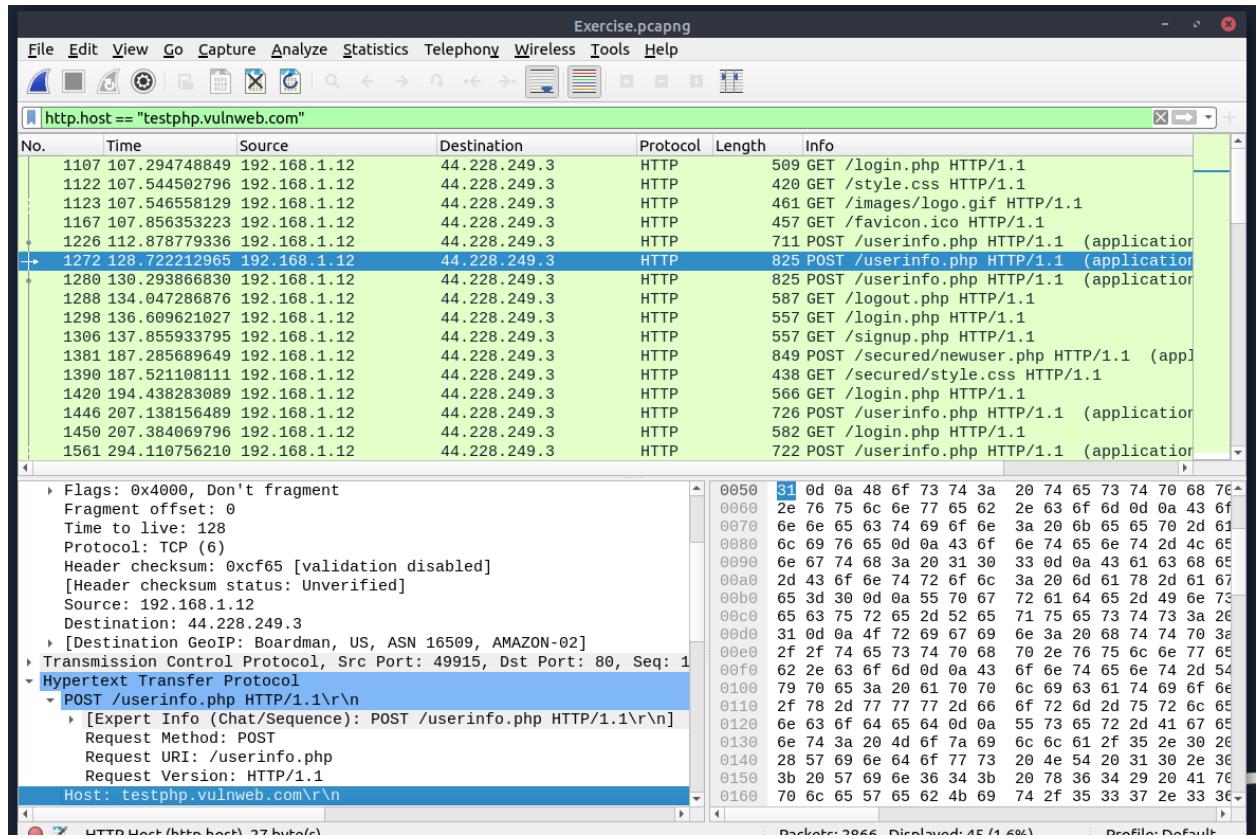
Answer: 6

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

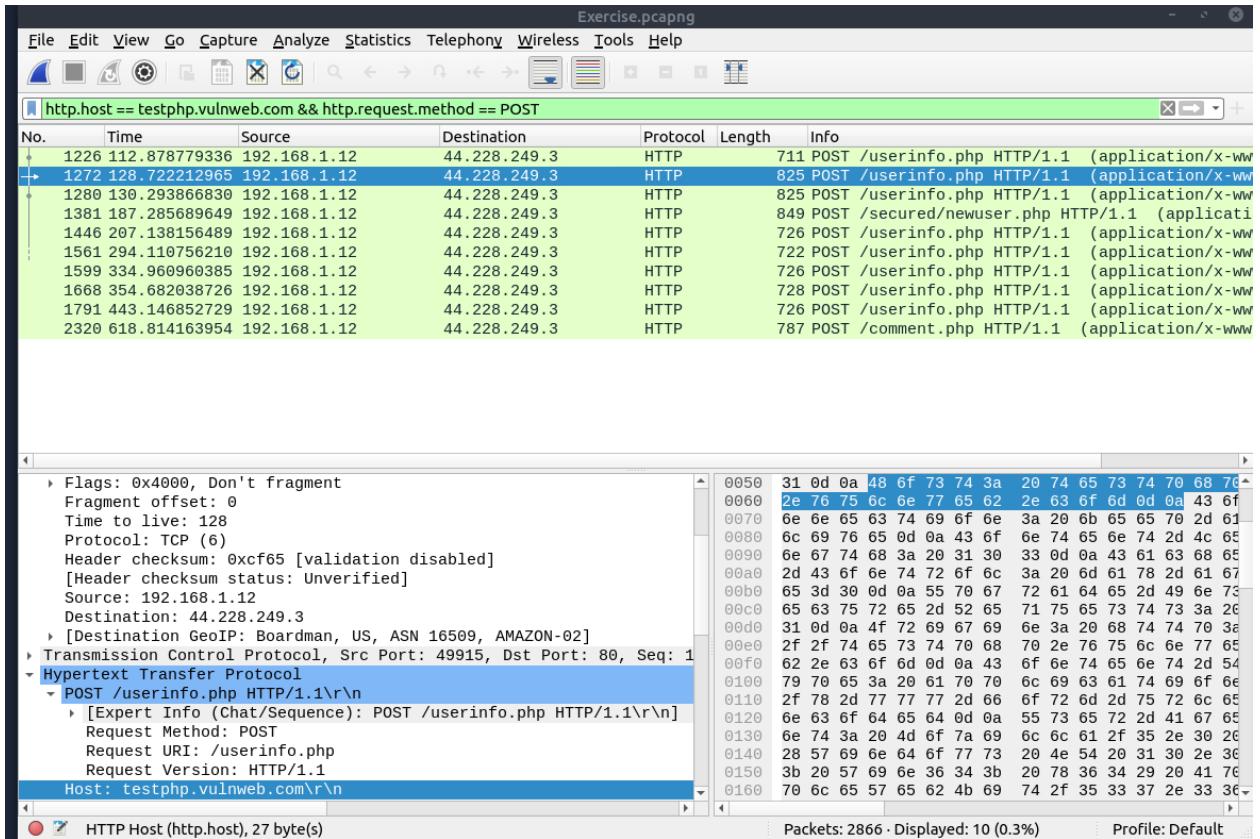
#### 4. What is the password of the “Client986”?

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

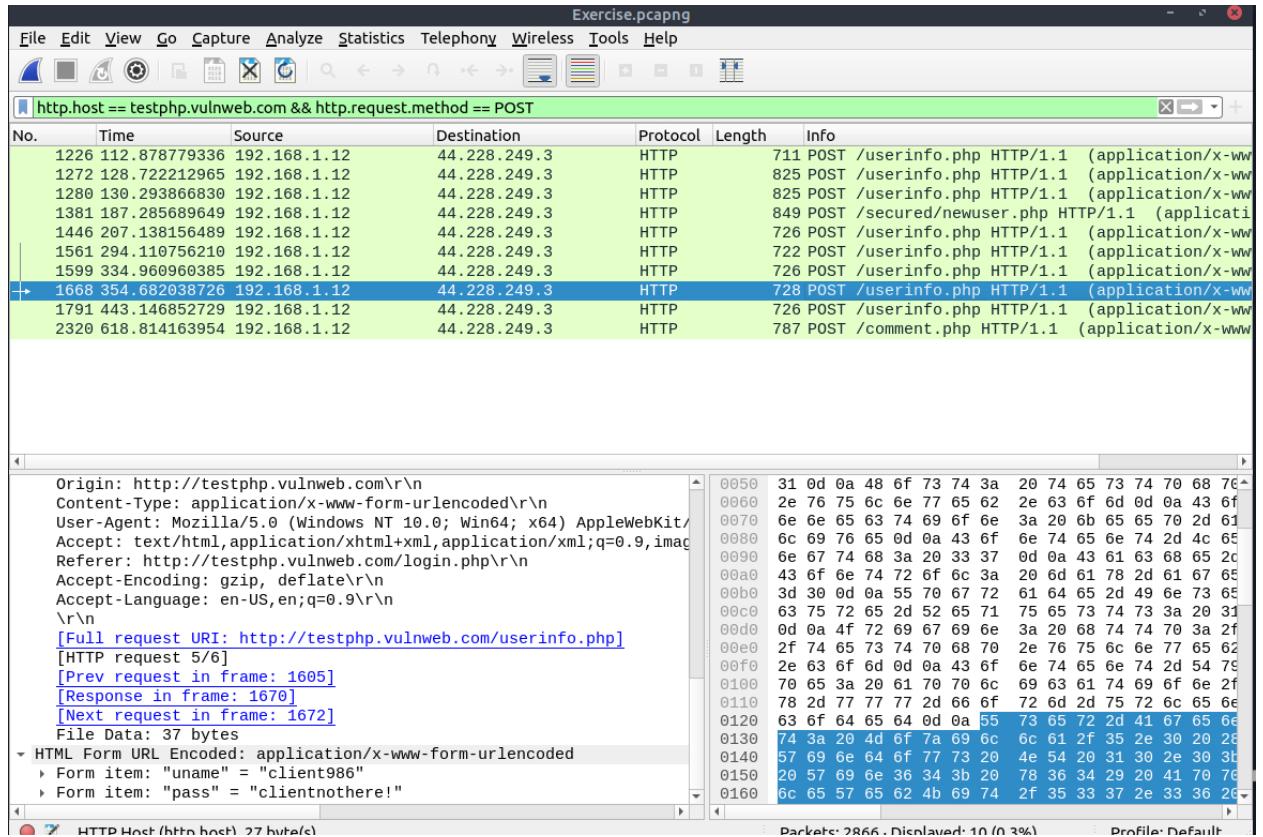




- Next, we want to generate all http “POST” requests.
- If we review the “HTTP” protocol. We can use the `http.request.method == POST`
- `http.host == testphp.vulnweb.com && http.request.method == POST` <--- This filter is telling Wireshark to look for only POST request for the host of “testphp.vulnweb.com”



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

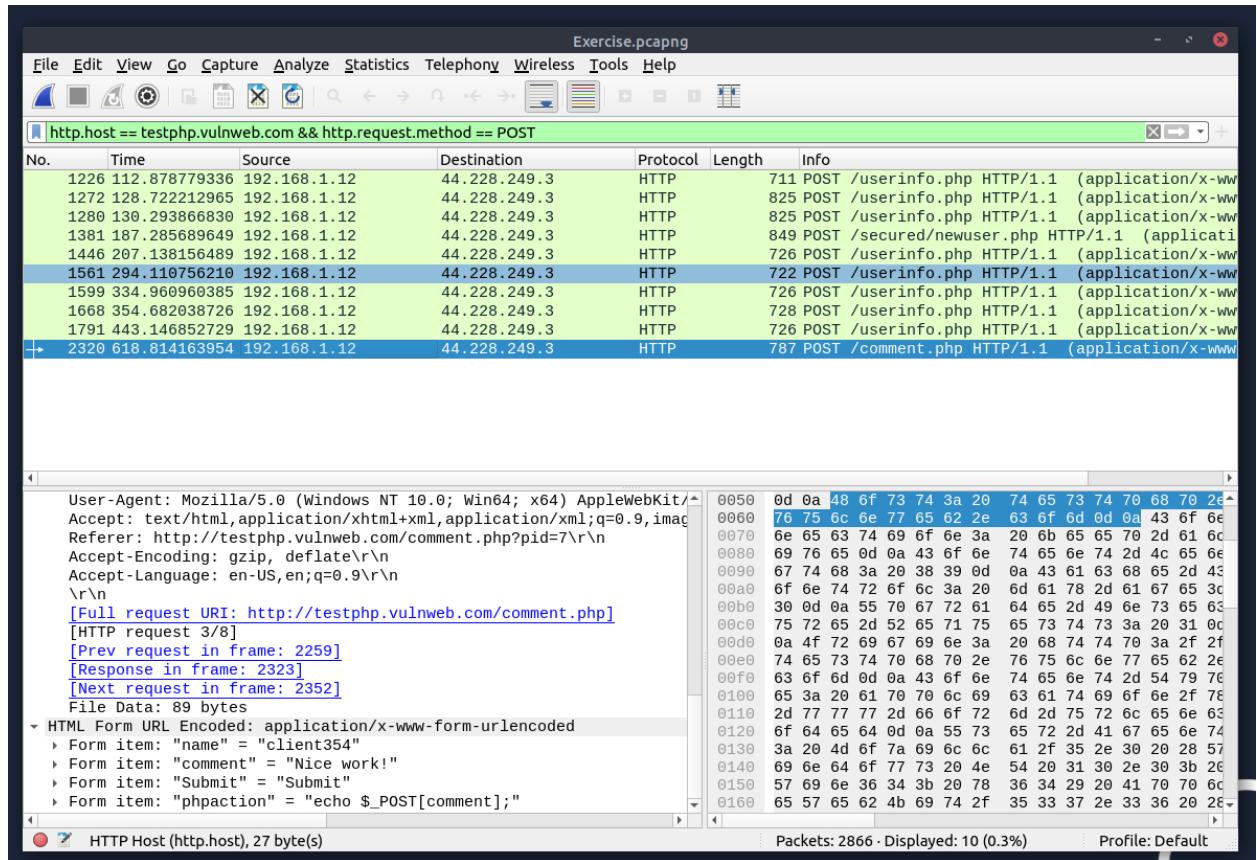


Answer: clientnothere!

- 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.



Answer: Nice work!