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.

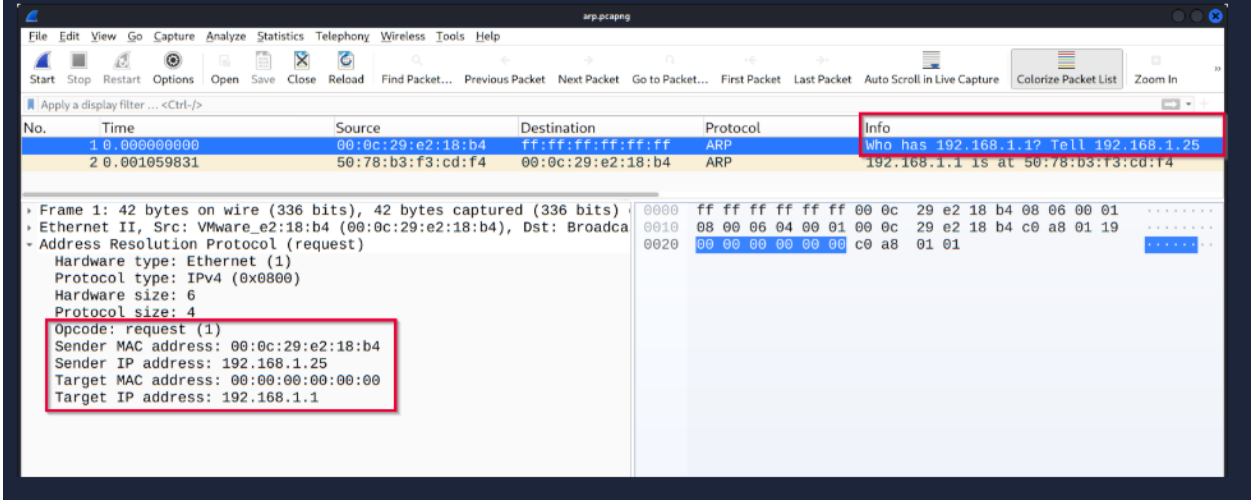
How to detect ARP attacks in Wireshark

ARP analysis

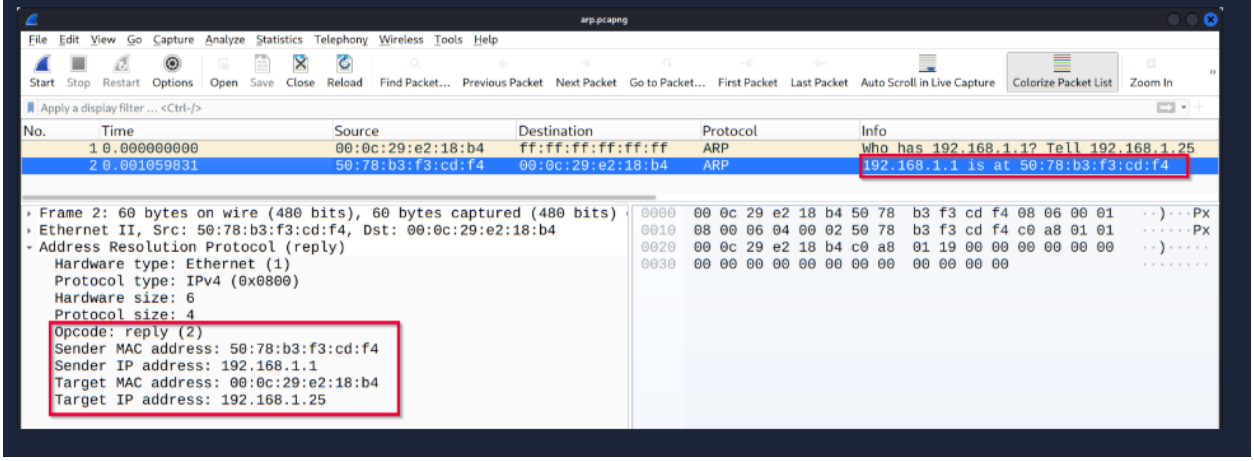
- Works on the local network
- Enables the communication between MAC addresses
- Not a secure protocol
- Not a routable protocol
- Common patterns are *request* & *response*, *announcement* and *gratuitous* packets

Notes	Wireshark filter
Global search	• `arp`
"ARP" options for grabbing the low-hanging fruits: • Opcode 1: ARP requests. • Opcode 2: ARP responses. • **Hunt:**Arp scanning • **Hunt:**Possible ARP poisoning detection • **Hunt:**Possible ARP flooding from detection:	• `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)`

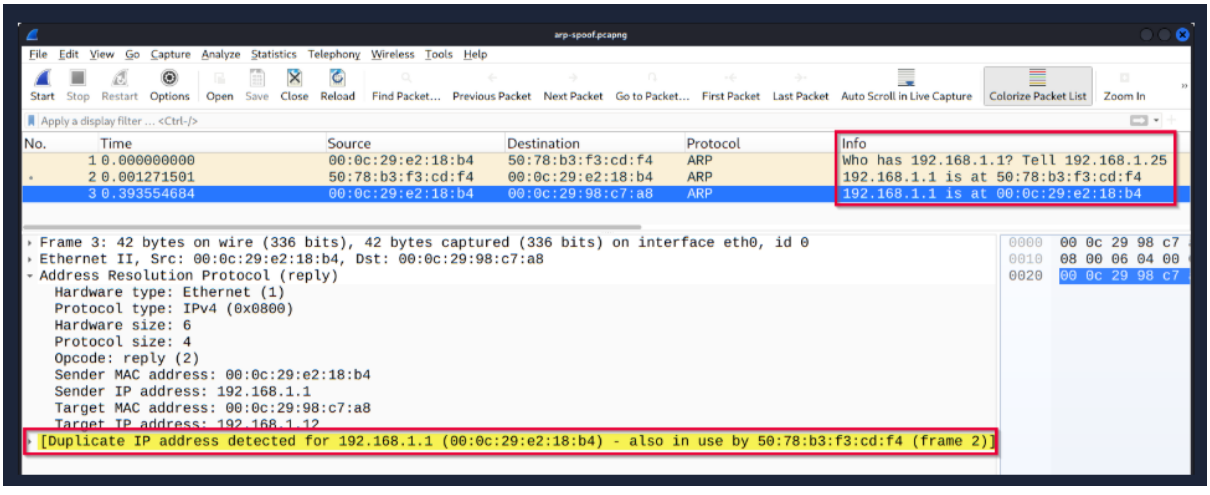
ARP Request



ARP Reply



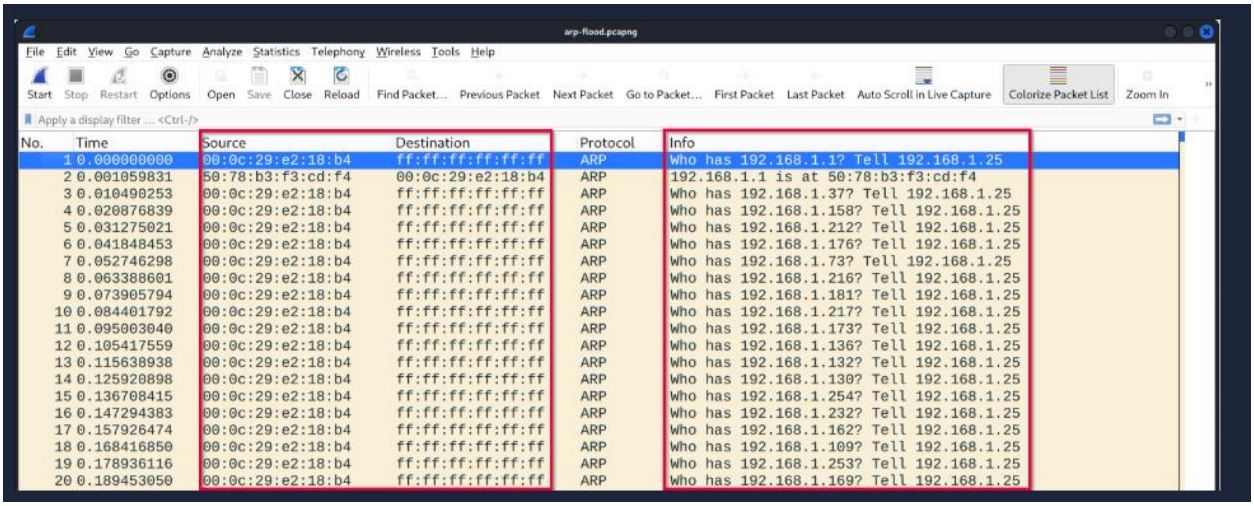
- what makes this suspicious is noticing two different responses for the same IP address



- 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.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.25
Possible ARP 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.	• MAC1: 50:78:b3:f3:cd:f4 • MAC 2: 00:0c:29:e2:18:b4
Possible ARP flooding attempt.	The MAC address that ends with "b4" claims to have a different/new IP address.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.1

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

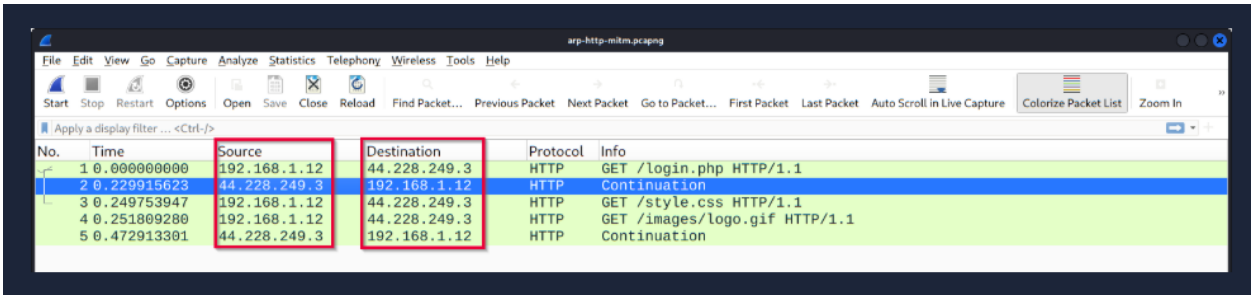


- Notice there is a flood of request occurring from the same IP address
- Also notice the MAC address ending in “b4” is crafting multiple ARP request with the IP address “192.168.1.25”

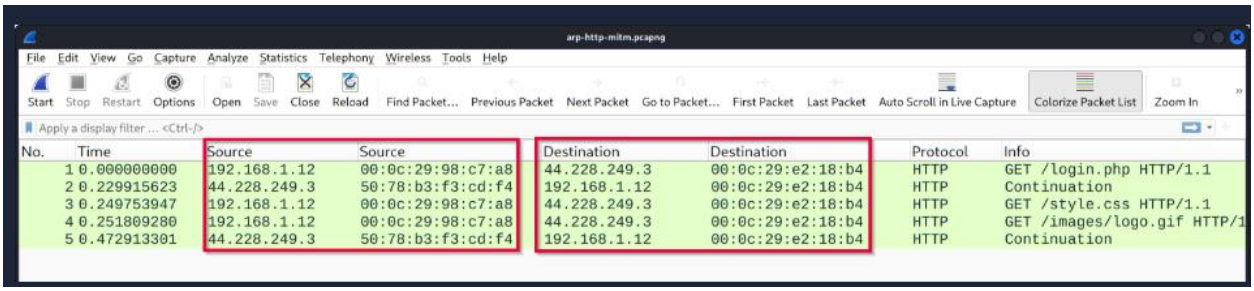
Notes	Detection Notes	Findings
Possible IP address match.	1 IP address announced from a MAC address.	• MAC: 00:0c:29:e2:18:b4

		• IP: 192.168.1.25
Possible ARP 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.	• MAC1: 50:78:b3:f3:cd:f4 • MAC 2: 00:0c:29:e2:18:b4
Possible ARP spoofing attempt.	The MAC address that ends with "b4" claims to have a different/new IP address.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.1
Possible ARP flooding attempt.	The MAC address that ends with "b4" crafted multiple ARP requests against a range of IP addresses.	• 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.



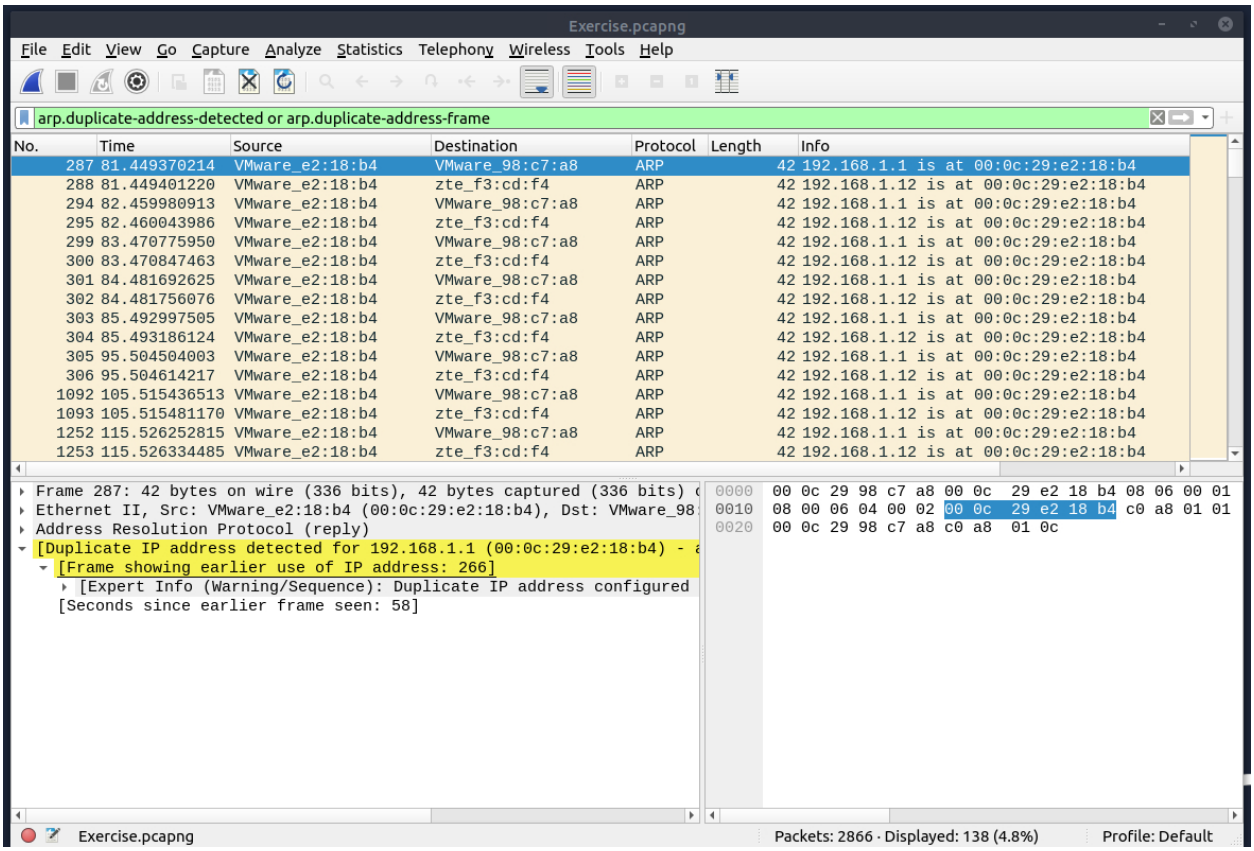
- As we see here, the traffic looks normal, but don’t let the adversary fool the eyes. Wireshark detects everything that is occurring within the network
- Here we will add the mac address column to further investigate.



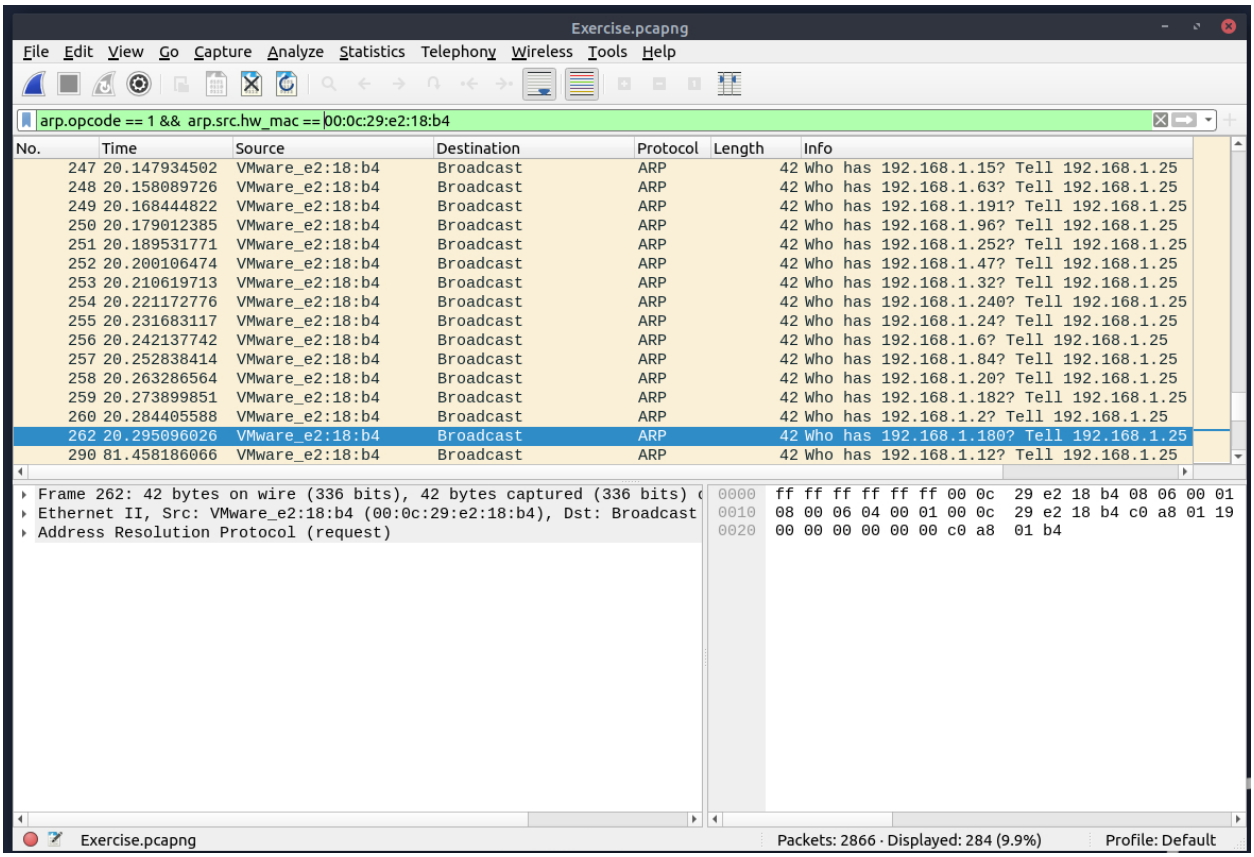
- Notice the same MAC address ending in “b4” is the destination of all the http packets.
- This assures us that there is a MitM attack occurring.
- The adversary is the host with the MAC address ending in “b4” and all the traffic linked to the IP address “192.168.1.12”.

Notes	Detection Notes	Findings
IP to MAC matches.	3 IP to MAC address matches.	• 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 ARP packets.	• MAC: 00:0c:29:e2:18:b4 = IP: 192.168.1.25
Router/gateway	Gateway address.	• MAC: 50:78:b3:f3:cd:f4 = IP: 192.1681.1
Victim	The attacker sniffed all traffic of the victim.	• 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.

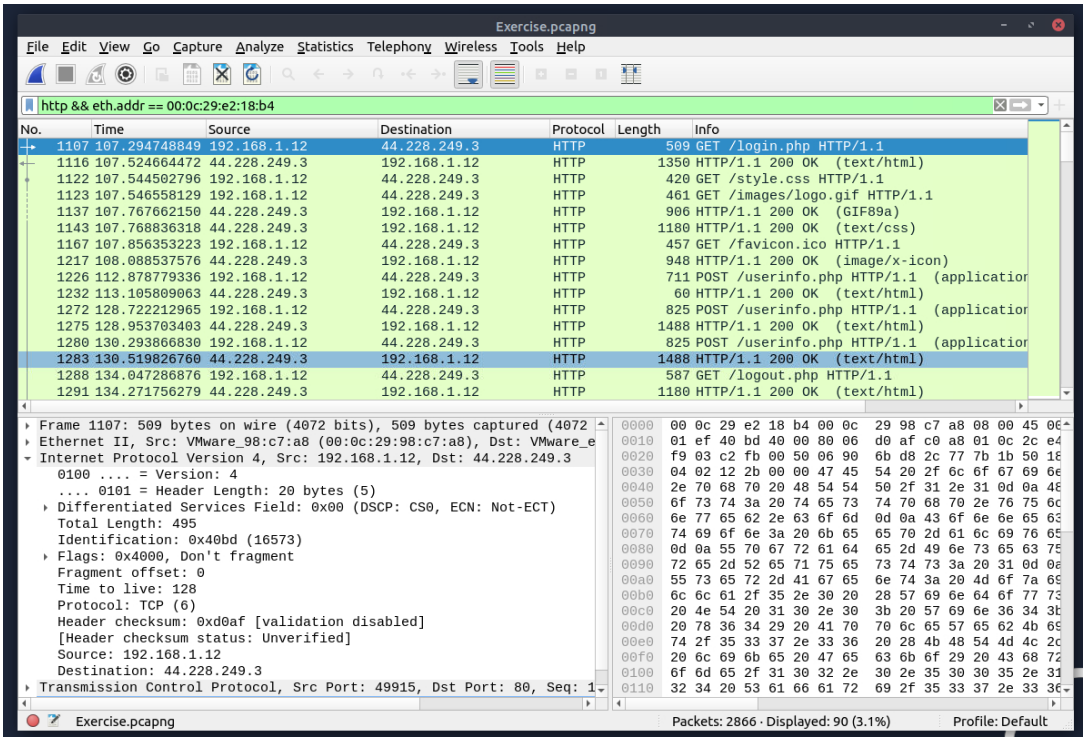


- Highlighted in yellow, there is a MAC address that is causing the duplication.
- From here we can now input a filter than can determine how many requests was made by that particular MAC address “00:0c:29:e2:18:b4”
- arp.opcode ==1 && arp.src.hw_mac == 00:0c:29:e2:18:b4 <--- this filter is telling Wireshark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



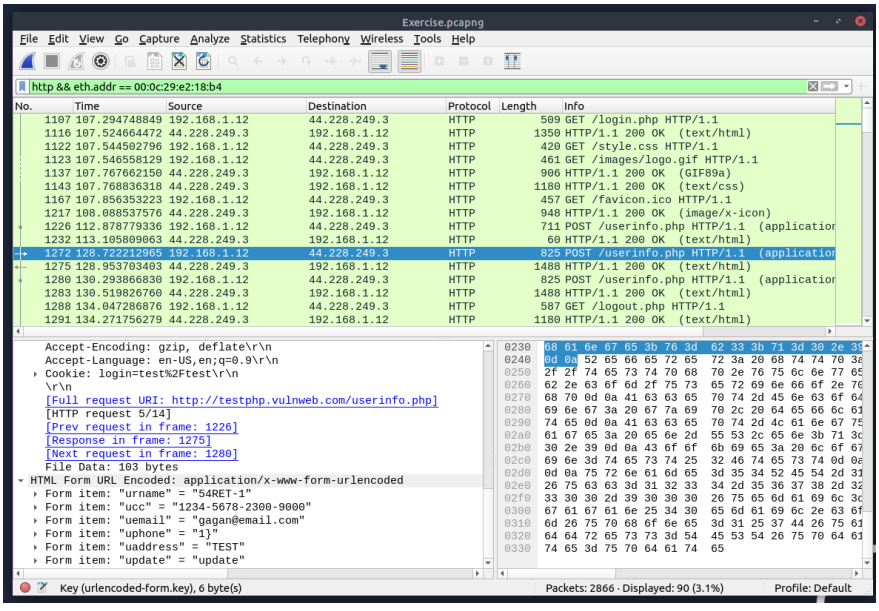
Answer: 284

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

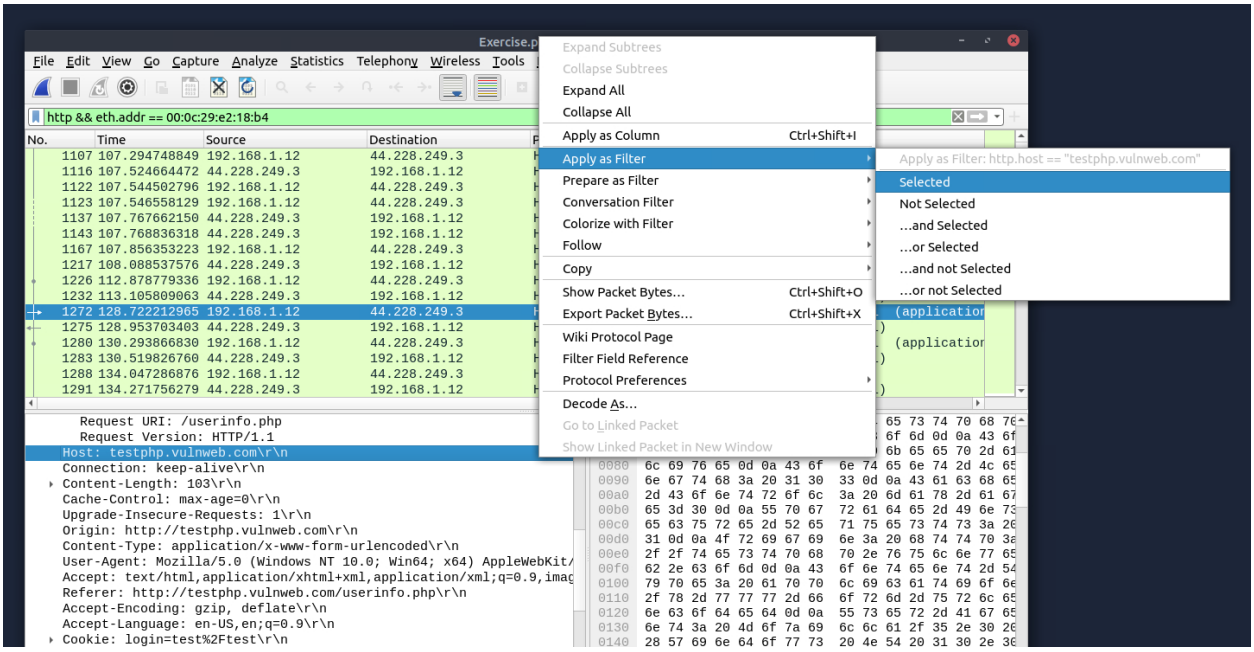
3. **What is the number of sniffed username & password entries?**
   - Notice in the screenshot above there are some POST info within the packet sniffer. Let's observe what we can find.

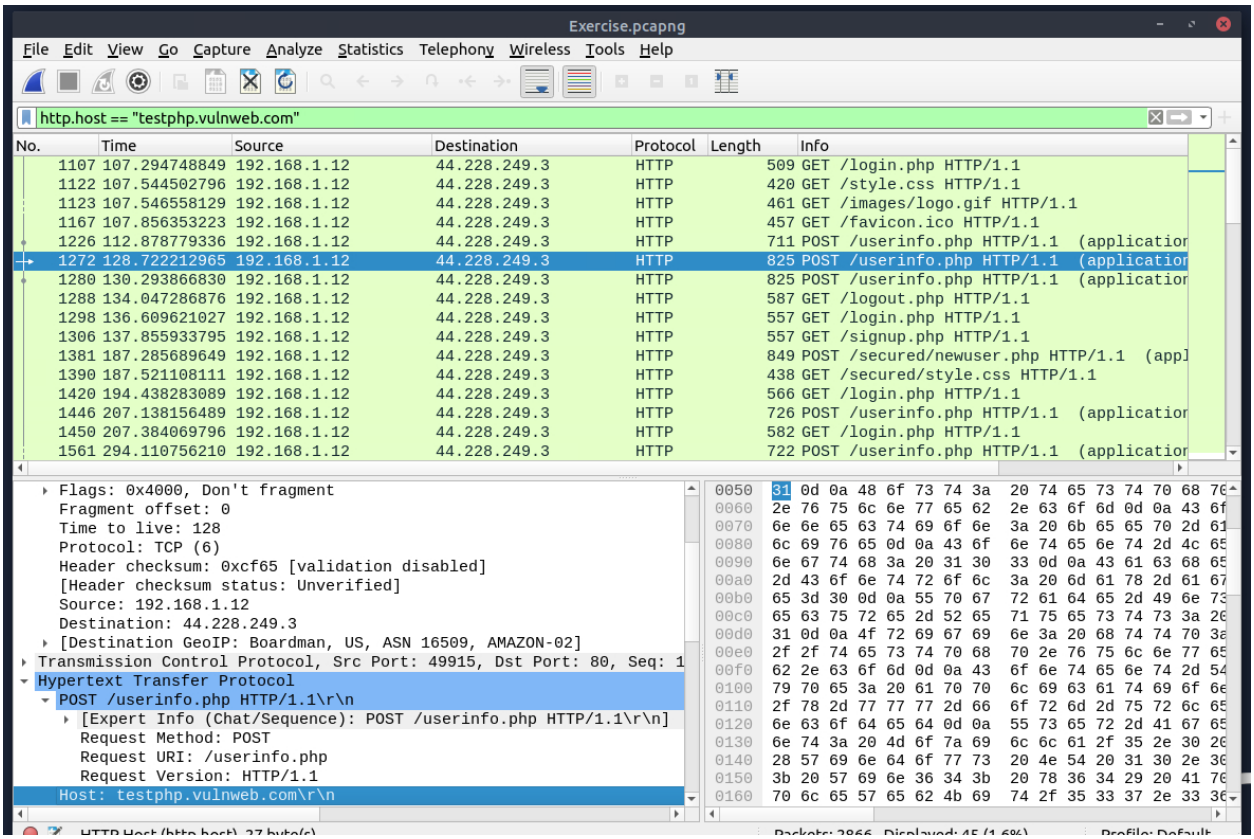


Answer: 6

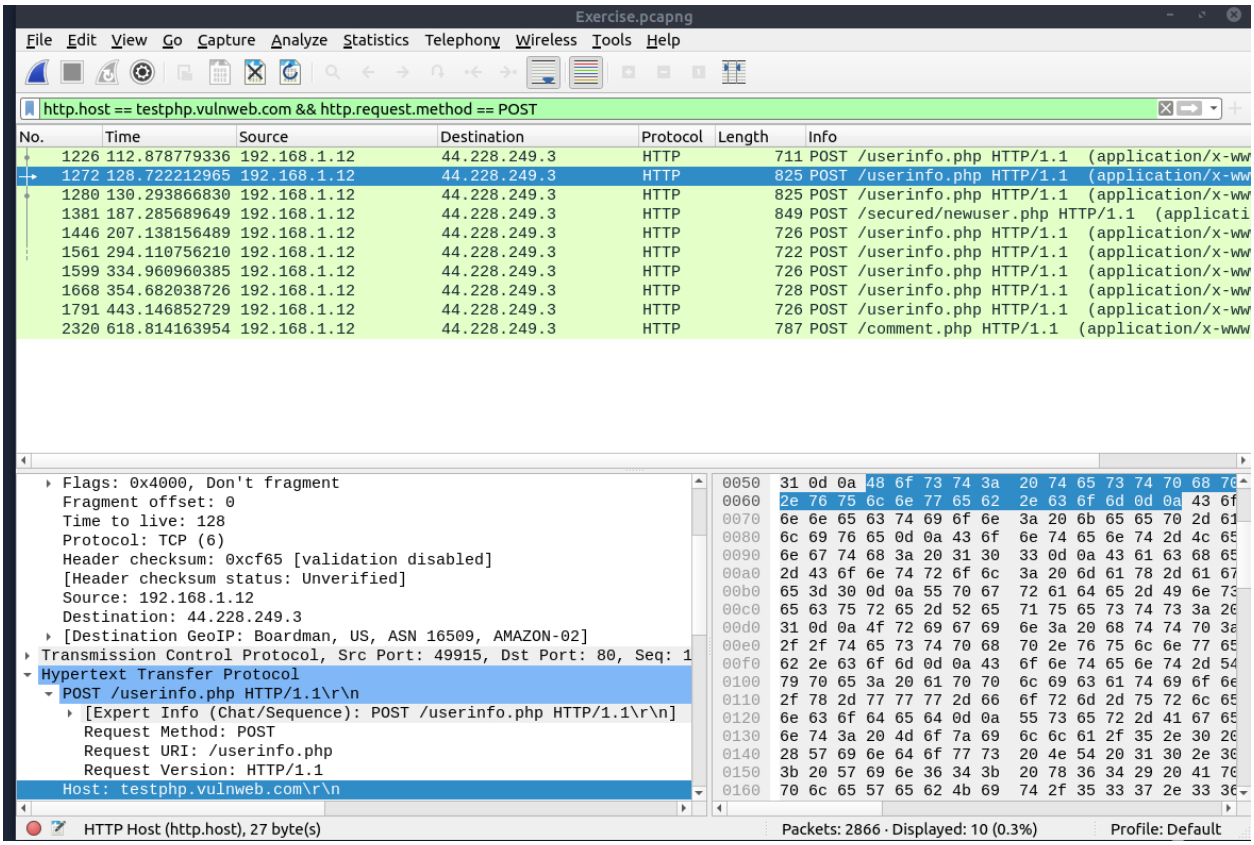
- Looking at the “userinfo.php ” POST response, I was able to identify a few username and passwords.

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.

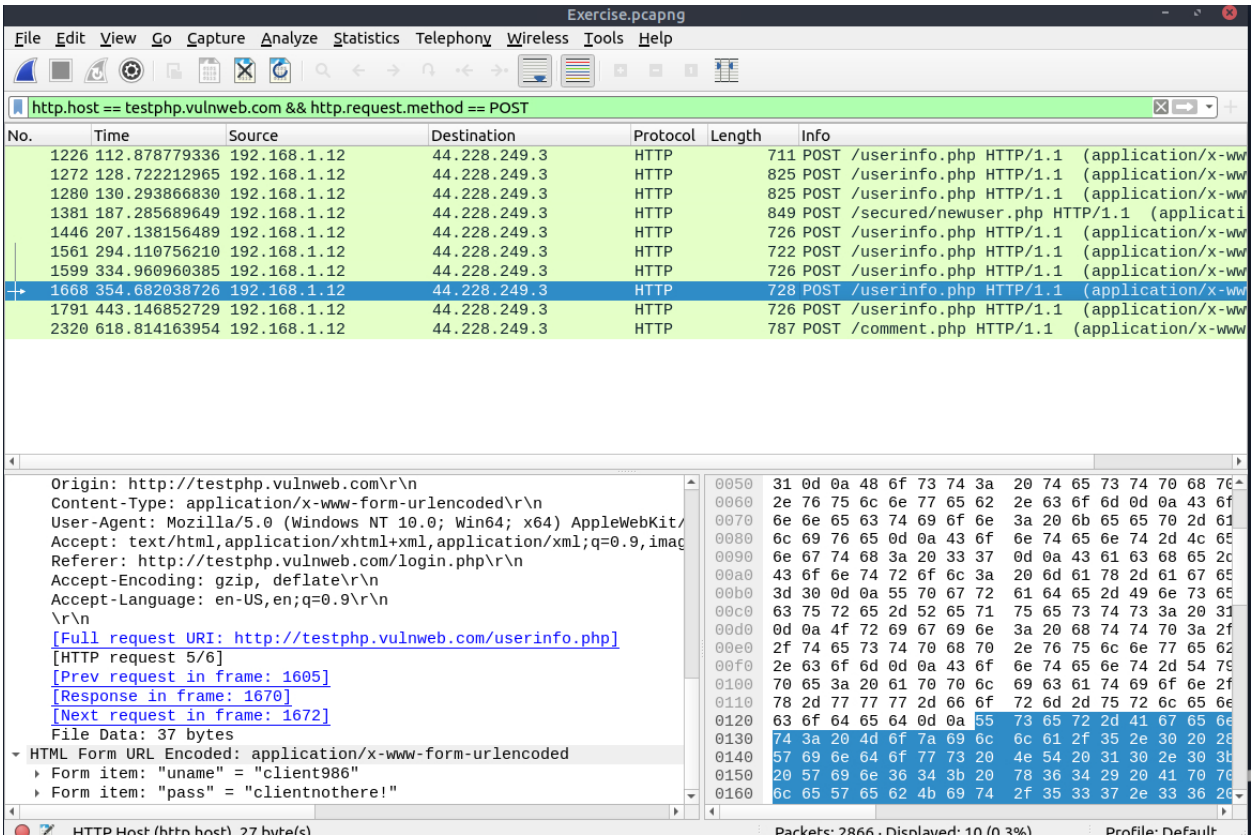




- Next, we want to generate all http “POST” requests.
- If we review the “HTTP” protocol. We can use the http.request.method == POST
- http.host == testphp.vulweb.com && http.request.method == POST <--- This filter is telling Wireshark to look for only POST request for the host of “testphp.vulnweb.com”



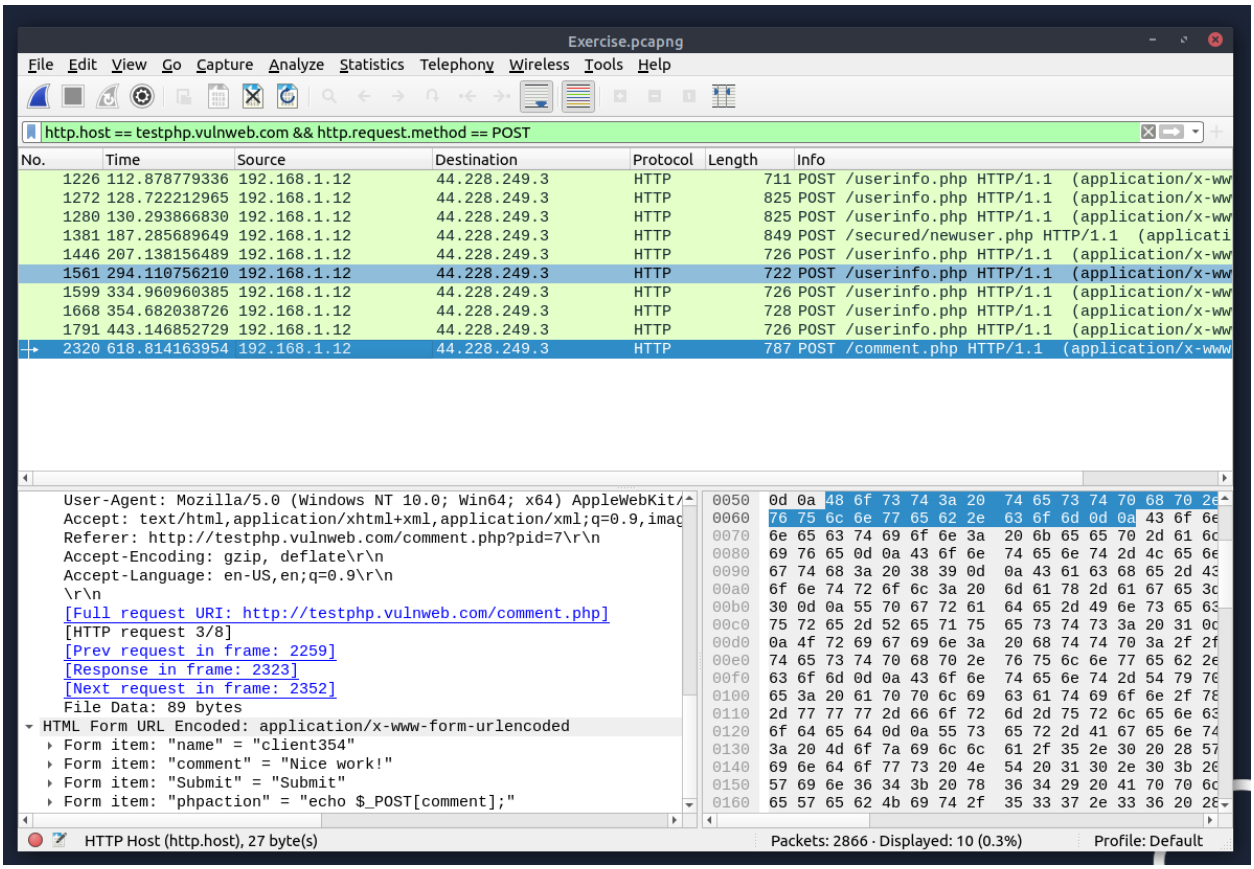
- We have 10 results.
- Lets comb through the results and see what we can find that associate with the user “Client986”



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!