

P4Pi

Unmatched MAC Address

The image shows a Wireshark packet capture window titled "Capturing from enx0c37965f8a23". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar with various icons for packet capture and analysis. Below the toolbar is a list of captured packets. The first packet is selected, and its details are shown in the bottom pane.

No.	Time	Source	Destination	Protocol	Length	Info
192	6.300502137	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
193	6.355748561	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
194	6.356299136	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
195	6.411773039	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
196	6.412332168	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
197	6.464403456	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
198	6.464936780	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
199	6.531940309	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22
200	6.532450762	192.168.10.1	192.168.10.2	UDP	64	50000 → 1024 Len=22

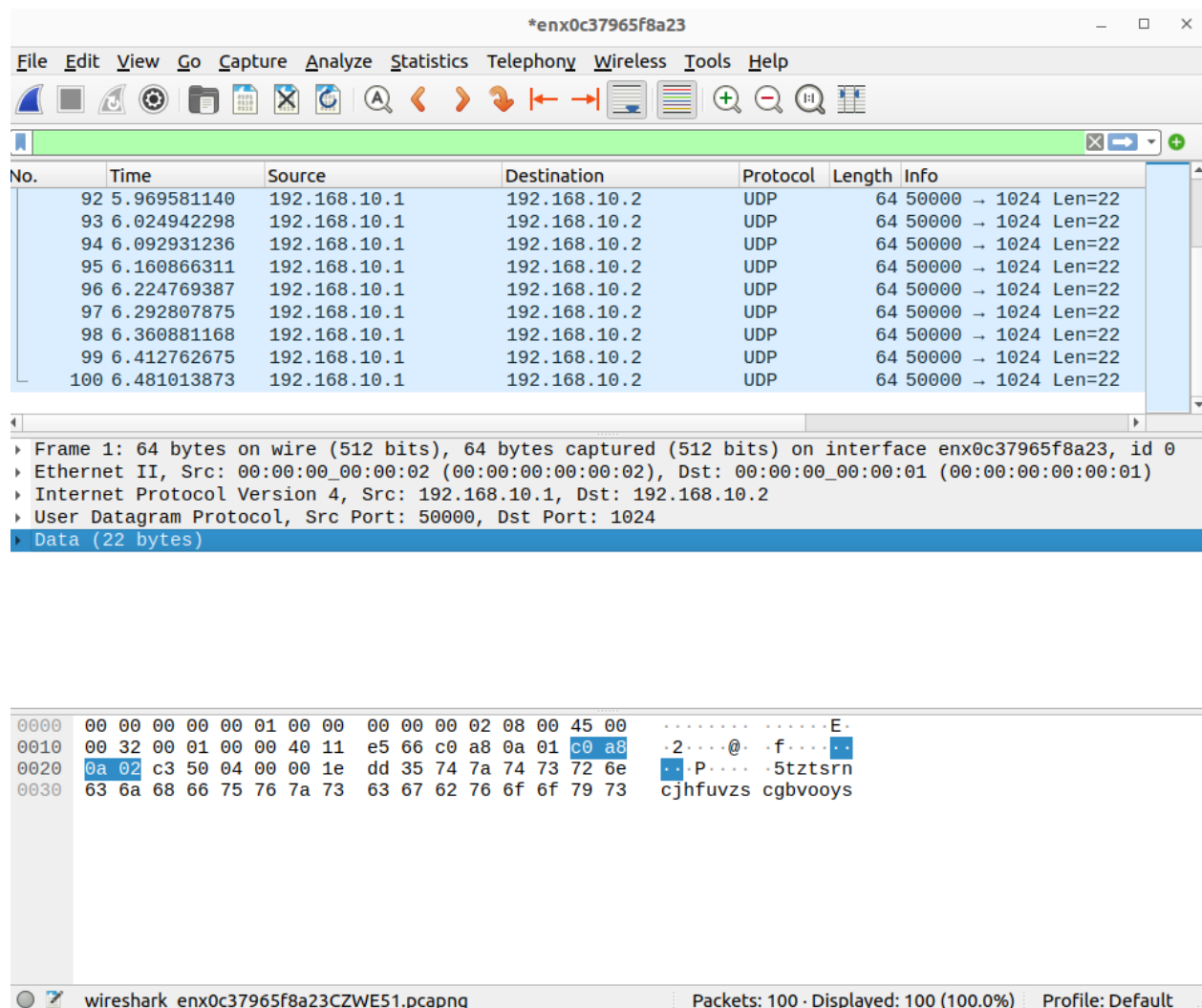
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f8a23, id 0
Ethernet II, Src: 00:00:00_00:00:02 (00:00:00:00:00:02), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.10.2
User Datagram Protocol, Src Port: 50000, Dst Port: 1024
Data (22 bytes)

```
0000  00 00 00 00 00 01 00 00 00 00 00 02 08 00 45 00  .....E.
0010  00 32 00 01 00 00 40 11 e5 66 c0 a8 0a 01 c0 a8  .2...@. .f.....
0020  0a 02 c3 50 04 00 00 1e e0 61 66 7a 6c 76 6c 73  ...P... .afzlvls
0030  74 67 6a 75 77 78 71 62 74 65 66 62 75 64 6b 63  tgjuwxqb tefbudkc
```

Data (data), 22 byte(s) Packets: 200 · Displayed: 200 (100.0%) Profile: Default

No packets dropped because the MAC address doesn't match the new entry to the table. As such, when 100 packets were sent, wireshark detected 200, an action defined in the reflector when it switches the source and destination addresses.

Matched MAC Address



Wireshark packet capture showing a list of UDP packets. The packet list is filtered to show only packets from 192.168.10.1 to 192.168.10.2. The packet details pane shows the structure of a frame: Ethernet II, Internet Protocol Version 4, and User Datagram Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
92	5.969581140	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
93	6.024942298	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
94	6.092931236	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
95	6.160866311	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
96	6.224769387	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
97	6.292807875	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
98	6.360881168	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
99	6.412762675	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22
100	6.481013873	192.168.10.1	192.168.10.2	UDP	64	50000 -> 1024 Len=22

Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface enx0c37965f8a23, id 0
Ethernet II, Src: 00:00:00_00:00:02 (00:00:00:00:00:02), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.10.2
User Datagram Protocol, Src Port: 50000, Dst Port: 1024
Data (22 bytes)

```
0000  00 00 00 00 00 01 00 00 00 00 02 08 00 45 00  .....E.  
0010  00 32 00 01 00 00 40 11 e5 66 c0 a8 0a 01 c0 a8  .2...@.f...  
0020  0a 02 c3 50 04 00 00 1e dd 35 74 7a 74 73 72 6e  .P...5tztsrn  
0030  63 6a 68 66 75 76 7a 73 63 67 62 76 6f 6f 79 73  cjhfvuzs cgbvooyo
```

Here, we have added a table entry with the exact MAC address stated in send.py, and this caused all the packets sent to be dropped. Therefore, there was no reflection, resulting in wireshark detecting only 100 packets, the same number that was sent.

Yes, you can add table entries to make the program drop packets. The above example demonstrates this. To accomplish this, we need to input a table entry that matches the source MAC address as seen in send.py (00:00:00:00:00:02).

Experimentation with CWM

CWM already includes the somewhat long clunky instructions as macros. Some minor name changes make the program compatible with our working folder, and running “cwm compile” and “cwm run” does the same thing as the previous part. This is a better way to work with these commands.