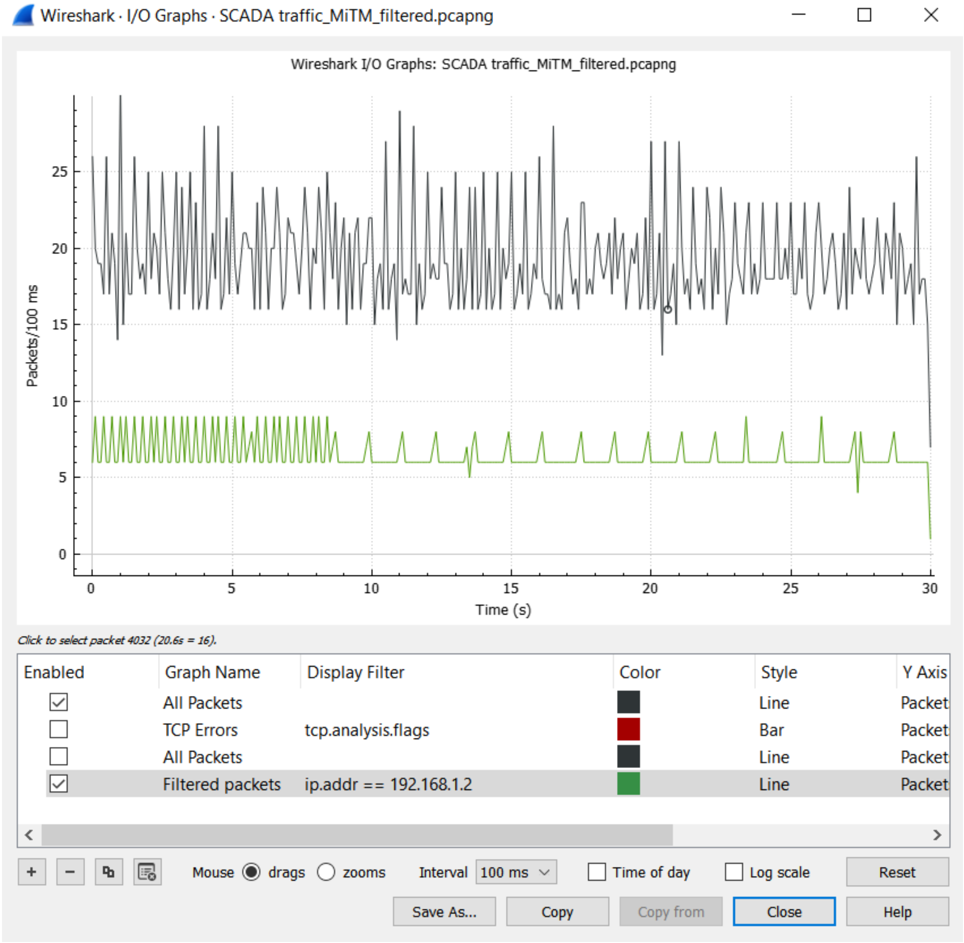
Analysis Example

Objective:
Identifying a type of attack by analyzing a packet capture through Wireshark.

File Name: Sample1

Analysis:
On the first view, this looks like any other packet capture. Messing around with different filters, nothing is obvious.

Let’s start with **Input/Output analysis**. This enables an at-a-glance view of total packets transferred in the time frame. Reducing the time frame interval to 100ms and checking each IP address as a filter, you find some anomaly in IP address 192.168.1.2.

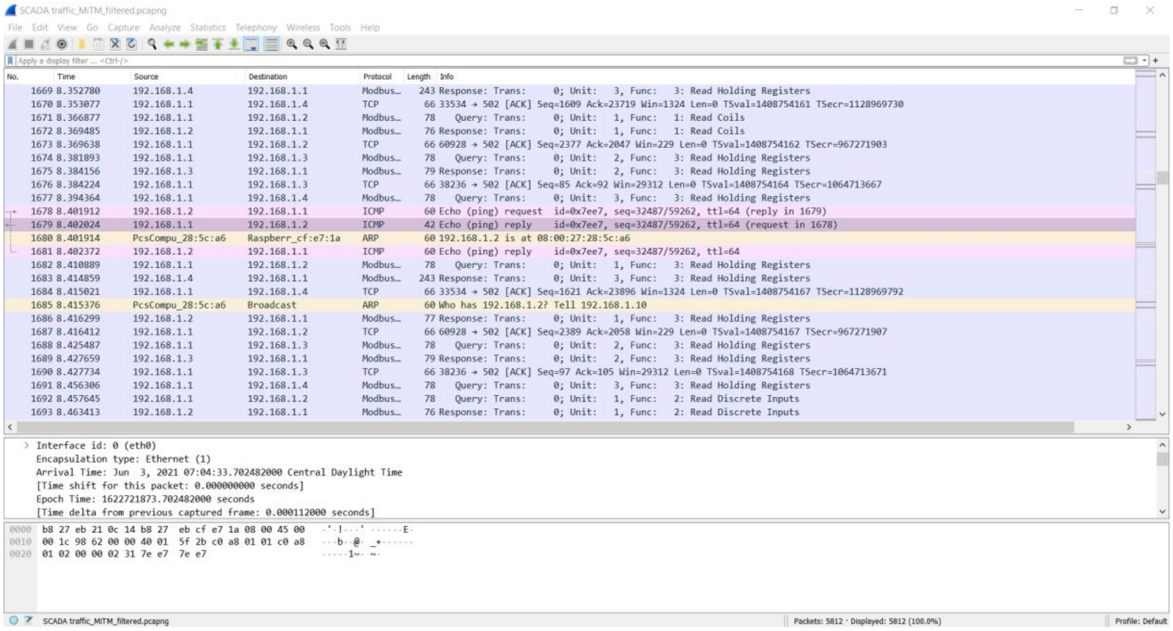

Wireshark · I/O Graphs · SCADA traffic_MiTM_filtered.pcapng
Wireshark I/O Graphs: SCADA traffic_MiTM_filtered.pcapng
Packets/100 ms
0
5
10
15
20
25
Time (s)
0
5
10
15
20
25
30
+
−
Reset
Save As...
Copy
Copy from
Close
Help

Click to select packet 4032 (20.6s = 16).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis
☑	All Packets		⬛	Line	Packet
☐	TCP Errors	tcp.analysis.flags	🟥	Bar	Packet
☐	All Packets		⬛	Line	Packet
☑	Filtered packets	ip.addr == 192.168.1.2	🟩	Line	Packet

Mouse ● drags ○ zooms Interval 100 ms ☐ Time of day ☐ Log scale

It shows a sharp decrease in the packets received by 192.168.1.2 after the initial 8s. Checking around that 8 second mark, you can see a few ICMP packets as well as a couple of ARP packets being transferred.



SCADA traffic_MiTM_filtered.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time      Source             Destination        Protocol  Length Info
 1669 8.352780  192.168.1.4        192.168.1.1        Modbus…   243 Response: Trans:     0; Unit:   3, Func:   3: Read Holding Registers
 1670 8.353077  192.168.1.1        192.168.1.4        TCP        66 33534 → 502 [ACK] Seq=1609 Ack=23719 Win=1324 Len=0 TSval=1408754161 TSecr=1128969730
 1671 8.366877  192.168.1.1        192.168.1.2        Modbus…    78    Query: Trans:     0; Unit:   1, Func:   1: Read Coils
 1672 8.369485  192.168.1.2        192.168.1.1        Modbus…    76 Response: Trans:     0; Unit:   1, Func:   1: Read Coils
 1673 8.369638  192.168.1.1        192.168.1.2        TCP        66 60928 → 502 [ACK] Seq=2377 Ack=2047 Win=229 Len=0 TSval=1408754162 TSecr=967271903
 1674 8.381893  192.168.1.1        192.168.1.3        Modbus…    78    Query: Trans:     0; Unit:   2, Func:   3: Read Holding Registers
 1675 8.384156  192.168.1.3        192.168.1.1        Modbus…    79 Response: Trans:     0; Unit:   2, Func:   3: Read Holding Registers
 1676 8.384224  192.168.1.1        192.168.1.3        TCP        66 38236 → 502 [ACK] Seq=85 Ack=92 Win=29312 Len=0 TSval=1408754164 TSecr=1064713667
 1677 8.394364  192.168.1.1        192.168.1.4        Modbus…    78    Query: Trans:     0; Unit:   3, Func:   3: Read Holding Registers
 1678 8.401912  192.168.1.2        192.168.1.1        ICMP       60 Echo (ping) request  id=0x7ee7, seq=32487/59262, ttl=64 (reply in 1679)
 1679 8.402024  192.168.1.1        192.168.1.2        ICMP       42 Echo (ping) reply    id=0x7ee7, seq=32487/59262, ttl=64 (request in 1678)
 1680 8.401914  PcsCompu_28:5c:a6  Raspberr_cf:e7:1a  ARP        60 192.168.1.2 is at 08:00:27:28:5c:a6
 1681 8.402372  192.168.1.2        192.168.1.1        ICMP       60 Echo (ping) reply    id=0x7ee7, seq=32487/59262, ttl=64
 1682 8.410889  192.168.1.1        192.168.1.2        Modbus…    78    Query: Trans:     0; Unit:   1, Func:   3: Read Holding Registers
 1683 8.414859  192.168.1.4        192.168.1.1        Modbus…   243 Response: Trans:     0; Unit:   3, Func:   3: Read Holding Registers
 1684 8.415021  192.168.1.1        192.168.1.4        TCP        66 33534 → 502 [ACK] Seq=1621 Ack=23896 Win=1324 Len=0 TSval=1408754167 TSecr=1128969792
 1685 8.415376  PcsCompu_28:5c:a6  Broadcast          ARP        60 Who has 192.168.1.2? Tell 192.168.1.10
 1686 8.416299  192.168.1.2        192.168.1.1        Modbus…    77 Response: Trans:     0; Unit:   1, Func:   3: Read Holding Registers
 1687 8.416412  192.168.1.1        192.168.1.2        TCP        66 60928 → 502 [ACK] Seq=2389 Ack=2058 Win=229 Len=0 TSval=1408754167 TSecr=967271907
 1688 8.425487  192.168.1.1        192.168.1.3        Modbus…    78    Query: Trans:     0; Unit:   2, Func:   3: Read Holding Registers
 1689 8.427659  192.168.1.3        192.168.1.1        Modbus…    79 Response: Trans:     0; Unit:   2, Func:   3: Read Holding Registers
 1690 8.427734  192.168.1.1        192.168.1.3        TCP        66 38236 → 502 [ACK] Seq=97 Ack=105 Win=29312 Len=0 TSval=1408754168 TSecr=1064713671
 1691 8.456306  192.168.1.1        192.168.1.4        Modbus…    78    Query: Trans:     0; Unit:   3, Func:   3: Read Holding Registers
 1692 8.457645  192.168.1.1        192.168.1.2        Modbus…    78    Query: Trans:     0; Unit:   1, Func:   2: Read Discrete Inputs
 1693 8.463413  192.168.1.2        192.168.1.1        Modbus…    76 Response: Trans:     0; Unit:   1, Func:   2: Read Discrete Inputs
```

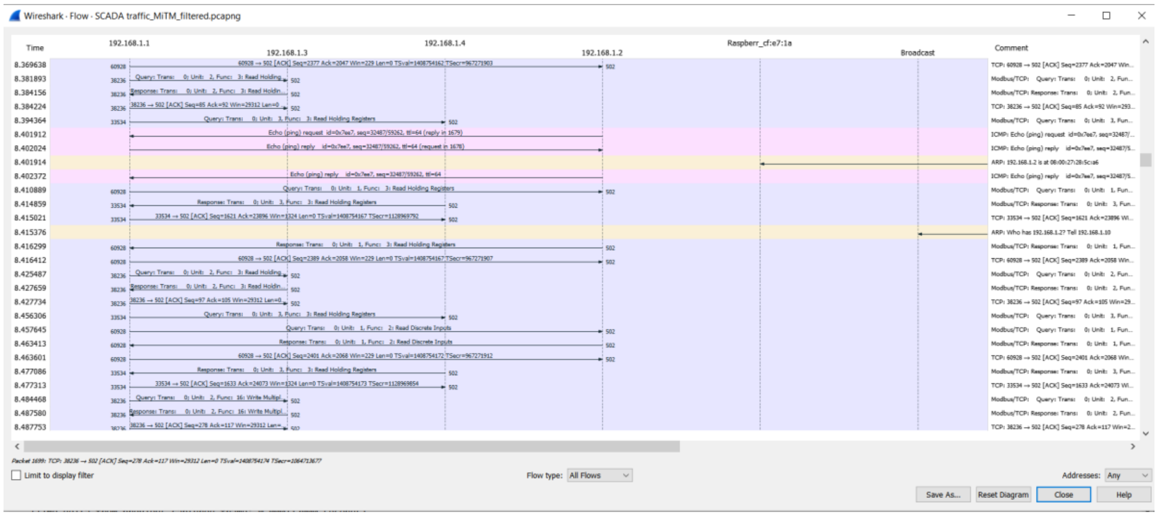
```
> Interface id: 0 (eth0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Jun  3, 2021 07:04:33.702482000 Central Daylight Time
  [Time shift for this packet: 0.000000000 seconds]
  Epoch Time: 1622721873.702482000 seconds
  [Time delta from previous captured frame: 0.000112000 seconds]
```

```
0000  b8 27 eb 21 0c 14 b8 27  eb cf e7 1a 08 00 45 00   ·'·!···' ······E·
0010  00 1c 98 62 00 00 40 01  5f 2b c0 a8 01 01 c0 a8   ···b··@· _+······
0020  01 02 00 00 02 31 7e e7  7e e7                     ·····1~· ~·
```

SCADA traffic_MiTM_filtered.pcapng
Packets: 5812 · Displayed: 5812 (100.0%)
Profile: Default

This looks like a spoofed ICMP packets. ARP is a protocol used in a LAN to resolve the MAC address of the next or final destination IP.

Following the communication flow in Wireshark makes it even more evident that the ICMP packets were sent to spoof a ARP request.



```
Time      Comment
8.369638  TCP: 60928 → 502 [ACK] Seq=2377 Ack=2047 Win...
8.381893  Modbus/TCP:    Query: Trans:     0; Unit:   2, Fun...
8.384156  Modbus/TCP: Response: Trans:     0; Unit:   2, Fun...
8.384224  TCP: 38236 → 502 [ACK] Seq=85 Ack=92 Win=293...
8.394364  Modbus/TCP:    Query: Trans:     0; Unit:   3, Fun...
8.401912  ICMP: Echo (ping) request  id=0x7ee7, seq=32487/...
8.402024  ICMP: Echo (ping) reply    id=0x7ee7, seq=32487/5...
8.401914  ARP: 192.168.1.2 is at 08:00:27:28:5c:a6
8.402372  ICMP: Echo (ping) reply    id=0x7ee7, seq=32487/5...
8.410889  Modbus/TCP:    Query: Trans:     0; Unit:   1, Fun...
8.414859  Modbus/TCP: Response: Trans:     0; Unit:   3, Fun...
8.415021  TCP: 33534 → 502 [ACK] Seq=1621 Ack=23896 Wi...
8.415376  ARP: Who has 192.168.1.2? Tell 192.168.1.10
8.416299  Modbus/TCP: Response: Trans:     0; Unit:   1, Fun...
8.416412  TCP: 60928 → 502 [ACK] Seq=2389 Ack=2058 Win...
8.425487  Modbus/TCP:    Query: Trans:     0; Unit:   2, Fun...
8.427659  Modbus/TCP: Response: Trans:     0; Unit:   2, Fun...
8.427734  TCP: 38236 → 502 [ACK] Seq=97 Ack=105 Win=29...
8.456306  Modbus/TCP:    Query: Trans:     0; Unit:   3, Fun...
8.457645  Modbus/TCP:    Query: Trans:     0; Unit:   1, Fun...
8.463413  Modbus/TCP: Response: Trans:     0; Unit:   1, Fun...
8.463601  TCP: 60928 → 502 [ACK] Seq=2401 Ack=2068 Win...
8.477086  Modbus/TCP: Response: Trans:     0; Unit:   3, Fun...
8.477313  TCP: 33534 → 502 [ACK] Seq=1633 Ack=24073 Wi...
8.484468  Modbus/TCP:    Query: Trans:     0; Unit:   2, Fun...
8.487580  Modbus/TCP: Response: Trans:     0; Unit:   2, Fun...
8.487753  TCP: 38236 → 502 [ACK] Seq=278 Ack=117 Win=2...
```

Packet 1699: TCP: 38236 → 502 [ACK] Seq=278 Ack=117 Win=29312 Len=0 TSval=1408754174 TSecr=1064713677

A Man In The Middle attack works by spoofing a MAC address within a LAN in response to a victim's ARP request. Immediately after the ICMP packets it sends a fake ARP replay (As seen in the above image). If the MAC of the intended machine is successfully accepted and spoofed with the attacker's machine, then the victim will send traffic to the spoofed MAC address instead of the destination MAC address.

ARP spoofing is used as a tool to alter victim's routing, helping in gaining MITM position.

Conclusion

Based on these reasoning/evidence, I think this is an example of a Man In The Middle attack which uses ICMP packets for ARP spoofing to achieve its goal.