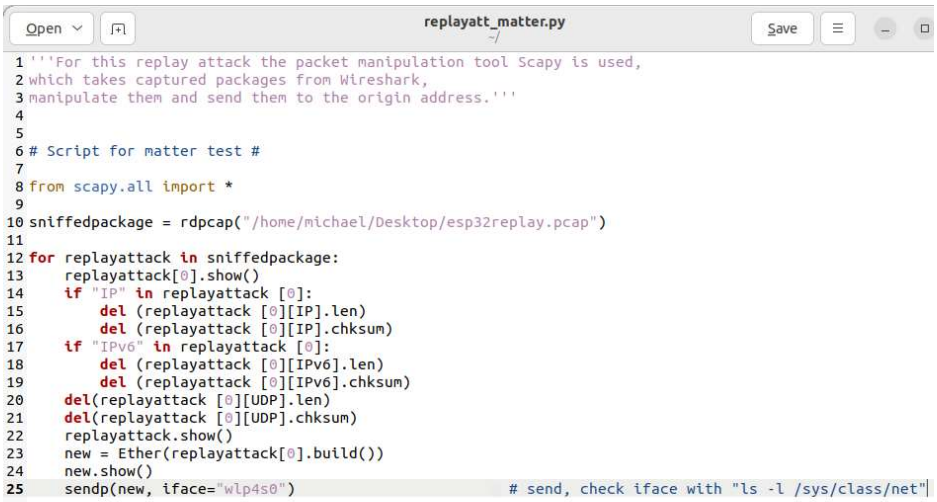
Replay attack

A replay attack is a cryptanalytic form of attack on the authenticity of data in a network. Here, an attacker sends previously recorded data packets to impersonate a legitimate identity and execute a command. Thus, the replay attack falls into the spectrum of man-in-the-middle attacks. Since the successful execution of a replay attack can result in unauthorised commands being resent as often as required, this type of attack poses a serious risk, as it can have a significant impact on the function of devices in a network.

In order to be able to carry out a replay attack, validly sent packages are required in order to manipulate and resend them. To capture these packets, the Wireshark program was used, which can record the outgoing and incoming packets from devices (Wireshark, N.D.). The packets can be captured remotely, but this requires appropriate hardware in the form of a Wi-Fi dongle that is able to switch to monitor mode, as well as an adaptation of the software, which was unsuccessful in the project (Kismet, N.D.). Therefore, the packets were captured on the computer from which the commands were sent to the Matter device. In order for a computer to act as a controller/hub in a Matter network, the chip-tool tool was used, which can be used for testing and troubleshooting Matter devices (Kajor et al., 2023). While this procedure cannot be used to completely reconstruct a replay attack, since the packets necessary for this attack were not captured remotely, executing the attack in this way can still be used to investigate whether Matter devices are fundamentally vulnerable to replay attacks. A Python script was created to manipulate and resend the packages, which works with the scapy tool. Scapy is a tool that is able to read, manipulate and resend packets of various standards (Scapy, N.D.) . The following image shows the created scapy script.



```python
'''For this replay attack the packet manipulation tool Scapy is used,
which takes captured packages from Wireshark,
manipulate them and send them to the origin address.'''


# Script for matter test #

from scapy.all import *

sniffedpackage = rdpcap("/home/michael/Desktop/esp32replay.pcap")

for replayattack in sniffedpackage:
    replayattack[0].show()
    if "IP" in replayattack [0]:
        del (replayattack [0][IP].len)
        del (replayattack [0][IP].chksum)
    if "IPv6" in replayattack [0]:
        del (replayattack [0][IPv6].len)
        del (replayattack [0][IPv6].chksum)
    del(replayattack [0][UDP].len)
    del(replayattack [0][UDP].chksum)
    replayattack.show()
    new = Ether(replayattack[0].build())
    new.show()
    sendp(new, iface="wlp4s0")                          # send, check iface with "ls -l /sys/class/net"
```

Figure 22: Matter replay attack python script

The following refers to the video ‘Replay_attack’.

It can initially be seen two terminals on the left side. The Matter device is monitored in the upper terminal. To do this, the command 'idf.py –p /dev/ttyUSB0 monitor' is used at the beginning. After entering the command, the Matter device restarts and a message is displayed in the lower terminal in which the chip-tool is used that a new Matter device has been found.

Now the recording of Wi-Fi packets is started with Wireshark, which is located on the right so that the sent packets can be recorded. The command 'pairing ble-wifi [Node_ID] [SSID] [Password] [setup_passcode] [discriminator]' is now entered in the chip-tool terminal, which initialises the commissioning process. It can now be seen that the monitor, as well as the chip-tool terminal, gives feedback that commissioning is taking place. After the first packets are sent over the Bluetooth standard and the Wi-Fi information is exchanged, the communication switches to the Wi-Fi standard so that Wireshark is able to capture packets. It can be seen that different protocols are used. These include ICMPv6, WireGuard and UDP. However, only the WireGuard and UDP packages are necessary for the test, as these play a role in establishing the connection and transmitting commands.

After the commissioning was successful, the command is now sent via the chip-tool to the Matter device that it should be switched on and then that the lamp should be regulated to 100%. After all packets have been intercepted by Wireshark, they are selected and stored separately. The existing file name 'exp32replay.pcap' is chosen for this, as this is the file name that was used in the scapy python script.

The python script 'replayatt_matter.py' shown corresponds to the one shown in the image above. In line 8 the scapy tool is imported. Line 10 selects the .pcap file containing the captured packets. In lines 12 to 22, the IP and UDP lengths and checksums are deleted and recreated. Finally, the packages are sent in line 25.

The command 'python3 replayatt_matter.py' is now entered and executed in the lower terminal, which activates the python script discussed. The lower terminal shows that the script is running. In the upper terminal it can be seen that packets have been received from the Matter device. However, the lamp did not turn on.

Furthermore, the error message 'OnMessageReceived failed, err =70' is given. Error 70 refers to the message counter, which is responsible for the integrity of communication. It can therefore be stated that the replay attack could be carried out, but was not successful.

References

Kajor, M., Siu, I., Turon, M., Litvin, A., Zbarsky, B., Lunde, G. S., & Smith, M. (2023) Working with the CHIP Tool. Available from: https://github.com/project-chip/connectedhomeip/blob/master/docs/guides/chip_tool_guide.md [Accessed 13 July 2023].

Kismet (N.D.) Passive Capture. Available from: https://www.kismetwireless.net/docs/readme/intro/passive_capture/ [Accessed 24 July 2023].

Scapy (N.D.) What is Scapy? Available from: https://scapy.net/ [Accessed 13 July 2023].

Wireshark Foundation (N.D) The world's most popular network protocol analyzer. Available from: https://www.wireshark.org/ [Accessed 31 July 2023].