Studienprogramm Sachbearbeiter:in Digitale Forensik
Praktikum Betriebssysteme
Dozenten: Tim Wetterau
wetterau@hs-mittweida.de
Stand: 26.06.2023

HOCHSCHULE MITTWEIDA
University of Applied Sciences

Betriebssysteme

Praktikum 2 - SSH und Security

In diesem Praktikum lernen Sie, wie Sie eine gesicherte Verbindung mit einem Server per SSH herstellen können. Außerdem führen wir dazu einen kleinen Netzwerkscan mit nmap durch. Ebenso wird die Speicherung von Passwörtern unter Linux beleuchtet und wie diese ggf. gebrochen werden können. Dazu schauen wir uns das Tool john an und wollen dieses praktisch anwenden.

Inhalte des Praktikums:

- Verbindungen mit SSH
- Netzwerkscan mit nmap
- Passwörter unter Linux
- Passwortcracking mit john

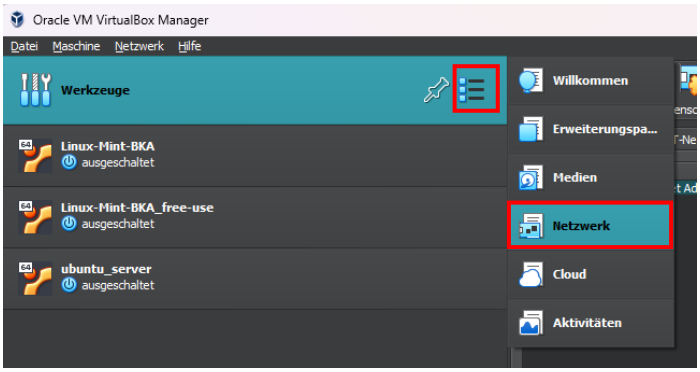
Vorbereitung

Zur Vorbereitung für dieses Praktikum haben wir im Praktikum 1 bereits openssh-client und nmap installiert. Zur Durchführung des Praktikums brauchen Sie außerdem eine weitere virtuelle Maschine, welche wir Ihnen unter Moodle im Kurs als Link auf Laufwerk R: zur Verfügung gestellt haben.

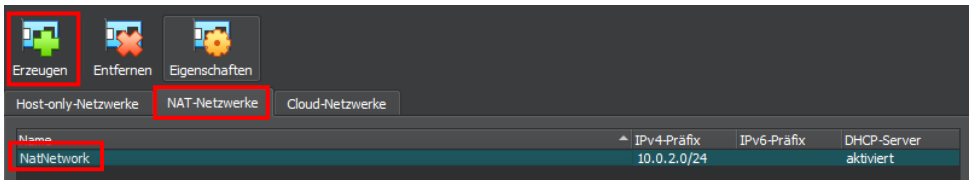
Zur Anmeldung am Ubuntu Server sind Ihnen folgende Login credentials gegeben:

- **Nutzername:** gustav
- **Passwort:** test-server
- **Servername:** ubuntu-srv-vb

Damit beide VMs problemlos miteinander und mit dem externen Netzwerk kommunizieren können, müssen wir erst ein paar Einstellungen verändern. Wir brauchen ein neues NAT-Netzwerk. Dazu gehen wir im Hauptscreen von VirtualBox auf „Werkzeuge“ und wählen „Netzwerk“

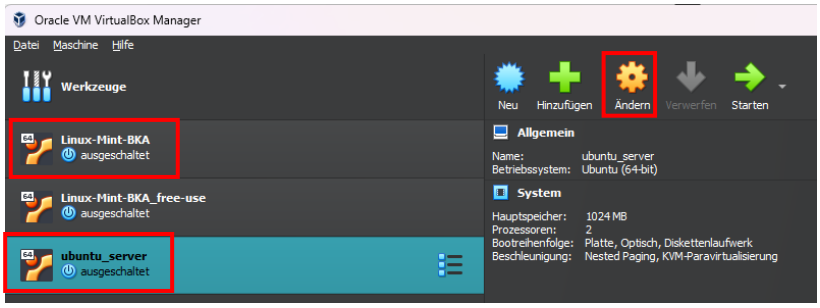


Anschließend wählen wir den Reiter „NAT-Netzwerke“ und wählen oben links erzeugen. Anschließend bekommen wir ein neues NAT-Netzwerk erzeugt.

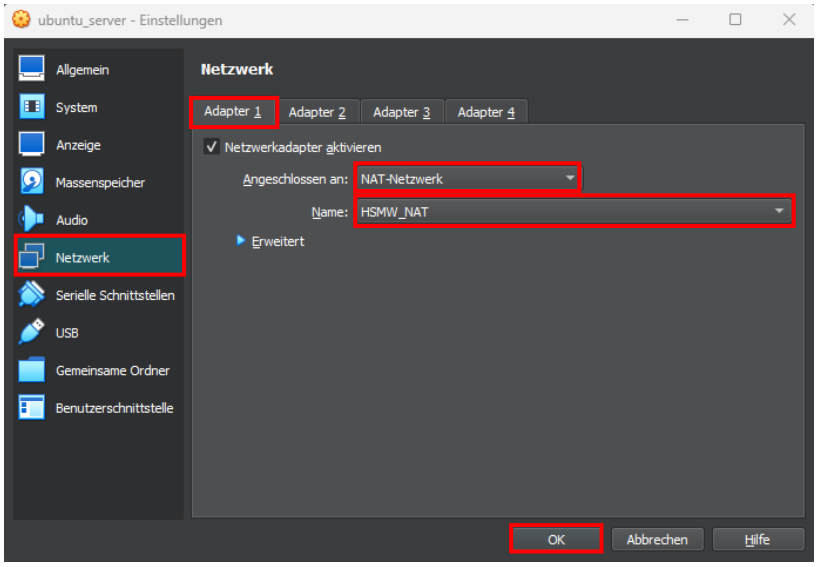


Im unteren Teil können wir dem Kind noch einen Namen und eine IP-Adresse geben. Beispielsweise:

Abschließend dazu klicken wir unten rechts noch auf „Sichern“. Anschließend weißen wir den beiden virtuellen Maschinen noch das gerade erzeugte NAT-Netzwerk zu. Dazu gehen wir auf wieder zurück auf unsere VM und auf „Ändern“



Dann weisen wir unter „Netzwerk“ im Adapter 1 das erzeugte NAT-Netzwerk zu und bestätigen unsere Einstellungen mit „OK“.

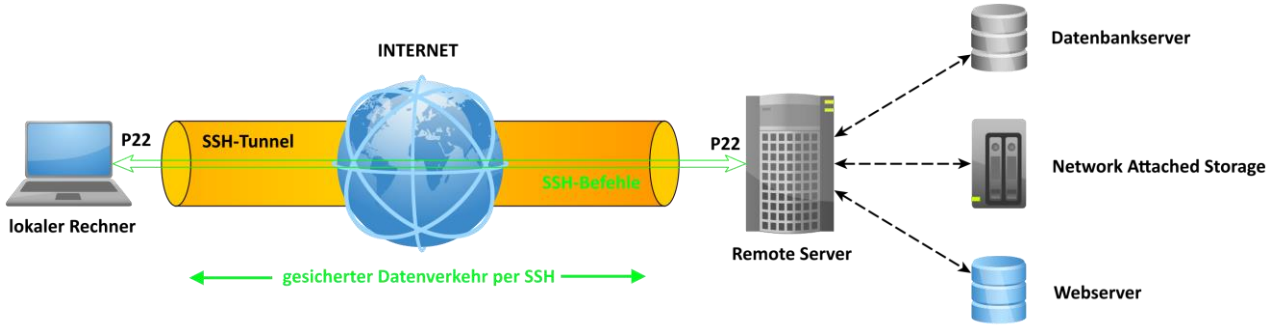


Jetzt haben wir alles vorbereitet, um in dieses Praktikum zu starten. Nehmen Sie dazu, wenn Sie es noch nicht getan haben, beide VMs in Betrieb mit einem Klick auf „Starten“. Den Ubuntu-Server können Sie dann einfach zu Seite legen, da wir mit diesem nur über die Linux Mint VM interagieren.

Verbindungen mit SSH

Server werden grundsätzlich remote gesteuert bzw. konfiguriert. Dafür bietet sich ein Protokoll namens SSH – Secure Shell an, um eine sichere Verbindung mit einem entfernten Rechner herzustellen. Der Standardport ist hier an der Stelle 22. Es wird meistens empfohlen diesen aus sicherheitstechnischen Gründen zu ändern. Im Praktikum soll uns der Port 22 reichen. SSH erzeugt einen Tunnel zwischen den beiden Kommunikationspartnern, der auf der

Verschlüsselung des Datenverkehrs basiert mittels der Authentifizierung von SSH-Keys, also Schlüsselpaaren. Das folgende Bild zeigt ein Beispielszenario, wobei der SSH-Verkehr durch das ungesicherte Internet getunnelt wird. Dabei können z.B. Befehle an den Datenbankserver gesendet werden oder Dateien vom NAS sicher übertragen werden.



SSH kann es uns ermöglichen einen Login per Login-Credentials durchzuführen. Es kann auch so konfiguriert werden, dass es nur Verbindungen zulässt, mit welchen ein Schlüsselpaar erzeugt wurde. Damit können nur Rechner auf die Maschine zugreifen, wenn der Rechner im Besitz eines Schlüssels ist. Das kann bspw. mittels RSA - Rivest–Shamir–Adleman umgesetzt werden.

Wir wollen folgend eine SSH-Verbindung zu unserem Ubuntu-Server herstellen unter der Annahme, dass wir dessen IP-Adresse nicht kennen. Dafür machen wir uns im nächsten Kapitel das installierte Tool nmap zunutze.

Netzwerkscan mit nmap

nmap ist ein Tool, welches besonders im Bereich von Pentesting genutzt wird. Damit können einzelne Rechner bzw. ganze Netzwerke gescannt werden. Wir müssen für die Untersuchung des aktuellen Netzwerks einen kleinen Netzwerkscan durchführen, um den Ubuntu-Server im Netzwerk zu finden. Zuerst schauen wir uns die Netzwerkkonfiguration auf Mint an mit

```
$ ifconfig -a
```

Jetzt sehen wir die aktuelle Netzwerkkonfiguration. *Können Sie bestätigen, dass der Computer an das gerade konfigurierte NAT-Netzwerk angeschlossen ist? Woran können Sie dies erkennen?*

Um den Ubuntu-Server im Netzwerk zu finden, machen wir einen Netzwerkscan mittels nmap. Dazu nutzen wir auf unserer Linux Mint Maschine folgenden Befehl:

```
$ nmap <Netzwerk-IP>/<CIDR>
```

Die Netzwerkadresse erkennen Sie bei der Ausgabe von **`ifconfig`** der Netzwerkkonfiguration im „**`enp0s3`**"-Adapter unter **`inet`**. Dabei geben Sie aber nur die Stellen, die in der Netzwerkmaske erfasst sind an mit der Maske selbst dazu in CIDR-Notation. *Was können Sie der Ausgabe von nmap entnehmen? Wie viele IP-Adressen wurden gescannt? Welche Maschinen mit welchen offenen Ports wurden erkannt?*

Als Ausgabe müssten Sie, bei korrekten Einstellungen, einen Rechner mit einem offenen SSH-Port (22) angezeigt bekommen. Das ist allen Anschein nach unser Ubuntu-Server. *Notieren Sie sich dessen IP-Adresse*. Diese brauchen wir, um eine Verbindung zu dem Server herzustellen. *Stellen Sie mit der folgenden Befehlssyntax eine Verbindung zu diesem her:*

```
$ ssh gustav@<IP-Adresse>
```

Sie bekommen eine Benachrichtigung, dass der Rechner nicht authentifiziert werden kann, da zum ersten Mal zu diesem eine Verbindung hergestellt wird. Sie bekommen außerdem den SHA256-Fingerprint des Rechners angezeigt und werden gefragt, ob die dem Server mit diesem Fingerprint vertrauen wollen. Das bestätigen Sie mit der Eingabe von „yes".

Geben Sie anschließend bei Aufforderung das Passwort von gustav ein. *Woran erkennen Sie, dass Sie sich auf dem entfernten Server erfolgreich angemeldet haben? Mit welchen Befehlen können Sie das bestätigen?* Spätestens jetzt können Sie nur noch auf die Arbeit mit der Kommandozeile zurückgreifen, da der Server keine grafische Oberfläche anbietet.

Passwörter unter Linux

Wir wollen uns auf dem Server einmal umschauen, was Passwörter betrifft. Wichtige Dateien für die Nutzerinformationen kennen wir bereits. Das sind die **`/etc/passwd`** und die **`/etc/shadow`**. Aus diesen Dateien lassen sich Informationen beziehen, welche uns das Knacken des Passwortes erleichtern. *Schauen Sie sich zuerst an, welche Nutzer am System eingerichtet sind.*

Standardmäßig werden Passwörter mit Salt auf aktuellen Distributionen mit den SHA512 gehasht und dann gespeichert. *Schauen wir uns dazu die Shadow-Dateien an.:*

```
$ sudo cat /etc/shadow
```

Wir sehen beispielhaft folgenden Aufbau in der Shadow-Datei:

```
usbmux:*:19451:0:99999:7:::
gustav:$6$FZLfINek9HdNs2Qa$FZKwkk5h6tljjt30lrMyXKr73U4kd6AbESEHZ6jfCqTDFEssKOUo1d.VLAsFjfQdFu8EYQu9Yn/B2xmij1glj1:19451:0:99999:7:::
```

Die Spalten in der Datei werden jeweils durch das Trennzeichen „**:**" separiert. In der ersten Spalte sehen wir den Nutzernamen. Danach folgt eine Spalte mit den eigentlichen Informationen über das Passwort des jeweiligen Nutzers. Bei „**usbmux**" sehen wir ein „*****", welches für kein Passwort steht. Das ist bei Verwaltungsnutzeraccounts standardmäßig so. Bei „**gustav**" hingegen sehen wir einen Eintrag, welcher wiederum drei Felder besitzt, welche durch ein „**$**" getrennt sind. Der erste Werte 6 steht für den Algorithmus, der zum Hashen des Passwortes verwendet wurde, in diesem Fall SHA512. In der zweiten Spalte befindet sich der Salt. Passwörter unter Linux werden meist gesalzen abgelegt. Zum Thema Salt belesen Sie sich bitte im Selbststudium, wozu ihnen folgender Link von Heise online gegeben ist. Grundlegend kann das Salzen von Passwörtern wie folgt schematisch beschrieben werden:

$$Hash_{pwd} = H(pwd_s) = H(concat("pwd" + "salt")) \mid H \sim Hashalgorithmus, pwd_s \sim gesalzenes\ Passwort$$

Die letzte durch „**$**" getrennte Spalte ist der eigentliche Passworthash, den es knacken gilt. Die nächste durch „**:**" getrennte Spalte beinhaltet die Tage seit dem 01.01.1970 an dem das Passwort zuletzt geändert wurde. Danach folgen die Mindestanzahl und die Maximalanzahl, wann das Passwort wieder geändert werden darf und muss. Die nächste Spalte ist die Anzahl an Tagen der Warnung, bevor das Passwort abläuft. Die letzten Spalten sind reservierte Felder für mögliche Erweiterungen. So viel zur Theorie.

Sicheres Kopieren entfernter Dateien per scp

Schauen wir uns die Praxis an und knacken ein paar Passwörter. Dazu haben wir uns im vorherigen Praktikum das Tool **`john`** heruntergeladen. Schauen wir uns dazu aber zuerst an, welche Nutzer auf dem entfernten Ubuntu-Server eingerichtet sind. *Schauen Sie sich dazu die* **`/etc/passwd`** *und die Shadow-Datei auf dem Ubuntu-Server an! Welche Nutzer mit Passworthash können Sie feststellen?*

Um effektiv mit john zu arbeiten, ist es sinnvoll die beiden Dateien zusammenzuführen. Diesen Vorgang nennt man „unshadow“. Diese neue Datei kann **john** einfach als Input nehmen. Wichtig an der Stelle ist, dass wir das Knacken der Passwörter nicht auf dem entfernten System durchführen, sondern auf dem lokalen Mint-System. Dazu müssen wir zuerst die Dateien auf unser System kopieren, wozu wir **scp** (secure copy) nutzen:

```
$ scp [option] <Quellverzeichnis> <Zielverzeichnis>
```

Dabei kann, je nach dem, von wo nach wohin eine Datei kopiert werden soll, auch eine IP-Adresse mit Zugangsdaten als Ziel oder Quelle angegeben werden nach dem Schema:

```
$ <Nutzername>@<IP-Adresse>:<dir/file.ext>
```

Kopieren Sie von ihrer Mint-VM aus, die Dateien **/etc/passwd** *und* **/etc/shadow** *vom Ubuntu-Server in ihr lokales Home-Verzeichnis* **/home/praktikus** *auf der Mint-VM. Sollten Sie eine SSH-Verbindung zum Ubuntu-Server herstellt haben, trennen sie diese zuerst mit dem Befehl exit. Speichern Sie die Dateien unter den Namen „**ubuntu-passwd**“ und „**ubuntu-shadow**“ ab. Stellen Sie etwas fest?*

Um das Problem zu beheben, loggen Sie sich erneut auf dem Ubuntu-Server ein und kopieren Sie die **/etc/shadow** *in das Home-Verzeichnis von gustav mit sudo-Rechten und ändern Sie den Besitzer der Datei zu gustav. Versuchen Sie dann die Datei aus dem Home-Verzeichnis von gustav erneut per* **scp** *zu kopieren.*

Passwortcracking mit John the Ripper

John the Ripper (**john**) ist einer der bekanntesten Passwortcracker. Er ist zwar heute nicht mehr besonders aktuell, aber zur Demonstration für uns ausreichend. **john** funktioniert standardmäßig in drei Etappen: Single, Wordlist und Incremental. Im Single-Modus werden vollständige Passwortdateien sehr schnell und effektiv abgearbeitet. Der Wordlist-Modus verwendete Listen bekannter Passwörter und gleicht diese gegen den Hash ab. Der Inkrementelle Modus ist IMMER erfolgreich! Die Frage ist nur wann… (Bruteforcing).

Um die bereits erwähnte Passwortdatei zu erstellen, nutzen wir wie erwähnt unshadow zum Zusammenführen von passwd und shadow. Dazu nutzen Sie bitte folgende Syntax:

```
$ unshadow <passwd-Datei> <shadow-Datei>
```

Fügen Sie die beiden Dateien zusammen und schreiben Sie die Ausgabe in die Datei passwords.txt! Anschließend können wir unser Glück auch schon mit john probieren. Dazu nutzen wir die einfache Syntax:

```
$ john <Passwortdatei>
```

*Lassen Sie john mit der bekannten Syntax auf die erstellte Passwortdatei los, lassen Sie den Vorgang ca. 2 Minuten laufen und brechen Sie ihn mit „**q**“ oder „**Strg + C**“ ab*. Sie müssten feststellen, dass **john** bereits 2 Passwörter herausgefunden hat. Ein weiteres sollte Ihnen schon bekannt sein. Aber eins fehlt noch. **john** kann weitere Passwortlisten laden. Dazu habe ich Ihnen einen der bekanntesten in gekürzter Form zur Verfügung gestellt. *Laden Sie diese mit dem folgenden Befehl herunter:*

```
$ curl https://tuc.cloud/index.php/s/LK6dNMcgdrrNx5Z/download/rockyou_short.txt -o rockyou.txt
```

Verwenden Sie die eben heruntergeladene Liste zum Brechen des letzten offenen Passwortes! Verwenden Sie die Manpages! Der Vorgang dürfte je nach Systemleistung ca. 1 Minute beanspruchen. Nach dem Finden eines Passwortes können Sie den Vorgang wieder abbrechen.

Anschließend haben Sie bis auf das Passwort des Nutzers gustav, welches Sie ja bereits kennen, geknackt. Sollten Sie auch dieses noch knacken wollen, dann schreiben sie dessen Passwort einfach in eine neue Datei und geben Sie diese als Wortliste an. Bringt das den gewünschten Erfolg?

john beendet nun automatisch das Knacken der Passwörter, da im angegebenen File keine Hashes mehr offen sind. Die bereits geknackten Passwörter können Sie sich jederzeit anschauen über den Befehl:

```
$ john --show <Passwortdatei>
```

Versuchen Sie sich nun per SSH auf dem entfernten System mit den geknackten Passwörtern anzumelden, um deren Richtigkeit kurz zu überprüfen.

Wie bereits erwähnt ist John the Ripper schon etwas in die Jahre gekommen. Es ist außerdem nur für CPUs konzipiert und daher nicht besonders schnell. Ein aktuelleres Beispiel mit intiuitiver Bedienung ist hashcat. Es ist in der Lage mit der entsprechenden Anbindung auch die im System vorhandene Grafikkarte zu nutzen. Diese hat deutlich mehr Leistung und verspricht somit eine erhöhte Performanz beim Brechen von Passwörtern. Außerdem unterstützt hashcat eine Vielzahl an Hashverfahren und einen vorherigen Benchmarktest auf dem System.

Festzuhalten ist, dass der Erfolg vom Knacken von Passwörtern immer vom Passwort, aber auch vom verwendeten Hashalgorithmus abhängig ist. Im Gegensatz zu Linuxpasswörtern brauchen Windowspasswörter deutlich weniger Zeit, um gebrochen zu werden. Das liegt am unterschiedlichen Hashverfahren. Die NTLM-Hashs unter Windows können deutlich schneller berechnet werden als SHA512 (ca. 40x schneller). Hinzu kommt, das Linux die Passwörter zusätzlich salzt. Außerdem sollten Passwörter nie einen Personenbezug aufweisen, was OSINT-Techniken aushebelt. Außerdem sollten Passwörter aktuell min. 12 Zeichen lang sein und aus Buchstaben, Zahlen und Sonderzeichen bestehen, um die Möglichkeit des Bruteforcens gegen 0 gehen zu lassen. Rechnen wir ein kleines Beispielszenario durch:

Nehmen wir die Konfiguration und die Benchmark-Ergebnisse des IT-Forensik Wikis her:

Beschleunigung:	8x Nvidia GTX 1080 Founders Edition
Geschwindigkeit NTLM:	334 Mrd. Hashs/s
Zeichensatz:	Großbuchstaben (29) + Kleinbuchstaben (29) + Ziffern (10) = 68 Zeichen
Passwortlänge:	12 Zeichen

$68^{12} = 9.774.779.120.406.941.925.376 \approx 9{,}7 * 10^{21}\ Möglichkeiten \mid /334 * 10^{9}\ H/s$

$= 29.265.805.749{,}72\ s \mid /60$

$= 487.763.429{,}16\ m \mid /60$

$= 8.129.390{,}49\ h \mid /24$

$= 338.724{,}60\ d \mid /365$

$\underline{= \mathbf{928\ y}}$

Mit einer Passwortlänge von 12 Zeichen und der einfachen Verwendung von Groß- und Kleinbuchstaben mit Umlauten und Ziffern von 0-9 würden wir mit der genannten Konfiguration etwa 928 Jahre brauchen. (Wenn ich mich nicht verrechnet habe). Das bedeutet, dass wir ohne Vorkenntnis kaum eine Chance haben, solch ein Passwort mit Bruteforcing zu erraten. Außerdem sollten niemals Passwörter verwendet werden, welche in Datenlecks gefunden worden. Das können Sie auf der Seite von haveibeenpwned.com überprüfen und feststellen, ob das eingegebene Passwort bereits Teil eines Datenlecks war.