M346, VM Lab 1.

Lernziele:

- Sie richten ein Netzwerk mit 3 VMs ein (Firewall, Linux Client, Windows Client).
- Sie verbinden sich von einem Windows Client per ssh mit einem Ubuntu Remote Host.
- Sie verwenden eine Public Key Authentifizierung für den passwortlosen Zugriff.
- Sie erstellen ein Schlüsselpaar anhand von Anforderungen.

Form:

Einzelarbeit, Zeit: 60 Minuten

Hilfsmittel:

- M346 VMWare Lab einrichten
- M346 Public Key Authentifizierung
- Internet

Erwartetes Resultat:

Mit Printscreens dokumentierte Antworten

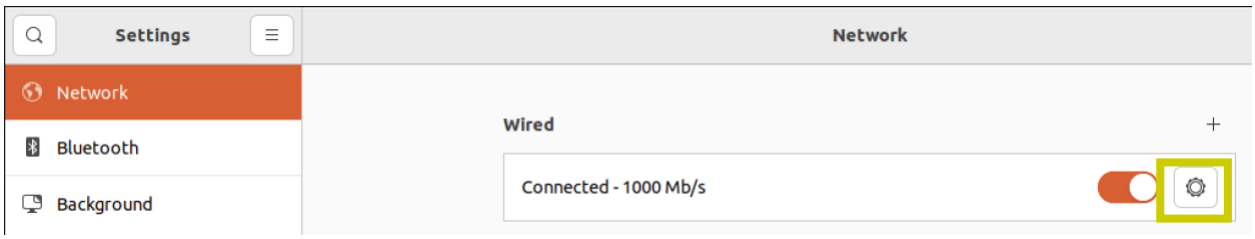
Aufgabe 1: VM-Ware Netzwerk einrichten

Richten Sie ein Netzwerk mit folgenden VMs ein:

- Firewall LF25 (**keine Änderung erforderlich, muss immer als erstes gestartet werden!**)
- Windows Client WP-1-21H2 mit IP-Adresse 192.168.210.2
- Ubuntu-Client LP-22.04_LTS mit IP-Adresse 192.168.210.3

Ubuntu Client konfigurieren:

Unter Settings – Network- Wired - «Tool Settings» IPV4 kann die IP-Adresse zugewiesen werden,

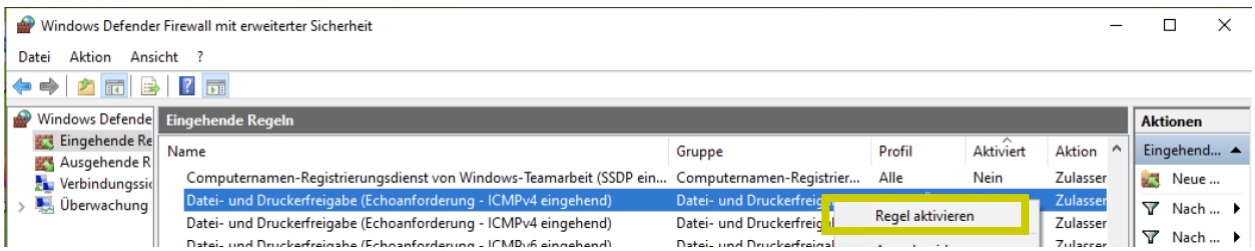


Hinweis: Nach einer Änderung muss der «Schieber» unter Wired «Aus» und wieder «Ein» geschaltet werden. Erst danach sind die getätigten Einstellungen aktiv.

Windows Client konfigurieren:

In den Windows-Einstellungen «Netzwerk und Internet» kann die IP-Adresse zugewiesen werden.

In Einstellungen «Windows Defender Firewall mit erweiterter Sicherheit» muss unter «Eingehende Regeln» die Regel «Datei- und Druckerfreigabe (Echoanforderung - ICMPv4 eingehend)» aktiviert werden. Sonst ist kein eingehender Ping möglich.



Prüfen Sie Ihre Umgebung wie folgt:

Windows-Client:

Prüfen Sie in der Konsole, ob die IP-Adresse korrekt zugewiesen ist:
*Hinweis: Verwenden Sie den Befehl „**ipconfig**“.*

Prüfen Sie, ob der Ubuntu-Client von Ihrem Windows-Client erreichbar ist:
*Hinweis: Verwenden Sie den Befehl „**ping**“.*

Ubuntu-Client:

Prüfen Sie im Terminal, ob die IP-Adresse korrekt zugewiesen ist.
*Hinweis: Verwenden Sie den entweder den Befehl „**hostname**“ mit Option **-I** oder „**IP**“.*

Prüfen Sie, ob der Ubuntu-Client von Ihrem Windows-Client erreichbar ist:
*Hinweis: Verwenden Sie den Befehl „**ping**“.*

Aufgabe 2: SSH Windows -> Ubuntu (mit User / Passwort)

Mittels SSH (Secure Shell) werden Sie nun eine verschlüsselte Verbindung zu Ihrem Ubuntu-Client als Remote-Host herstellen.

Installieren Sie zuerst den für einen Zugriff erforderlichen SSH-Server mit folgenden Befehlen:

```
sudo apt-get update
sudo apt-get install openssh-server
```

Öffnen Sie nun von Ihrem Windows-Client aus eine ssh Verbindung auf den Ubuntu Remote-Host.
*Hinweis: der Befehl lautet **ssh [username]@[remote_host]**, der zu verwendende User ist **vmadmin***

```
ssh vmadmin@192.168.210.150
```

Erstellen Sie auf dem Windows-Client per ssh auf dem Remote-Host unter ~/dummy.txt ein leeres File.

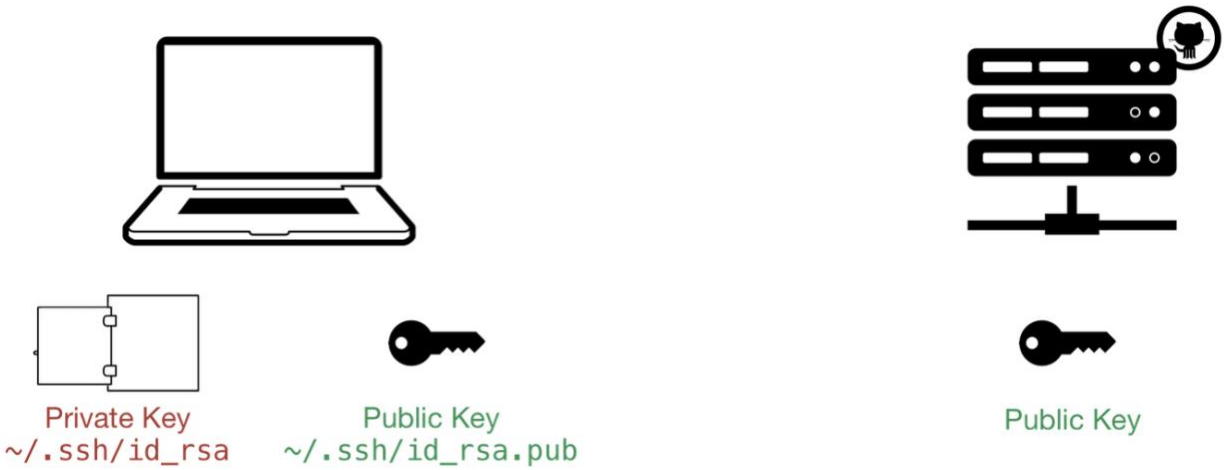
Prüfen Sie auf dem Ubuntu-Client mit Hilfe des Terminals, ob die Datei tatsächlich angelegt wurde.

Stoppen Sie den ssh-Server auf dem Ubuntu-Client und vergewissern Sie sich, dass ein Verbindungsaufbau per ssh nicht mehr möglich ist. Wie lautet die Fehlermeldung?
*Hinweis: Der Befehl lautet **sudo service ssh [stop | start]***

Starten Sie den ssh-Server auf dem Ubuntu-Client wieder und vergewissern Sie sich, dass der Verbindungsaufbau wieder funktioniert.

Aufgabe 3: SSH mit Public Key Authentifizierung

Der ssh-Zugriff soll nun so konfiguriert werden, dass keine Eingabe von Benutzername und Passwort mehr erforderlich ist. Dazu muss auf dem zugreifenden Client der private Schlüssel (Private Key) und auf dem Remote-Rechner der dazugehörige öffentliche Schlüssel (Public Key) korrekt abgelegt sein.



Der private Schlüssel wird normalerweise mit dem öffentlichen Schlüssel im Verzeichnis ~/.ssh abgelegt.
*Hinweis: Unter Windows kann ~ für das Home-Verzeichnis nicht verwendet werden. Es muss stattdessen **/users/[user]/** verendet werden.*

Liegt der private Schlüssel an einem anderen Ort, muss dem ssh-Befehl der Pfad zum Schlüssel mit Option – i übergeben werden.

Generieren Sie auf dem Windows-Client ein neues Schlüsselpaar und legen Sie es im Default-Verzeichnis ab.
*Hinweis: Der Befehl lautet **ssh-keygen**.*

Im Windows Terminal "ssh-keygen" eingeben

Der öffentliche Schlüssel muss auf dem Remote-System in Datei ~/.ssh/authorized_keys abgelegt sein. authorized_keys kann mehrere öffentliche Schlüssel enthalten.

Es gibt verschiedene Möglichkeiten, den öffentlichen Schlüssel auf den Remote-Host zu bringen:

- Befehl ssh-copy-id [user]@[remote host] *(Wird von Windows leider nicht unterstützt!)*
- Öffentlichen Schlüssel mit Editor eintragen
- Öffentlichen Schlüssel mit echo und >> eintragen
- Öffentlichen Schlüssel per type-Befehl übermitteln und eintragen

Probieren Sie folgende drei unter Windows funktionierenden Varianten durch.

Variante 1: Öffentlichen Schlüssel mit Editor eintragen:

Kopieren Sie den öffentlichen Schlüssel auf den Remote-Host und nehmen Sie den Inhalt in die Zwischenablage.

Falls auf dem Remote-Host Datei ~/.ssh/authorized_keys noch nicht existiert, erzeugen Sie die Datei.
*Hinweis: Verwenden Sie dazu den Befehl **touch**.*

Öffnen Sie ~/.ssh/authorized_keys mit einem Editor (z.B. nano, vi) und fügen Sie den Schlüssel aus der Zwischenablage ein.

Vergewissern Sie sich, dass der Zugriff per ssh ohne Eingabe von Username und Passwort funktioniert.

Variante 2: Öffentlichen Schlüssel mit echo und >> eintragen

Generieren Sie eine neues Schlüsselpaar auf dem Windows-System.
Hinweis: Das bestehende Schlüsselpaar wird nicht mehr benötigt und kann überschrieben werden.

Kopieren Sie den öffentlichen Schlüssel auf den Remote-Host und nehmen Sie den Inhalt in die Zwischenablage.

Tragen Sie den Schlüssel mit echo und >> in ~/.ssh/authorized_keys ein:
Hinweis: echo "ssh-ed25519 ...." >> ~/.ssh/authorized_keys

Vergewissern Sie sich, dass der Zugriff per ssh ohne Eingabe von Username und Passwort funktioniert.

Variante 3: Öffentlichen Schlüssel per type-Befehl übermitteln und eintragen

Generieren Sie eine neues Schlüsselpaar auf dem Windows-System.
Hinweis: Das bestehende Schlüsselpaar wird nicht mehr benötigt und kann überschrieben werden.

Die komfortabelste Variante ist, den öffentlichen Schlüssel mit einem Befehl an den Remote-Host zu übermitteln und in Datei ~/.ssh/authorized_keys einzutragen:
Hinweis: type [Path to Public Key] | ssh [user]@[remote_host] "cat >> .ssh/authorized_keys"

Beachten Sie, dass diese Variante nur funktioniert, wenn ~/.ssh/ auf dem Remote-Host existiert!
Falls Verzeichnis ~/.ssh/ auf dem Remote-Host noch nicht existiert, erstellen Sie das Verzeichnis.
*Hinweis: Der Befehl lautet **mkdir.***

Vergewissern Sie sich, dass der Zugriff per ssh ohne Eingabe von Username und Passwort funktioniert.

Aufgabe 4: Passwort-Authentifizierung deaktivieren

Nun soll der Login mit Passwort deaktiviert werden. Danach ist mit ssh nur noch eine Authentifizierung per Public Key möglich. Ändern Sie dazu in Datei **/etc/ssh/sshd_config** die Einstellung PasswordAuthentication yes zu **PasswordAuthentication no**.

*Hinweis: Verwenden Sie für die Anpassung einen Terminal-Editor wie **nano** oder **vi**.*

Vergewissern Sie sich, dass der ssh-Zugriff per Passwort nicht mehr funktioniert. Entfernen Sie dafür den Public-Key aus der Datei authorized_keys.

Aufgabe 4: Schlüsselpaare generieren

Studieren Sie die möglichen Optionen des Befehls ssh-keygen.

Generieren Sie folgende Schlüsselpaare:
- Schlüsselpaar mit Schlüsseltyp ECDSA
- Schlüsselpaar mit Schlüsseltyp RSA und einer Schlüssellänge von 1024