Lektion 7 – Laboration 2

Denna labb går lika bra att köra i docker/podman om du vill.

1: ssh-agent

För att slippa skriva in lösenord till sin privata nyckel om igen kan man använda ”ssh-agent”. Här sätter vi upp ”ssh-agent” **i din VM**.

- Kör ”ssh-agent” och se vad kommandot skriver ut.

För att få inställningarna att bita måste man köra de rader som ”ssh-agent” skriver ut, och bara en gång per inloggning. Det finns många sätt att få ”ssh-agent” att köras automatiskt, och beror på vilket OS och distribution man kör.

- Just nu kommer vi starta ”ssh-agent” manuellt. Kör:

```
eval $(ssh-agent)
```

Nu kan du lägga till din **privata** ssh-nyckel med ”ssh-add”. Efteråt kommer du inte behöver logga in, så länge agenten är igång.

- Testa igen att logga in från VM till VM för att verifiera att agenten hjälper till med nyckeln.

2: Fjärrkommandon

Använd ssh från din värd-dator för att kolla systemstatus med hjälp av fjärrkommandon.

- Kolla ledigt diskutrymme
- Kolla minnesutnyttjande
- Kolla alla processer som körs av din användare
- Kolla vilka användare som finns

3: Kopiera ett filträd

- Använd scp för att kopiera filträdet ”/etc/apt” i din VM till en mapp ”aptconf” på din lokala dator. Se manbladet för att hitta flaggan för att kopiera hela trädet.
- Jämför modifieringsdatum et.c. på destinationsfilerna med källan.
- Se manbladet för att hitta flaggan för att bevara datum m.m. vid kopiering.
- Gör om kopieringen med flaggan och verifiera.

4: Lägg till användarspecifik serverkonfiguration

- Se exempelavsnittet i slutet av /etc/ssh/sshd_config för användarspecifik konfiguration.
- Lägg till en konfiguration för någon av dina alternativa användare.
- Konfigurationen ska vägra användaren att logga in, och meddela användaren att kontot bara får användas för tunnling.
    - Tips: använd ”ForceCommand” och ett kommando för att skriva ut text.
- Starta om ssh-servern för att ändringen ska bita.
- Gör så att din ssh-identitet för din vanliga användare tillåts logga in som din alternativa användare genom att lägga till din publika nyckel i den alternativa användarens ”~/.ssh/authorized_keys”.
- Logga nu in som din alternativa användare med ssh. Alltså, nåt sånt här:

```
vanlig@host:$ ssh alternativ@localhost
```

- Verifiera att du får meddelandet du konfigurerat.

5: Sätt upp en tunnel

Det kommer att finnas tillfällen då vi önskar att vi öppnat en speciell port på en server som nu är stängd. Man kan då ta till tunnlar och låta trafiken till/från den oöppnade porten på servern gå igenom den port som vi använder för ssh (22), *eller (2222) om vi kör virtuellt på samma maskin.*

Starta upp din docker/podman som ni gjorde i labb 1.

OBS! Dessa steg är bara för en server som snurrar i VirtualBox,
i vår docker-image finns redan katalogen och servern är redan igång:

1) Skapa en katalog i din hemkatalog som heter **Public**
2) Starta webbservern så här: **python3 -mhttp.server -d Public 8080**

Plot: Till vårt förtret har vi inte access till den http-server som snurrar där på port 8080.

Uppgiften är:

- Skapa en fil i katalogen **Public** i user's hemkatalog som heter **index.html**.
  Den ska innehålla texten **Hello World!** (om du kan html kan du snygga till det)
- Skapa sedan en ssh-anslutning från din riktiga dator (där du har en webbläsare) och öppna en tunnel från lokal port 8900 till serverns port 8080.
- Starta webbläsaren och surfa till **http://localhost:8900**