# Handnotizen Präsentation SSH

## Beginn – SM

* Begrüßung (F1)
  + Namen
  + Stellen heute u.a. den SSH-Dienst vor
* Zuerst – Hands On Vorbereitung:
  + Website aufsuchen, VM runterladen (F2)
  + Relativ groß
  + Wenn keine Berechtigung o. Speicherplatz
  + Im Hands On Gruppen bilden

Inhalt - TB

* Kurze Einführungen
  + Geschichte und allgemeines
    - Telnet
    - RDP
    - VNC
    - TeamViewer
* Genaueres Vorstellen von
  + SSH
    - Geschichte
    - PDU Aufbau
    - Verwendungsbereiche
* Hands-On

Telnet Geschichte - SM

* Teletype Network
* Entwickelt 1969 für ARPANET
  + Grundlage des Internet, damit älter als das Internet von heute
* Ziel:
  + Remote nutzbare Rechenzeit, Anwendungen und Datenbanken
* Erster Einsatz 1974

Telnet Allgemein – TB

* Dienst für
  + Virtueller Computerzugriff
  + Bereitstellung bidirektionaler textbasierter Verbindung
* Nach Möglichkeit nicht mehr nutzen
  + Weil Plain-Text, alles unverschlüsselt
* Well known port : 23

RDP Geschichte - SM

* Remote Desktop Protokoll
* Proprietärer Service für Fernzugriff auf überwiegend Windows-Maschinen
* Entwickelt 1998 von Microsoft
* RDP als Dienst ist Verfügbar ab Windows NT4 o. WinXP SP1

## RDP Allgemein 1 – TB

* Ermöglicht Übertragung grafischer Inhalte
* Stellt Peripherie bereit
* Verfolgt Cient/Server-Modell
  + Heißt Gerät A hat die Remote Desktop Services
  + Alle anderen haben einen Verbindungs-Client

RDP Allgemein 2 – SM

* Minimum an Input wird an den RDP-Server geschickt
* Und Minimum an Output an den Client geschickt
* Well known Port ist 3389

## VNC Geschichte – TB

* Virtual Network Computing
* In den 90ern im Olivetti Research Laboratory entwickelt
* Ist seit 1998 unter der GNU General Public License lizensiert

## VNC Allgemein – SM

* Arbeitet mit eigenem Protokoll (Remote Framebuffer Protokoll)
* Und ist damit Plattformunabhängig
* Dieses Protokoll wurde März 2011 in einem RFC veröffentlicht
* Well known Ports sind 5800 und 5900

## TeamViewer Geschichte – TB

* Proprietäre Fernwartungssoftware
  + Aus Faulheit oder Optimierungsbedarf denn
  + Der Hersteller wollte unnötige Kundenbesuche meiden
* Erstmalig 2005 veröffentlicht und schrittweise ausgebaut

## TeamViewer Allgemein – SM

* Auf fast jedem gängigen Betriebssystem Verfügbar
  + Windows, macOS, iOS, Android und Debian/Red Hat basierte Linux Distributionen
* Erfordert TeamViewer auf beiden Geräten
* End to End Verschlüsselt über RSA und AES
* Well-Known-Port ist 5938

SSH Geschichte - TB

* Steht für Secure Shell
* Entwickelt 1995 von Tatu Ylönen
  + Bemerkte Sniffing-Angriff im Netz der Uni Helsinki auf RSH und Telnet Verbindungen
    - Remote Shell-Zugriff war bis dato über RSH/rlogin und Telnet möglich
    - Alles Plaintext-Dienste/Protokolle aus Prä-Internet Zeit

SSH Allgemein 1 & 2 – SM

* Gedacht für eine sichere Remote Shell-Verbindung
* Eigentlich in erster Linie für die Verschlüsselung von Datenaustausch über TCP bzw. Absicherung der Transportschicht
* Sichtbar ist:
  + Dass eine SSH-Verbindung besteht
  + Möglicherweise wieviele Datenpakete
  + Und möglicherweise In welcher Geschwindigkeit
* Nicht sichtbar:
  + Der Inhalt
* Beispiel: Amazonpakete vor der Haustür der Nachbarn

SSH Allgemein 3 – TB

* Aufbau eines SSH Pakets
  + Zuerst die Paketlänge in Plaintext mit 4 Bytes
  + Dann die Paddinglänge mit 1 Byte in Verschlüsselt
    - Gleich Erklärung von Padding
  + Danach kommt die Payload in Verschlüsselt
    - Kann grundsätzlich alles sein, es muss nichts mit Shell zu tun haben!
  + Dazu noch das Padding, bestehend aus Random Bytes, in Verschlüsselt
    - Damit wird die Verschlüsselung sicherer, man kann nicht auf die Größe der Payload schließen
    - Beispiel von vorhin: weitere leere Amazonpakete zusätzlich
      * Oder alle Pakete gleichgroß und der Rest mit Luftpolsterfolie gefüllt
  + Zuletzt der Authentifizierungscode in Plain Text
    - Zum Überprüfen der Authentizität

## SSH Allgemein 4 – SM

* Ablauf
  + Aufbau TCP-Verbindung
    - Muss aber nicht, auch Websockets möglich
  + Server Identifziert sich
  + Es wird eine Symmetrische Verschlüsselung eingerichtet
  + Der Client authentifiziert sich
  + Dabei wird ein SSH-Channel pro Verbindung eingerichtet
    - Damit ist Multiplexing möglich,
    - es können mehrere Verbindungen zwischen dem selben Client und Server parallel laufen

## SSH Hands-On – TB

* Vorbereitung
  + In diesem Git Repository findet ihr die Aufgaben und ggf. Erklärungen
  + Diese Bitte entweder alleine oder in Gruppen bearbeiten
  + Nicht alle Aufgaben müssen am Ende des Hands-On erledigt sein, insbesondere Aufgabe 5
* Folgende Aufgaben
  + 1.) SSH-Server in der Ubuntu-VM einrichten
  + 2.) Verbinden und Authentifizieren via Kennwort
    - Dann den eigenen SSH-Key auf dem Server ablegen
  + 3.) Verbinden und Authentifizieren via SSH-Keys
  + 4.) Upload und Download von Dateien via SCP (Secure Copy)
  + 5.) Wenn ihr noch Zeit habt Graphische Programme über SSH remote nutzen
    - Soll noch mal verdeutlichen, dass SSH an sich nichts mit Shells zu tun hat
    - Dafür müssen Client und Server unter Linux laufen
    - Wenn ihr einen Windows-Rechner nutzt könnt ihr probieren
      * Es unter WSL2 zu machen oder
      * Die VM zu klonen und von VM zu VM zu verbinden