Hausaufgaben Laborübung 04 – Einfache Netzwerke

Aufgabe A - Planung des physischen Netzes

Sie planen in Vierergruppen die Netzinfrastruktur für ein kleines LAN mit je vier Rechnern.

1.) Machen Sie sich die Funktion der einzelnen Rechner- & Netzwerkkomponenten klar.
   - Rechner – inkl. VM & Peripherie (Monitor, Tastatur etc.)
   - Netzwerkkabel – Aufgabe im Netzwerk
   - Switch – Aufgabe im NW & Einordnung ins OSI-Modell
   - Ethernet-Port – physikalisches Netzwerkinterface

2.) Recherchieren Sie entweder mit [1, S. 461ff] oder mithilfe der folgender Links was eine Netzwerktopologie ist.
   - https://www.elektronik-kompendium.de/sites/net/0503281.htm
   - https://en.wikipedia.org/wiki/Network_topology
   - https://www.lifewire.com/computer-network-topology-817884

3.) Wählen Sie eine geeignete Netztopologie und skizzieren Sie diese mit geeigneten Symbolen.
Hinweis: Unter http://iacis.org/iis/2008/S2008_967.pdf finden Sie auf S. 241 eine Möglichkeit, wie dies aussehen könnte.
Ordnen Sie die Geräte auf der Skizze so an, wie sie auch vor ihnen im Raum bzw. auf dem Tisch angeordnet sein sollten.

4.) Planen Sie die Netzkonfiguration:

(i) Rekapitulieren Sie kurz was eine IP-Adresse ist. Welche Aufgabe haben diese Adressen in einem Netzwerk?
Hinweis: Ein guter Start wäre: [1, S. 331ff]

(ii) Momentan werden vor allem *IPv4* und *IPv6* als Netzwerkschichtprotokolle genutzt. Recherchieren Sie einige wichtige Unterschiede zwischen *IPv4* und *IPv6*.

(iii) Recherchieren Sie was eine Subnetzmaske ist und wofür diese gebraucht wird.

(iv) Wie spielen IP-Adresse und Subnetzmaske zusammen?

(v) Bestimmte IP-Adressbereiche werden nicht ins Internet weitergeleitet, sie werden als private IP-Adressen bezeichnet. Diese Adressen gibt es sowohl unter *IPv4* als auch unter *IPv6*. Recherchieren Sie, welche IP-Adressbereiche nicht ins Internet geroutet werden.

(vi) Wählen Sie beispielhaft eine Netzwerkadresse (IP-Addresse – ip address) und Subnetzmaske (subnet mask) für einen möglichst kleinen IP-Adressbereich, der genau für vier Rechner ausreicht.

(vii) Sollten Sie die Berechnung von IP-Ranges in der Vorlesung noch nicht behandelt haben, nutzen Sie folgende Links:

- `https://www.calculator.net/ip-subnet-calculator.html`
- `https://www.tunnelsup.com/subnet-calculator/`

Bitte stellen Sie spätestens in der Übung sicher, dass Sie die Berechnung der IP-Ranges anhand der Subnetzmaske verstanden haben.

Aufgabe B – Tools

Um den Übungsbetrieb etwas effizienter nutzen zu können, sollen Sie sich zunächst mit den Standardwerkzeugen der Netzwerkadministration vertraut machen. Mithilfe der Werkzeugsammlungen *iproute2* sowie *net-tools* wird dies in der Regel unter Linux und Unix-Betriebssystemen bewerkstelligt.

1.) Im ersten Übungsblatt haben Sie bereits das Rechtemodell kennengelernt. Verschiedene Nutzer*innen haben verschiedene Rechte. Für die Konfiguration des Systems soll im allgemeinen nur bestimmte Nutzer*innen zuständig sein. Recherchieren Sie welche Rechte der *root*-User hat und was das Kommando *sudo* in diesem Zusammenhang leistet.

2.) In Betriebssystemen gibt es verschiedene Dienste/ Hintergrunddienste (Daemons), die die Verwaltung des Systems in Teilen organisieren. Da Debian (bzw. Arch Linux) das Betriebssystem auf den Rechnern ist, kommt Systemd zum Einsatz (Mglw. wird auch FreeBSD eingesetzt). [1]

a.) Recherchieren Sie einige wichtige Dienste, die durch Systemd gesteuert werden.

b.) Systemd verfügt über die Möglichkeit bestimmte Dienste zu starten, stopen, etc. Recherchieren Sie wie der entsprechende Befehl lautet. Das Wiki bzw. die Man-Page ist eine gute Anlaufstelle!
Notieren Sie sich die Syntax Wort für Wort, sowie die Bedeutung jedes Wortes (Tokens).

[1] Eigentlich war Systemd als Alternative des System-V Init-Daemons gedacht, hat aber über die Zeit immer mehr Funktionalitäten bekommen.

c.) Wichtige Dienste für die nächste Laborübungen sind der Networking-Service und DHCP. Notieren Sie sich:

i Wie der Status eines Daemons abgefragt werden kann.

ii Wie ein Daemon gestartet, gestoppt werden kann.

iii Wie ein Daemon permanent eingeschaltet bzw. ausgeschaltet werden kann (d.h. auch nach einem Neustart automatisch gestartet werden kann.)

3.) Übliche Befehle zum Einrichten von Netzwerkadaptern sind *ifconfig* (BSD *net-tools*) oder auch *ip* aus der Werkzeugsammlung *iproute2*. Der Befehl *ifconfig* gilt in manchen Linux-Distributionen als veraltet (In BSD, Solaris etc. ist dies nicht der Fall!). Recherchieren Sie kurz, worin sich beide Tools-Sammlungen unterscheiden und notieren Sie sich wesentliche Unterschiede.
Digital Ocean hat ein schönes HowTo dazu: goo.gl/w1MN5x

4.) Bringen Sie in Erfahrung, wie Sie die Konfiguration bereits existierende Netzwerkkonfigurationen mit den Tools *ip* und *ifconfig* in Erfahrung bringen.

5.) Recherchieren und notieren Sie sich, wie mithilfe des Befehls *ip addr* Netzwerkadapter(n) eine (oder mehrere) IP-Adressen und Subnetzmasken zugewiesen wird.
Wie wird dies mit *ifconfig* gehandhabt.
(Auch hier gilt: Notieren Sie sich das Kommando sowie dessen Bedeutung Wort/-Schrittweise)!

6.) Recherchieren Sie, wie Sie die IP-Konfiguration in einer Datei festlegen und speichern können, sodass diese weiterhin nach einem Neustart gültig ist.
Achtung: Bedenken Sie für welches Betriebssystem diese Konfiguration erfolgen soll!

a.) In welcher Datei wird die Konfiguration abgelegt?

b.) Welcher User kann auf diese Datei zugreifen?

c.) Notieren Sie sich, wie eine Konfiguration beispielhaft aussieht und was die einzelnen Zeilen bedeuten!

7.) Recherchieren Sie wie der Status eines Netzwerkadapters mit den *net-tools* und *iproute2* abgefragt werden kann.
Welche Stati kann ein Adapter besitzen?
Wie kann der Status geändert werden?

Aufgabe C – Ping

Um festzustellen ob eine Verbindung funktionstüchtig ist, wird oftmals das Tool *ping* genutzt. D.h. *ping* analysiert ob Datenpakete überhaupt und wie viele Pakete von einem Host (bspw. Ihrem Rechner) zu einem Ziel (wie etwa der Webserver der HTW-Berlin) gelangen. Falls Sie ein wenig mehr zu Ping recherchieren wollen, kann ich Ihnen folgenden Artikel empfehlen: https://openmaniak.com/ping.php

1.) Recherchieren Sie die Syntax von *ping*. Ein guter Anlaufpunkt wäre die Man-Page (*man ping*) oder https://linux.die.net/man/8/ping.

2.) **Optional:** Arbeiten Sie folgendes Tutorial durch: https://www.thegeekstuff.com/2009/11/ping-tutorial-13-effective-ping-command-examples/

Aufgabe D – Address Resolution Protocol (ARP) & Neighbor Discovery Protocol (NDP)

Vielleicht ist Ihnen aufgefallen, dass Ihr Netzwerk in der Planung zwar IP-Adressen nutzt, aber kein Router Verwendung findet (Router/Gateways arbeiten fast immer auf OSI-Layer 3). Der Switch (`OSI-Layer 2`) benötigt keine IP-Adressen, dieser arbeitet unterhalb des Network-Layers und ist lediglich auf Ethernet-Frames angewiesen. Ihre VMs verlangen jedoch zwingend eine IP-Adresse von Ihnen.
Um den Knoten ein wenig zu lösen, schauen Sie sich das *Address Resolution Protocol (ARP)* an.

1.) Recherchieren Sie im Kurose [1, S. 461ff, 465] oder mithilfe folgenden links: https://en.wikipedia.org/wiki/Address_Resolution_Protocol, was *ARP* ist und wie dies funktioniert.

2.) Wie adressiert ein Switch die Pakte zwischen den Endknoten (VMs)?

3.) Erläutern Sie das Adressschema von *MAC*-Adressen. Kann dieses Adressschema auch zu Problemen führen?

4.) Da unser Uplink (Gateway des Labors) „nur“ das alte *IPv4* spricht ist *ARP* notwendig. Unter *IPv6* gibt es kein *ARP*, wie wird dies dort gehandhabt?

5.) Erklären Sie wie die Adressauflösung mittels *NDP* aussieht? Welche Schritten sind hier notwendig?

6.) Recherchieren Sie wie die Werkzeuge *arp* und *ip neigh* in unixoiden Betriebssystemen genutzt werden können, sowie deren Syntax.

a.) Wie kann der *arp*-Cache ausgelesen werden?

b.) Wie löscht man *arp* Einträge?

c.) Wie kann in Wireshark nach ARP-Nachrichten gefiltert werden?

Aufgabe E – MITM & ARP-Cache

ARP besitzt keinerlei Mechanismen, um die Nutzer vor Angriffen zu schützen. Mit der folgende Aufgabe sollen Sie herausfinden, wie hoch der Aufwand für eine solche Manipulation ist.
Hinweis: Die hier vorgestellten Techniken sollen Ihnen ermöglichen Angriffsszenarien zu verstehen. Nicht diese in fremder Infrastrukturen anzuwenden. Die Skripte sollten Sie nur im Labor oder dem eigenen Netzwerk testen!

1.) Recherchieren Sie, wie die Datenstruktur „Cahce“ funktioniert. Was sind die Eigenschaften eines Caches?

2.) Anschließend daran: Recherchieren Sie was es mit dem ARP-Cache auf sich hat.

3.) Erläutern Sie wie der ARP-Cache-Mechanismus funktioniert.

4.) Recherchieren Sie was ein Man-In-The-Middle-Angriff (*MITM*) ist.

5.) Da der ARP-Cache keinerlei Validierungsmöglichkeiten hat, ist ein Manipulation des ARP-Caches möglich. Überlegen Sie sich zunächst, welche Schritte hierfür notwendig wären, wenn Sie als Angreifer den Cache eines anderen Systems verändern wollen.

- Welche Voraussetzungen müssen gegeben sein?
- Welche Informationen über das Angriffsziel benötigen Sie?
- Welche Schritten müssen für den Angriff erfolgen. Denken Sie zunächst abstrakt darüber nach.
- **Anmerkung:** Falls Sie die vorige Aufgabe nicht bewerkstelligen konnten, lesen Sie ein Tutorial zu ARP-Cache-Poisoning.
  Bspw.: https://www.tutorialspoint.com/ethical_hacking/ethical_hacking_arp_poisoning.htm

6.) Im Moodle (sowie auf den VMs) steht ein Angriffstool für das sogenannte ARP-Spoofing bzw. ARP-Cache-Poisoning bereit. Lesen dieses Skript und versuchen Sie diese weitestgehend zu verstehen.
Notieren Sie sich den Ablauf! Notieren Sie sich Fragen zum Quellcode, die Sie nicht verstehen.

Literatur

[1] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach (6th Edition)*. Pearson, 6th edition, 2012.