Allgemeine Hinweise

Gruppen

Es wird in Zweiergruppen gearbeitet und abgegeben. Die Abnahme erfolgt hierbei vor Ort und als Gruppe.

Umgebung

Das ganze Praktikum findet in einer virtualisierten Umgebung statt. Hierbei wird so weit wie möglich Docker verwendet, um die Netze aufzubauen. Sollten wir die Grenzen Dockers erreichen, so wechseln wir evtl. partiell auf VMs.

Docker

Docker ist an sich keine Virtualisierung, sonder eine Containerisierung. Hierbei werden die Container voneinander isoliert, so dass diese sich nicht gegenseitig beeinflussen. Des weiteren werden virtuelle Netzwerke geschaffen über diese die Container kommunizieren können.
Für den Großteil dieses Praktikums sind die Details von Docker irrelavant und es können die bereitgestellten Container wie eigenständige Maschinen behandelt werden.

Management Oberfläche

Für die Container der Studenten existiert eine Management Oberfläche, welche genutzt wird, um das ganze Projekt zu starten und zu verwalten. Diese Oberfläche ist unter `141.22.34.6` zu erreichen. Über diese Oberfläche können die Container gestartet und gestoppt werden.
Es ist eine Eigenentwicklung und hat evtl. noch ein paar Fehler. Wir bitten es uns nachzusehen.

Dokumentation und Scripte

Es wird stark angeraten alle Schritte genau zu dokumentieren oder besser alle Schritte als Script festzuhalten!
Wir übernehmen keine Garantien dafür, dass nicht mal Container abhanden kommen oder durch andere Studenten gelöscht werden.
Also bitte haltet nicht nur in den Containern fest was ihr gemacht habt!

Viel Erfolg!

Aufgabe 1.0: Infrastruktur Workaround

Leider haben wir noch etwas mit der HAW-Infrastruktur zu kämpfen. Wir warten immernoch auf den DNS-Eintrag und auch das gültige TLS-Zertifikat.
Deshalb muss entweder in die /etc/hosts folgendes eingetragen werden, oder immer direkt die IP verwendet werden.

```
141.22.34.6 pnskss.informatik.haw-hamburg.de
```

Außerdem muss derzeit das Zertifikat per Ausnahme akzeptieren, da es selbst signiert ist. Im folgenden werden wir pnskss.informatik.haw-hamburg.de aka 141.22.34.6 mit pnskss abkürzen.

Aufgabe 1.1: Zugriff vorbereiten

SSH wird das Mittel der Wahl sein für den Zugriff auf die Netzwerkknoten. Um eine reibungslose Verbindung zu gewährleisten werden wir auf PublicKey-Verfahren setzen zur Authentifizierung. Wer nicht weiß was das im allgemeinen und speziell für SSH ist, liest es bitte nach!
Führt die folgenden Schritte für jedes Teammitglied aus!

- Erstellt ein lokales SSH-Schlüsselpaar
  Sowohl ssh (openssh) als auch Putty stellen dafür Tools zur Verfügung.
  Beachtet, dass Putty ein anderes Format benutzt und die Server OpenSSH verwenden.
- Logt euch auf pnskss an Port 443 per SSH ein und mit eurer HAW-Kennung + Passwort.
  Achtet darauf, dass es sich bei 443 NICHT um den Standard Port handelt.
  Ihr könnt zur Vereinfachung auch direkt euren lokalen SSH-PublicKey übertragen, um euch später ohne Passwort einloggen zu können (ssh-copy-id).
- Erstellt auf pnskss ein SSH-Schlüsselpaar.
- Sammelt alle erzeugten öffentlichen Schlüssel für später zusammen.

Aufgabe 1.2: Projekt Starten

Jetzt ist es an der Zeit das eigentliche Projekt zu starten.

- Geht per https auf pnskss.
- Unter Projects / pns1 / Details das Formular zur Erstellung öffnen und ausfüllen.
  Füllt bei Authorized Keys die zuvor erzeugten Schlüssel ein. Je einer pro Zeile.
  Denkt euch einen Teamnamen aus. **Hängt and disen als Prefix die Übungsgruppennummer an.** Der Name muss eindeutig sein. Derzeit checkt das System das leider nicht und es würden Projekte anderer Studenten überschrieben – sorry ich arbeite dran.

Wählt ein Gruppenpasswort. Darüber wird später jedes Gruppenmitglied Zugriff auf die Container haben (starten/stoppen etc) – work in progress…

- Start It!

Aufgabe 1.3: (Optional) OpenVPN

Da es erweitere Berechtigungen benötigt, sind nur Leute an Laptops in der Lage ein OpenVPN aufzubauen. Nutzt die Gelegenheit! Der Zugriff ist das etwas direkter auf eure Maschinen.

- Ladet hierzu die OpenVPN Config von der Management-Oberfläche herunter.
- Evtl. müsst ihr die IP ändern, wenn ihr keinen /etc/hosts-Eintrag gemacht habt.
- sudo openvpn aaa123.ovpn

Aufgabe 1.4: Explore the Network

Es wird Zeit sich die Umgebung anzugucken!

- Verbindet euch mit pnskss per SSH oder mit dem OpenVPN
- Sucht eure Maschinen.
  Ihr seht von dem SSH-Hop oder durch das VPN wirklich alle Maschinen.
  Identifiziert eure eigenen.
- Loggt euch auf einigen der Maschinen ein.
- Findet heraus, was für ein Betriebssystem läuft und welche Distribution.

Aufgabe 1.5: Netzwerk Scan

Das Netzwerk kann auch gründlicher bzw. automatisch untersucht werden. Hierfür wird häufig das Tool nmap verwendet. Lernt dieses zu bedienen, um folgendes zu lösen:

- Ihr habt auf euren eigenen Maschinen root-Zugriff und könnt somit eigene Softwarepakete installieren. Installiert nmap selbständig wo benötigt.
- Scant alle eure eigenen Netze nach Maschinen und notiert den Output.
- Führt eine OS-Erkennung durch.
- Findet offene Ports auf den Maschinen.

Aufgabe 1.6: Erstellt eine Netzwerk-Karte

Nein, nicht ein Hardware-Interface, sondern eine Karte (map) analog zu eine Landkarte. Hierfür könnt ihr die Informationen der letzten Aufgabe nutzen bzw. die Tools geschickt einsetzen.

- Die Karte soll zeigen welche Knoten mit welchen Netzwerk verbunden sind.
- Die Karte soll enthalten welche Knoten welche Ports geöffnet haben.
- Die Karte soll enthalten welche Interfaces genutzt werden (für welche Netz). Hierfür müsst ihr sehr wahrscheinlich auf die Knoten selbst.
- Die Karte soll natürlich die IPs der Knoten enthalten.
- Die Karte soll die Routen der Knoten enthalten.
- Subnetze sollen gut erkennbar sein.

Aufgabe 1.7: Kommunikation

Derzeit ist die Kommunikation nur innerhalb des jeweiligen Subnetzes möglich. Dies ist nicht unbedingt sinvoll! Deshalb:

- Lernt wie man Routen hinzufügt sowohl mit ip/iproute2 sowie mit route.
- Fügt Routen den Knoten hinzu, so dass eine Kommunikation durch das gesamte Netzwerk möglich ist.
- Notiert die gewählt Route mittels des Tools traceroute für jeweils einen Knoten pro Subnetz.
- Testet die Kommunikation der Knoten mittels nc (netcat).
  Findet hierbei heraus, wie ihr nc als Server (listen) und Client nutz.