

Aufgabenblatt 1¹

Internetzwerke, Protokollverfolgung, Netzwerktools

Ziele

- Protokollabläufe kennenlernen und im Kontext des OSI Referenzmodelles darstellen
- Paketaufbau und –einbettung verfolgen (Stichwort: Encapsulation)
- Mit dem Netzwerksniffer (Wireshark) und Netzwerktools umgehen

Ergebnis

- Maximal 3-seitiges (bezogen auf den Fließtext) Praktikumsprotokoll (deutsch oder englisch), das die angeforderten Teilaufgaben umfasst und in geeigneter Form darstellt. Eine Auflistung von Stichpunkten ist nicht ausreichend. Das Protokoll muss von einer dritten Person ohne Vorlage der Aufgabe verstanden werden können.
- Bearbeiten Sie das Aufgabenblatt bereits vor Beginn des Praktikumstermins vollständig. Halten Sie das Protokoll und Ihren Lösungsansatz zum Abnahmegespräch bereit. Es ist erforderlich, dass Sie Ihren Bildschirm während des Abnahmegesprächs teilen. Überprüfen Sie die *Screen-Share*-Funktionalität vor Praktikumsbeginn in ihrem Team.
- Bei nicht ausreichender Leistung wird vor dem nächsten Praktikumstermin eine weitere Nacharbeit zugelassen. Eine Nichterfüllung im angesetzten Zeitrahmen führt zum Ausschluss.
- Das Protokoll inkl. des Lösungsansatzes ist innerhalb einer Woche nach dem Praktikumstermin abzugeben. Details zur Abgabe finden Sie im MS Teams Raum im Kanal *Praktikum*.

Hinweise

- Der Aufgabenteil B ist als Vorbereitung zu verstehen und ist nicht als Abgabe zu erbringen; einzig, wenn Schwächen bei der Bearbeitung der Aufgaben 2 und 3 erkennbar sind, ist der Nachweis über die Durchführung zu führen.
- Für das erste Praktikum wird nur eine Abgabe des Protokolls für Teil A erwartet.

A Pflichtteil

A.1 HTTP-Dialog

Zeichnen Sie den Webseitenabruf `http://scimbe.de/_index.html` mit dem Netzwerksniffer Wireshark auf. Analysieren Sie den HTTP-Dialog und folgen Sie dem TCP-Strom, der dieser HTTP-Kommunikation zugrunde liegt.

Tipp: Um Nebeneffekte zu vermeiden nutzen Sie bitte eins der folgenden Konsolen-Werkzeuge: `wget`, `curl` oder `netcat`. `wget` ist für alle Plattformen erhältlich. Diskutieren Sie das gerne in MS Teams.

¹basiert auf der Aufgabenstellung von Prof. Thomas C. Schmidt und Prof. Dr. Martin Becke

Protokoll

- Dokumentieren und erklären Sie den HTTP-Dialog.
- Erklären Sie exemplarisch den Aufbau (Schichten) und die Aufgabe der einzelnen Pakete, soweit es in der Vorlesung besprochen wurde. **Nicht darüber hinaus.**
- Diskutieren Sie die Frage, wie es dem Netzwerksniffer gelingt, den zugrundeliegenden TCP-Strom aus den Einzelpaketen zusammenzusetzen (Stichwort: Fragmentierung)? Können Sie die dafür nötigen Informationen identifizieren?

A.2 HTTPS-Dialog

Zeichnen Sie hiernach den Webseitenabruf `https://cads.informatik.haw-hamburg.de/` mit dem Netzwerksniffer auf. Identifizieren und analysieren Sie die zugehörigen Pakete.

Protokoll

- Dokumentieren und erklären Sie zunächst allgemein das veränderte Erscheinungsbild.
- Dokumentieren und erklären Sie welche Informationen über den Aufruf noch zu erkennen sind? Zu welchen Schichten lassen sich die Protokolle im OSI Referenzmodell zuordnen und zu welchen im TCP/IP Modell.
- Erklären Sie die Rolle der Verschlüsselung aus Sicht des Netzwerkschichtenmodells.

A.3 SSH

Versuchen Sie mit einem SSH Client (Konsole oder Putty) eine Verbindung zu `usershell.informatik.haw-hamburg.de` aufzubauen. Nutzen Sie ihre HAW-Login-Informationen (A-Kennung). Identifizieren Sie das Betriebssystem mit `uname`. Beobachten Sie Ihre Tätigkeiten mit Wireshark.

Protokoll

- Stellen Sie Ihre Beobachtungen dar und erläutern Sie diese Beobachtungen.

A.4 IP

Ermitteln Sie ihre lokale IP Adresse mit einem der folgenden Befehle: `ip`, `ifconfig` oder `ipconfig`.

Protokoll

- Dokumentieren Sie alle relevanten Verbindungsinformationen.
- Nennen Sie ihre nutzbaren IP Adressen, Netzadressen und Broadcastadressen.

A.5 VPN

Wählen Sie sich mit einem VPN Client in die HAW ein (siehe `https://www.haw-hamburg.de/vpn/`). Protokollieren Sie die Ihrem Rechner zugeordneten IP-Adressen. Was fällt Ihnen auf?

A.6 Weitere Linux Netzwerk-Werkzeuge

Suchen Sie sich mindestens zwei weitere Werkzeuge und testen Sie diese systematisch. Eine Aufzählung wichtiger Netzwerktools: `arp`, `dig`, `netcat`, `netstat`, `nmap`, `ping`, `socat`, `ss`, `traceroute`.

Protokoll

- Schreiben Sie eine kurze Anleitung wie der Einsatz dieser Werkzeuge gelingen kann.
- Präsentieren Sie Ergebnisse aus Ihrem Testaufbau.

B Multi-Threading Wiederholung

Diese Aufgabe ist eine Vorbereitung für zukünftige Praktikumstermine, mit einer Wiederholung von *C*, Threads, Erzeuger-Verbraucher. Dieser Teil ist nicht verpflichtend, aber angeraten!

Aufgabenstellung

Erstellen Sie eine einfache Konsolen-Applikation `file_creator` in *C*. Die Applikation soll, wie der Name vermuten lässt, Dateien parallel erstellen. Ausgangspunkt ist eine Konfigurationsdatei mit Dateinamen. Jeder Dateiname steht in dieser Konfigurationsdatei einer neuen Zeile.

Anforderungen

- Alle Threads sollen mit dem Einlesevorgang der Konfigurationsdatei gleichzeitig gestartet werden.
- Die Hintergrund-Threads erstellen die Dateien, die ein Einlese-Thread aus der Konfigurationsdatei liest. Jeder Dateiname ist als eigenständiger Auftrag **Task** zu verarbeiten.
- Die Anzahl der Hintergrund-Threads soll beim Start konfigurierbar sein. Die Konfiguration ist über einen Schalter `-T` zu organisieren. Des Weiteren soll es einen Schalter `-Q` geben, um die Größe der Auftrags-Queue festzulegen. Über den Schalter `-h` bzw. `-help` soll ein Hilfetext ausgegeben werden.
- Wie Listing 1 zu entnehmen ist, muss das Argument **FILE** (die Konfigurationsdatei) übergeben werden. Wird die Anwendung ohne Argumente aufgerufen, soll die Usage-Information aus Listing 1 angezeigt werden.
- Testet Eure Anwendung mit geeigneten Listen von Dateinamen.

```
1 Usage: file_creator [OPTIONS...] FILE
2 where
3   OPTIONS := { -Q queue_size | -T number_of_threads | -h[elp] }
```

Listing 1: `file_creator` Usage-Information