Laborübung:

Einstieg in die Socket Programmierung

# 1. Einleitung

🡺 Einleitung

# 2. Vorbereitung

🡺 Einleitung

# 3. Mail

## 3.1 SMTP über telnet

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Der oben zu sehende Screenshot zeigt einen WireShark Trace (Filter auf „smtp“) bei welchem die folgenden Schritte ausgeführt wurden, um eine Mail mittels SMTP Protokoll zu versenden:

1. Starten von „telnet“ in einer cmd-Line
2. Verbindungsaufbau zum Mail-Server mittels:

open asmtp.htwg-konstanz.de 587

1. Senden des „ehlo“ Kommand (siehe No. 64 & 66) mittels:

ehlo asmtp.htwg-konstanz.de

1. Authentifizierung des Login (No. 79 & 81) mittels:

auth login

1. Übermitteln des Nutzer als base64-kodierte Zeichenketten mittels:

cm5ldGlu

1. Übermitteln des Passwort ase64-kodierte Zeichenketten mittels:

VWViZW44ZnVlcjhSTg==

1. Übermitteln des „Mail-From“ Feld mittels:

mail from: <rnetin@htwg-konstanz.de>

1. Übermitteln des „Mail-To“ (rcpt) Field mittels:

rcpt to: <\*\*\*\*\*\*\*\*\*@gmail.com>

1. Ankündigung der Datenübermittlung mittels:

data

1. Übermitteln des Betreff mittels:

Subject: Test

1. Übermitteln der Nachrichtendaten:

Test

1. Beenden der Übertragung mittels:

**Carraige Return + Line Feed**

**.**

**Carraige Return + Line Feed**

* Siehe Folgeseite

Betrachten Sie die Aufzeichnung in WireShark. Was fällt Ihnen auf?

Alle ausgeführten Schritte finden sich in dem WireShark Trace. Da die Daten unverschlüsselt zum SMTP Server übertragen werden sind alle Daten (auch Passwörter) als Klartext lesbar (zwar kodiert lassen sich jedoch leicht decoden)

Schreiben Sie noch eine Mail an ihren Email-Account. Verwenden Sie willkürliche Email-  
Adressen für MAIL-FROM sowie für das „from:“-Feld in der Mail. Siehe auch Wikipedia  
Beispiel. Lesen Sie die Mail in ihrem Postfach. Was fällt Ihnen auf?  
Es ist mittels des SMTP Server möglich Emails auch von anderen Absendern zu versenden (Änderung des *local-part*, vor @ möglich) jedoch scheint der Server den *domain-part* der Mailadresse (nach dem @) zu prüfen. So ist es möglich eine Email von [angela.merkel@rnetin.de](mailto:angela.merkel@rnetin.de) zu senden jedoch nicht von [angela.merkel@bundesregierung.de](mailto:angela.merkel@bundesregierung.de) .

## 3.2 SMTP in Python

Hier wurden die oben händisch ausgeführten Schritte durch ein Python Skript automatisiert ausgeführt (das Resultat bleibt das Gleiche).

Python Code siehe: smtp\_client.py im zugehörigen Unterordner

# 4. Rechner Server

## 4.1 Lokale Kommunikation

1. für jedes gesendete Paket bestimmen, welcher Befehl in welchem Skript (Client/Server) dafür  
verantwortlich ist, dass das Paket gesendet wird.

|  |  |  |
| --- | --- | --- |
| Client | Server |  |
|  | Wartet auf Client | sock.listen(1) |
| Verbindung zu Server aufbauen |  | sock.connect((Server\_IP, Server\_PORT)) |
|  | Verbindung akzeptieren | conn, addr = sock.accept() |
|  | Warten auf Daten [BLK 1] | data = conn.recv(1024) |
| Senden von Daten [FREE 1] |  | sock.send(MESSAGE.encode('utf-8')) |
| Warten auf Server OK [BLK 2] |  | msg=sock.recv(1024).decode('utf-8') |
|  | Senden von OK [FREE 2] | conn.send(data[::-1]) |
|  | Schließen TCP Verbindung | conn.close() |

2. für jeden blockierenden Befehl bestimmen, die Ankunft welches Pakets dafür verantwortlich ist,  
dass die Ausführung des Befehls vervollständigt wird.  
🡪 Siehe oben

## 4.2 Netzwerk-Kommunikation

1. Wie können Sie im Client Python-Skript die IP-Adresse und Port-Nummer des verwendeten  
lokalen Sockets bestimmen („bestimmen“ im Sinne von herausfinden)?

sock.getsockname()

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

2. Wann (in welcher Code-Zeile) und woher erhält ein Client seine IP-Adresse und Port-Nummer?

In Zeile nachdem sich der Client mit dem Server verbunden wurde???. Die abfrage in Zeile 9 führt zu einem Fehler.

3. Wie können Sie im Client-Skript die IP-Adresse und Port-Nummer des Sockets setzen?

Mittels:

sock.bind(('127.0.0.2', 50001))

Der Server zeigt die Änderung wie folgt:

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Die Verbindung über eine IP aus einem andern Netz ist nicht möglich:

sock.bind((179.0.0.2', 50001))

4. Warum müssen Sie Timeouts verwenden und wie funktioniert try ... except? Mit welchem Befehl  
können Sie einen gemeinsamen Timeout für alle Sockets setzen?

Der Timeout wird genutzt damit ein Client nicht endlos auf eine Antwort vom Server wartet sondern nach einer gewissen Zeit (engl.: Timeout, tada) terminiert. Dies wird durch den try, catch Block realisiert. Das Programm/Thread versucht solange eine Antwort zu erhalten bis der socket eine Timeout Exception wirft. Diese wird von except „gefangen“ und die Instruktion entsprechen ausgeführt.

5. Finden Sie experimentell heraus, ob Sie einen Server betreiben können, der ECHO-Anfragen auf  
dem gleichen Port für UDP und TCP beantwortet?

Dies ist möglich wenn ein Server mit UDP & TCP Socket erstellt wird welche in unterschiedlichen Threads auf anfragen wartet. Dies wurde mittels den Python Skripten in Order 4/2/5 getestet.

## 4.3 Unterstützung mehrere Clients

🡺 Siehe Python Skripte im zugehörigen Unterordner

# 5. Port Scanner

## 5.2 Versuch

Hierzu wurde erneut ein Pythonskript genutzt um mittels Threading und den oben beschriebenen Verfahren die offenen Ports zu ermitteln. Das Skript ist unten aufgeführt und im zugehörigen Unterordner zu finden.

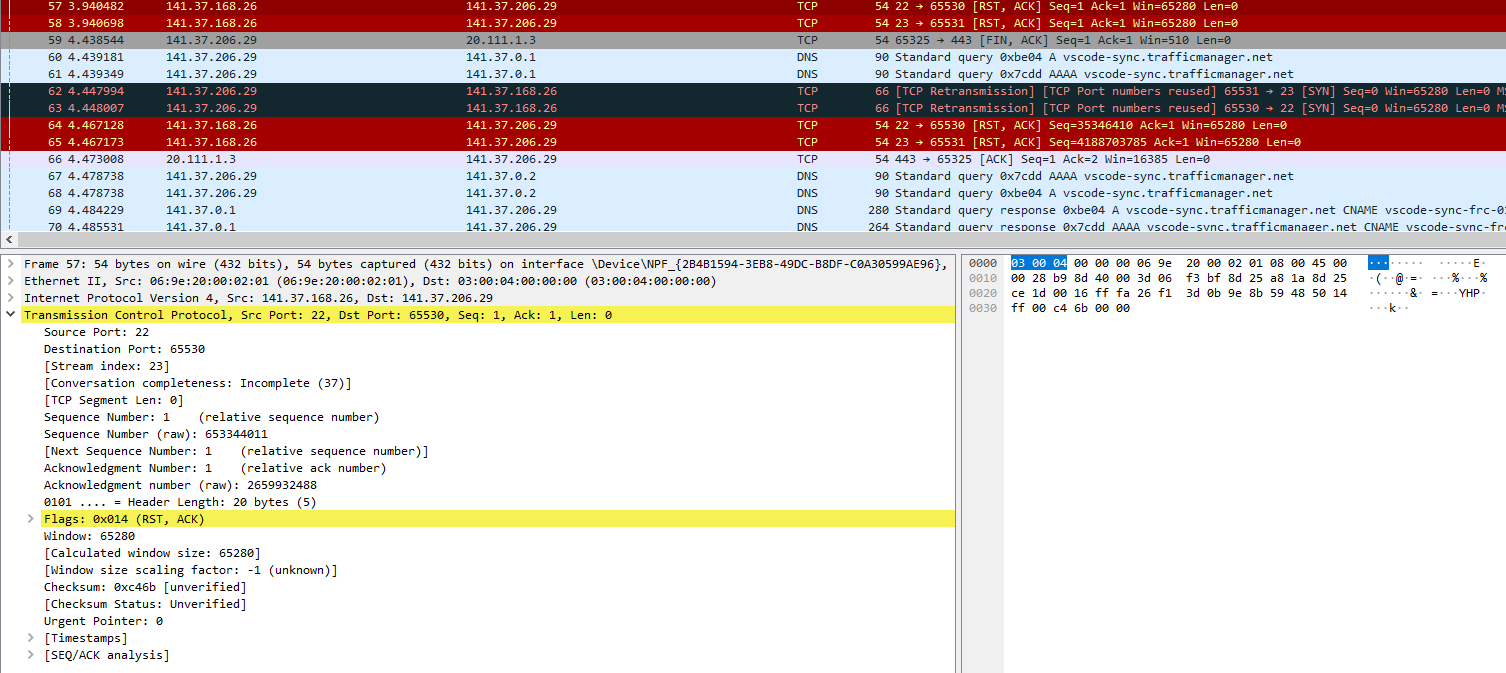
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## 5.3 Fragen

*1. Geben Sie die Liste der offenen TCP und UDP Ports an.*

Derzeit sind alle im Bereich liegenden Ports geschlossen bzw. eine Verbindung wird durch den Server explizit zurückgewiesen (Port 22 & 23 siehe WireShark Trace bzw. Bild)



2. Wählen Sie für TCP und UDP jeweils einen offenen und einen geschlossenen Port und  
erklären Sie die entsprechende Paketsequenz, die Sie in WireShark aufgezeichnet haben.

🡺 Derzeit nicht möglich  
3. Auf Port 7 des Servers läuft ein ECHO-Dienst. Testen Sie ihr Client-Script mit dem ECHO-  
Server. Versuchen Sie das TCP und das UDP Script.

* Derzeit nicht möglich