

Übungsblatt 03 – Einfache verteilte System mit Sockets und HTTP

Aufgabe A – Scapy

Eine gute Anlaufstelle für die Benutzung von *Scapy* finden sie unter: <https://scapy.readthedocs.io/en/latest/introduction.html>.

1. Starten sie eine interaktive *Scapy*-Session mithilfe des Terminals:

```
1 scapy
```

Diese funktioniert analog zur *ipython* Shell der letzten Übung.

- a) Welche *HTTP*-Protokolle und Funktionalitäten unterstützt *Scapy*?

```
>>> explore(scapy.layers.http)
```

- b) Lassen sie in *Scapy* anzeigen, wie eine Default-HTTP-Request aussieht.
 - c) Lassen sie in *Scapy* anzeigen, wie eine Default-HTTP-Response aussieht.
2. Wie können sie einzelne Layer (also Protokolle) laden?
 3. Laden sie den *http*-Layer.
 4. Lesen sie folgenden *HTTP*-Request:

```
load_layer("http")
req = HTTP()/HTTPRequest(
    Accept-Encoding=b'gzip, deflate',
    Cache-Control=b'no-cache',
    Connection=b'keep-alive',
    Host=b'HERE.WEB.URL',
    Pragma=b'no-cache'
)
a = TCP_client.tcplink(HTTP, "HERE.WEB.URL", PORT)
answer = a.sr1(req)
a.close()
with open("HERE.WEB.URL", "wb") as file:
    file.write(answer.load)
```

5. Passen sie den Code an, sodass sie eine *HTTP*-Anfrage an den Webserver der *HTW-Berlin* stellen können (www.htw-berlin.de).

Aufgabe B – Chat-Server

1. Unter folgender *URL* finden sie den Code für den Chat-Server: <https://pastebin.com/raw/XndYUZzn>.

- a) Laden sie den Code mittels *wget* herunter. **Hinweis:**

```
1 wget -O server.py URL
```

- b) Nachdem sie in den Hausaufgaben den Code analysiert haben, soll dieser nun auf der VM ausgeführt werden.
Öffnen sie den Server-Code!
- c) Mithilfe des Befehls *ifconfig em0* können sie die aktuelle IP-Adresse des Rechners in Erfahrung bringen. Notieren sie sich diese und tragen sie diese im Code des Servers ein.
- d) Welche Ports kommen prinzipiell in Betracht für den Server? M.a.W. für welche Ports benötigen sie spezielle Rechte und für welche nicht?
- e) Legen sie im Code den Port für ihren Server fest.
- f) Im Code ist noch kein Interpreter festgelegt worden. Tragen sie in die erste Zeile des Code folgende Code ein:

```
#!/usr/bin/env python
```

Diese Zeile legt fest, dass wir Python als Interpreter für die Ausführung nutzen möchten.

- g) Das Python-Script soll auf unserer VM ausgeführt werden. Hierfür benötigt es entsprechende Ausführungsrechte. Das Kommando *chmod* bietet diese Möglichkeit:

```
1 chmod u+x server.py
```

Geben sie dem Server Ausführungsrechte!

- h) Testen sie anschließend, ob ihr Server tatsächlich funktioniert! Bricht das Python-Script an einer Stelle ab, müssen wir eine Fehlerursachenfindung betreiben.

2. Unter folgender *URL* finden sie den Code für den Chat-Client: <https://pastebin.com/raw/Y4aHSba8>.

- a) Laden sie den Code mittels *wget* herunter. **Hinweis:**

```
1 curl -O client.py URL
```

- b) Nachdem sie in den Hausaufgaben den Code analysiert haben, soll dieser nun auf der VM ausgeführt werden.
Öffnen sie den Client-Code!

- c) In diesem Script müssen ebenfalls IP-Adresse und Port für den Socket angegeben werden. Wozu werden diese Angaben benötigt?
- d) Welche Angaben müssen hier hinterlegt werden?
- e) Legen sie im Code den Port für ihren Server fest.
- f) Im Code ist noch kein Interpreter festgelegt worden. Tragen sie in die erste Zeile des Code folgende Code ein:

```
#!/usr/bin/env python
```

Diese Zeile legt fest, dass wir Python als Interpreter für die Ausführung nutzen möchten.

- g) Das Python-Script soll auf unserer VM ausgeführt werden. Hierfür benötigt es entsprechende Ausführungsrechte. Das Kommando *chmod* bietet diese Möglichkeit:

```
1 chmod u+x client.py
```

Geben sie dem Server Ausführungsrechte!

- h) Testen sie anschließend, ob ihr Client tatsächlich funktioniert! Kann sich der Client mit dem Server verbinden?
- i) Sobald ihr Chat-System funktioniert können sie auch weitere Clients sich auf den Server Verbinden. Wenn sie dem Server eine öffentliche IP-Adresse gegeben haben, können auch andere VMs diese Server erreichen. Testen sie dies!