

# Labreport 02

Tronje Krabbe, Julian Deinert

19. Mai 2016

## Inhaltsverzeichnis

<b>Aufgabe 1 HTTP</b>	<b>2</b>
1.1 Telnet . . . . .	2
<b>Aufgabe 2 SMTP</b>	<b>2</b>
2.1 Mail Spoofing . . . . .	2
<b>Aufgabe 2 License Server</b>	<b>2</b>
3.1 DNS Spoofing . . . . .	2

# Aufgabe 1 HTTP

## 1.1 Telnet

Wir haben versucht uns mit dem Befehl `telnet` mit dem angegebenen Server zu verbinden.

```
$ telnet www.inf.uni-hamburg.de 80
Trying 134.100.56.130...
Connected to www.inf.uni-hamburg.de.
Escape character is '^['.
```

```
GET /de/inst/ab/svs/home.html HTTP/1.1
```

Als Antwort auf unseren GET-Request erhalten wir eine Website mit Returncode 302 `Found`, die uns sagt, dass das Dokument nur mittels `https` erreichbar ist. Da Telnet kein `https` kann, greifen wir auf `openssl` zurück.

```
$ openssl s_client -connect www.inf.uni-hamburg.de:443
```

```
GET /de/inst/ab/svs/home.html HTTP/1.1
```

Wir erhalten den HTTP-Fehlercode 400 `Bad Request` zurück. Dementsprechend können wir auch keine CSS-Dateien anfordern.

# Aufgabe 2 SMTP

## 2.1 Mail Spoofing

Wir verbinden uns mittels *Netcat* mit dem Mailserver `mailhost.informatik.uni-hamburg.de` auf Port 25. Nach dem wir die Felder `FROM`, `RCPT TO` sowie `DATA` gesetzt haben wird unsere Mail erfolgreich versendet. Der Empfänger kann anhand des Quelltextes erkennen, dass die Mail nicht von einem *Authenticated sender* geschickt wurde.

Es gibt keinen Unterschied zwischen einer gespooften gmail oder `informatik.uni-hamnburg` Adresse, da beide Mails direkt beim Mailhost eingereicht wurden.

# Aufgabe 3 License Server

## 3.1 DNS Spoofing

Wir haben uns mit *netcat* zum Server verbunden und werden aufgefordert einen von 4 validen Commands einzugeben. Diese sind:

- `help`
- `serial`
- `version`
- `quit`

Wenn wir eine Serial angeben, die zufällig keine gültige ist, bekommen wir die Meldung `SERIAL_VALID=0` zurück. Anhand dieser Information erstellen wir unseren TCP server so, dass bei jeder Serial die Meldung `SERIAL_VALID=1` zurückgegeben wird.

## 3.2 Eigener License Server