

Hochschule Konstanz Technik, Wirtschaft und Gestaltung (HTWG) Fakultät Informatik

Rechner- und Kommunikationsnetze Prof. Dr. Dirk Staehle

Vorlesung Rechnernetze

Laborübung

Einstieg in die Socketprogrammierung

Prof. Dr. Dirk Staehle

Die Abgabe erfolgt durch Hochladen der bearbeiteten Word-Datei in Moodle.

Bearbeitung in Zweier-Teams

Team-Mitglied 1: Yasmin Hoffmann

Team-Mitglied 2: Chris Jakob

Team-Mitglied 3: Axel Schwarz

1 Einleitung

In dieser Ubung lernen Sie die Programmierung mit Sockets kennen. Die Funktionen zum Umgang mit Sockets sind in jeder Programmiersprache ähnlich, da Sockets vom Betriebssystem angeboten werden. In der Laborübung wird als Programmiersprache Python verwendet. Python ist eine moderne Interpreter-Sprache, die als einfach zu erlernen gilt. Python hat sich in den

letzten Jahren zu einer der populärsten Sprache für die Entwicklung von Netzwerk- und Webanwendungen entwickelt.

Dieser Laborversuch besteht aus drei Aufgaben, die in zwei Laborstunden durchzuführen sind. Im ersten Versuch lernen Sie "telnet" kennen. Telnet ist ein Tool mit dem Sie über die Kommandozeile in einen Socket schreiben bzw. aus einem Socket lesen können. Als Beispiel für telnet dient die tägliche Mail, die Sie dann auch mit einem Python-Skript schreiben dürfen. Der zweite Versuch dient zum Einstieg in die Socket-Programmierung mit Python. Sie implementieren hier einen einfachen Rechenserver. Das dritte Beispiel ist das Erstellen eines Port Scanners, um den Umgang mit Sockets, Threads und Timeouts weiter zu üben.

2 Vorbereitung

Wenn Sie mit Python noch nicht vertraut sind, arbeiten Sie sich in Python ein, indem Sie ein Python-Tutorial durchgehen. Python ist auf den Laborrechnern bereits installiert. Sie können beispielsweise Spyder oder IDLE als IDE nutzen.

Sockets werden in Python von der Bibliothek "socket" unterstützt. Beschreibungen der SocketProgrammierung in Python finden Sie unter

- 1. https://docs.python.org/3/howto/sockets.html
- 2. https://docs.python.org/3/library/socket.html

3 Mail

In diesem Versuch verstehen Sie am Beispiel eines Mail-Clients, wie Anwendungen über Sockets kommunizieren. Sie lernen das Tool "telnet" kennen, mit dem ein Socket geöffnet werden kann, um über die Kommandozeile mit einem Server zu kommunizieren. Im ersten Versuch, bauen Sie eine Verbindung zu einem Mail-Server, um über SMTP eine Mail zu schreiben und über IMAP Mails zu lesen. In einem zweiten Versuch implementieren Sie einen Mail-Client in Python, um Mails zu versenden oder abzurufen.

3.1 SMTP über telnet

Mit telnet können Sie einen Socket zu einem Server auf einem bestimmten Port öffnen und dann über Kommandozeilen-Eingabe mit dem Server kommunizieren. In diesem Beispiel sollen Sie über telnet mit dem SMTP Protokoll Mails verschicken. Öffnen Sie dazu mit telnet einen Socket zum (A)SMTP-Port (587) des Mail-Servers asmtp.htwg-konstanz.de und melden Sie sich mit dem Account (Login: rnetin, Password: Ueben8fuer8RN) an. Schreiben Sie eine Email an einen ihrer Mail-Accounts und prüfen Sie, ob die Mail angekommen ist.

Hinweise:

- 1. Achten Sie darauf, dass Sie sich im VPN befinden.
- 2. Zeichnen Sie ihre SMTP Session mit WireShark auf.
- 3. Sie können "telnet" entweder in der Windowskommandozeile oder über "Putty" öffnen.

4. Gehen Sie vor wie beispielsweise auf Wikipedia beschrieben:

https://de.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol https://de.wikipedia.org/wiki/SMTP-Auth

5. Login und Passwort müssen als Base64-kodierte Zeichenketten übertragen werden. Sie können die Konvertierung mit Hilfe eines Online-Tools durchführen oder die PythonBibliothek base64 verwenden. Importieren Sie dazu die base64-Bibliothek in der Python Shell

(import base64) und verwenden Sie folgende Befehle zum Konvertieren zwischen ASCIIund Base64-Zeichenketten:

- 3. Die Antwort-Mail können alle Labor-Teilnehmer lesen, da alle den gleichen Mail-Account für ping asmtp die Laborübung nutzen.
- 4. Achten Sie bei der Eingabe von Text in telnet darauf, dass Sie sich nicht vertippen. Eine Korrektur ist nicht mehr möglich. Ein praktikabler Umgang mit diesem "Problem" besteht darin, die Kommandos in einem Editor zu entwerfen und dann in das "telnet-Fenster" zu kopieren. Das ist auch hilfreich für die nächste Aufgabe, wenn Sie den Mail-Client in Python implementieren.

→ Verwendete Befehle in telnet:

```
telnet asmtp.htwg-konstanz.de 587

EHLO example.net

AUTH LOGIN

cm51dGlu //username: rnetin

VWViZW44ZnVlcjhSTg== //passwort: Ueben8fuer8RN

MAIL FROM: <yasminhoffmann99@web.de>

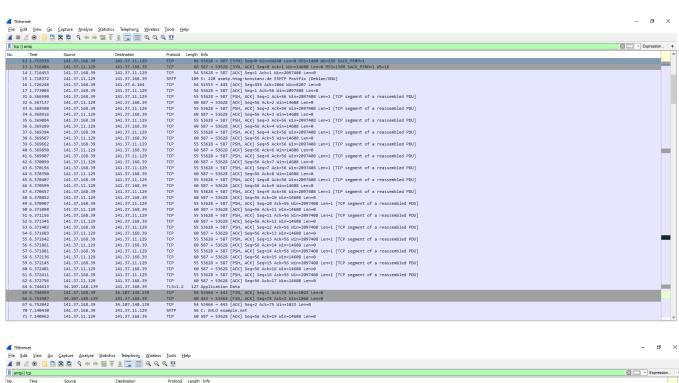
RCPT TO:<yasmin.hoffmann@htwg-konstanz.de>

DATA

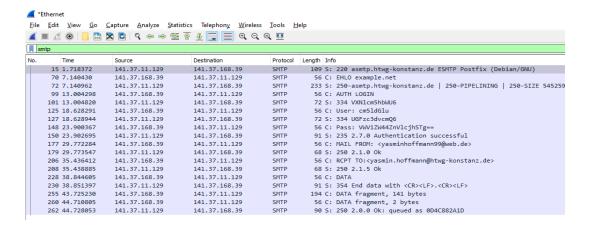
From: <yasminhoffmann99@web.de>
To: <yasminhoffmann@htwg-konstanz.de>
Subject: Testmail1

Hallo,
das ist eine neue Testmail fuer RN.
```

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



The Society Society Control (1987) 1 (1	net				- 1
Section Sect	it <u>V</u> iew	Go Capture Analyze St	atistics Telephony <u>W</u> irele	ess <u>T</u> ools <u>H</u> el	
The Soc	₫ 📵 📗	📙 🛅 🔀 🚨 Q 👄 👄 🖺	🖺 🕡 🎍 🕎 📕 🔍 🤄	ગ્લ્∭	
The Source Debtode Probable Probab	too				⊠ v Expr
66 6.72411	_	Source	Destination	Protocol	reath lafe
46 - 7.7443 14.17.11.12					
66 6.74433 M.187.146.139 M.187.146.139 TC 95.4544 - 4M [FH, ACT] Sept. Ack-72 Min-1023 Lenv8 66 6.753897 M.187.146.139 M.187.146.139 TC 95.45444 - 4M [FH, ACT] Sept. Ack-72 Min-1023 Lenv8 66 6.753897 M.187.146.139 M.187.146.139 TC 95.45444 - 4M [FM, ACT] Sept. Ack-72 Min-1023 Lenv8 67 7.75440 M.187.146.139 M.187.146.139 TC 95.45444 - 4M [FM, Sept. Ack-72 Min-1023 Lenv8 67 7.75440 M.187.146.139 M.187.146.139 TC 95.45444 - 4M [FM, Sept. Ack-72 Min-1023 Lenv8 67 7.75440 M.187.146.139 M.187.					
66 6.753897 341.77.16.3.93 341.77.16.3.93 7CP 54 35846 + 481 [FIJ, ACC) Sept 2 Ack-78 idn-1023 Lenes 67 6.753842 141.77.16.3.93 341.77.16.3.93 7CP 54 55846 + 491 [ACC) Sept 2 Ack-78 idn-1023 Lenes 67 6.753842 141.77.16.3.93 341.77.14.1.19 5FIF 54 55846 + 491 [ACC) Sept 2 Ack-78 idn-1023 Lenes 67 6.753842 141.77.16.3.93 341.77.16.1.93 5FIF 54 55846 + 491 [ACC) Sept 2 Ack-78 idn-1023 Lenes 72 7.146962 141.77.11.129 141.77.16.93 5FIF 54 55846 + 591 [ACC) Sept 2 Ack-78 idn-1023 Lenes 73 7.146962 141.77.11.129 141.77.16.93 5FIF 54 55846 + 591 [ACC) Sept 2 Ack-78 idn-1023 Lenes 74 1.156339 141.77.16.3.93 141.77.11.129 17CP 55 55028 + 597 [CR) Sept 2 Ack-78 idn-1023 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 55 55028 + 597 [CR) Sept 2 Ack-78 idn-1023 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 55 55028 + 597 [CR) Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 55 55028 + 597 [CR) Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 55 55028 + 597 [CR) Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 55 55028 597 [CR) Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn-1029 Lenes 75 7.146439 141.77.11.129 141.77.16.139 7CP 65 597 55028 [ACC] Sept 2 Ack-72 idn					
66 - 67,35867 34,197,148,139 34,197,148,139 170 69 43 + 35464 191, ACC) Sept. Acc. 20 int-1050 (cme) 70 7,140-509 141,77,111,129 141,77,111,129 170 50 51 5100 (acc. 20 51 51) (acc. 20 51					
19.753942 141,71.16.19 34.197.14.11.19 507 54.5144 - 44.3 [ACK] Sept2 Ack-75 Min-14081 Leme 77.746962 141,77.11.11.19 141,77.16.8.39 170 69.97 - 93082 [ACK] Sept-56 Ack-19 Min-14080 Leme 77.746962 141,77.11.12 141,77.16.8.39 170 54.5528 - 557 [ACK] Sept-56 Ack-21 Min-14080 Leme 77.746962 141,77.11.12 141,77.16.19 170 54.5528 - 557 [ACK] Sept-56 Ack-23 Min-14080 Leme 77.746962 141,77.11.12 141,77.16.19 170 54.5528 - 557 [ACK] Sept-56 Ack-23 Min-14080 Leme 77.746962 141,77.11.12 141,77.16.19 170 54.5528 - 557 [ACK] Sept-56 Ack-23 Min-14080 Leme 77.746962 141,77.11.12 141,77.16.19 170 69.97 - 53028 [ACK] Sept-25 Ack-23 Min-14080 Leme 77.74697152 Leme 77.74	6 6.7519		141.37.168.39	TCP	
79 7,144949					
77 7.149902 141,97.11.129 141,97.181.99 17C 69 587 + 53628 [ACK-19 Min-14608 Len-9 72 7.149904 141,97.11.129 141,97.181.93 517 55 53028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 11.65439 141,97.11.129 TCP 54 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 64 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65 55028 + 587 [ACK] Seq-19 Ack-23 Min-2697152 Len-1 [TCP segment of a reassembled POU] 141,97.11.129 TCP 65					
12 1,19982 141,77,11.12	1 7.1409	962 141.37.11.129	141.37.168.39	TCP	60 587 → 53628 ACK1 Seg=56 Ack=19 Win=14608 Len=0
11.545469 141.77.16.3.9 141.77.11.129 TCP 54.5828 + SF ZoCK Seq-19 Ack-253 intin-2897152 Lenn TCP segment of a reassembled POU				SMTP	
92 11.093408 141.77.11.129 141.77.163.39 141.77.11.1129 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 94 11.09409 141.77.11.129 141.77.163.39 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 94 11.09409 141.77.11.129 141.77.163.39 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=20 Win-14608 Lene9 180 17.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 180 17.094139 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 17.094139 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=24 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=24 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 181 181 181 181 181 181 181 181 181	3 7,1948			TCP	
92 11.093408 141.77.11.129 141.77.163.39 141.77.11.1129 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 94 11.09409 141.77.11.129 141.77.163.39 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 94 11.09409 141.77.11.129 141.77.163.39 TCP 60 537 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=235 Ack=20 Win-14608 Lene9 180 13.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=20 Win-14608 Lene9 180 17.094139 141.77.11.129 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 180 17.094139 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=236 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 17.094139 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=23 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=24 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 18.094239 141.77.163.99 TCP 60 587 = 33628 [ACK] Seq=23 Ack=24 Win-2007312 Lene1 [TCP segment of a reassembled PDU] 181 181 181 181 181 181 181 181 181 181					
911.69369 141.37.11.129 141.37.11.129 141.37.163.39 TC 65 57 52628 + SP7 [Psi, AC] Seq-20 Ack-205 Min-2007152 Lenn9 [TCP segment of a reassembled PDU] 913.084298 141.37.11.129 141.37.163.39 TCP 65 57 - SSS26 [ACK] Seq-223 Ack-20 Min-14068 Lenn9 101.10.084209 141.37.11.129 141.37.163.39 TCP 65 57 - SSS26 [ACK] Seq-223 Ack-20 Min-14068 Lenn9 101.10.08420 141.37.11.129 141.37.163.39 TCP 65 57 - SSS26 [ACK] Seq-235 Ack-20 Min-14068 Lenn9 111.17.18.139 141.37.11.129 141.37.163.39 TCP 65 57 - SSS26 [ACK] Seq-235 Ack-20 Min-14068 Lenn9 111.17.18.139 141.37.11.129 141.37.163.39 TCP 65 57 - SSS26 [ACK] Seq-235 Ack-20 Min-14069 Lenn9 111.17.18.139 141.37.11.129 TCP 55 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 111.17.18.139 141.37.11.129 TCP 65 57 - SSS26 [ACK] Seq-235 Ack-20 Min-14069 Lenn9 111.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 111.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 111.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 112.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 112.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 112.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 112.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-235 Ack-20 Min-14069 Lenn9 112.18.18.139 141.37.11.129 TCP 65 SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 112.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 112.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 112.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 112.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 112.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 113.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-14069 Lenn9 113.18.139 141.37.11.129 TCP SSS26 SSS [PSi, ACK] Seq-23 Ack-20 Min-1406	2 11.693	3488 141.37.11.129	141.37.168.39	TCP	
94 11,0946699 141,37,11.129 141,37,161,39 TCP 65 587 + 33028 [ACK] Seq.235 Ack-29 Min-14608 Lene9 19.10,04299 141,37,161,39 141,37,161,39 STCP 66 587 + 33028 [ACK] Seq.235 Ack-29 Min-14608 Lene9 100 13,004429 141,37,11.129 141,37,161,39 STCP 66 587 + 33028 [ACK] Seq.235 Ack-21 Min-14608 Lene9 101 13,004429 141,37,11.129 141,37,161,39 STCP 72 51,314 VORIGINSHOOD 101 17,004520 141,37,11.129 141,37,161,39 STCP 72 51,314 VORIGINSHOOD 101 17,004520 141,37,11.129 TCP 54 55028 - 587 [ACK] Seq.231 Ack-253 Min-2097152 Lene1 101 17,004520 141,37,161,39 141,37,11.129 TCP 54 55028 - 587 [PSM, ACK] Seq.231 Ack-253 Min-2097152 Lene1 101 17,004520 141,37,161,39 141,37,161,39 TCP 66 587 + 58028 ACK] Seq.232 Ack-253 Min-2097152 Lene1 101 17,004520 141,37,161,39 141,37,161,39 TCP 67 587 FSM, ACK] Seq.231 Ack-253 Min-2097152 Lene7 [TCP segment of a reassembled PDU] 101 17,004520 141,37,161,39 TCP 67 587 FSM, ACK] Seq.232 Ack-253 Min-2097152 Lene7 [TCP segment of a reassembled PDU] 101 17,004520 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-14668 Lene9 102 151 16,005373 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-14668 Lene9 103 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-14668 Lene9 104 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-2097152 Lene9 104 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-2097152 Lene9 104 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.235 Ack-30 Min-2097152 Lene9 104 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.237 Ack-41 Min-14608 Lene9 105 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.237 Ack-42 Min-14608 Lene9 105 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.237 Ack-43 Min-14608 Lene9 105 141,37,161,39 TCP 68 587 + 5826 ACK] Seq.237 Ack-43 Min-14608 Lene9 107 142 143,37,161,39 TCP 68 587 + 5826 ACK,30 Min-2097152 Lene9 108 141,37,161,39 TCP 68 587 + 5826 ACK,30 Min-2097152 Lene9 108 141,37,161,39 TCP 68 587 + 5826 ACK,30 Min-2097152 Lene9 109 141,37,161,39 TCP 68 587 + 5826 ACK,30 Min-2097152 Lene9 109 141,37,161,39 TCP 68 587 + 5826 ACK,30 Min-2097152 Lene9 109 141,37	3 11.693	3563 141.37.168.39	141.37.11.129	TCP	
99 13 (84298 141.37.11.129 141.37.11.129 58TP 58 CE AUTH LOGIN 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	4 11.694	4069 141.37.11.129	141.37.168.39	TCP	
180 13.084506 141.77.11.129 141.77.183.9 141.77.11.129 TC P5 55328 - 587 [PS], ACK] Seq-31 Ack-23 Min-2897152 Len-0 181 17.83944 141.77.183.99 141.77.11.129 TC P5 55328 - 587 [PS], ACK] Seq-31 Ack-23 Min-2897152 Len-1 181 17.83945 141.77.183.99 141.77.11.129 TC P5 55328 - 587 [PS], ACK] Seq-31 Ack-23 Min-2897152 Len-1 182 17.83946 141.77.183.99 141.77.11.129 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 17.83946 141.77.183.99 141.77.11.129 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-7 182 181.783946 141.77.183.99 141.77.11.129 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-7 182 181.783946 141.77.183.99 141.77.11.129 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-7 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-7 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-23 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-32 Min-18987152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-32 Min-18987152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC P6 557 - 58328 [ACK] Seq-32 Ack-32 Min-2897152 Len-1 182 181.783940 141.77.183.99 141.77.183.99 TC	9 13.004	4298 141.37.168.39	141.37.11.129	SMTP	56 C: AUTH LOGIN
181 13, 184840 141, 37, 113, 129 141, 37, 181, 39 5917 72 51 334 VOXICombinishing 141, 37, 114, 39 141, 37,	0 13.004	4819 141.37.11.129	141.37.168.39	TCP	60 587 + 53628 [ACK] Seq=235 Ack=31 Win=14608 Len=0
13 17 18 14 15 17 18 18 14 15 17 18 18 18 18 18 18 18	1 13.004	4820 141.37.11.129	141.37.168.39	SMTP	
19 17.85394 141.97.11.129 141.97.168.99 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-23 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-23 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-23 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-23 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-23 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-24 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-24 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-239 Ack-24 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-271 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-281 Linn-14688 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-281 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-281 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 587 - \$1962 [ACC] Seq-241 Ack-281 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 57 - \$1962 [ACC] Seq-241 Ack-281 Linn-2697152 Len-9 (1997) 141.97.11.129 TCP 69 57 -	2 13.045	5566 141.37.168.39	141.37.11.129	TCP	54 53628 → 587 [ACK] Seq=31 Ack=253 Win=2097152 Len=0
128 17.853441 141.57.11.129 141.57.11.129 TCP 61.55262 + SPT PSM, ACC Seq-22 Ack-253 Nin-2097152 Len-0 Len-7 TCP segment of a reassembled PDU				TCP	
120 17.83341 121.71.186.9.9 121.77.11.129 121.77.11.					
121 17.853846 141.77.11.129 141.77.168.39 TCP 69.57 + 35028 ACK) Seq.233 Ack-39 Min-14688 Len-0 122 18.622975 141.77.11.129 141.77.168.39 TCP 69.57 + 35028 ACK) Seq.233 Ack-39 Min-14688 Len-0 123 18.622975 141.77.11.129 141.77.168.39 TCP 69.57 + 35028 ACK) Seq.233 Ack-39 Min-14688 Len-0 124 18.622975 141.77.11.129 141.77.168.39 TCP 69.57 + 35028 ACK) Seq.231 Ack-32 Min-14688 Len-0 125 18.622975 141.77.11.129 141.77.168.39 TCP 54.53628 - 557 (ACK) Seq.241 Ack-271 Min-2097152 Len-1 142 23.1137420 141.77.11.129 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.241 Ack-271 Min-2097152 Len-1 143 23.1137420 141.77.11.129 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.271 Ack-241 Min-2097152 Len-1 142 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.271 Ack-241 Min-2097152 Len-19 TCP Segment of a reassembled POU] 142 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.271 Ack-241 Min-14688 Len-0 142 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.271 Ack-241 Min-14688 Len-0 143 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.272 Ack-271 Min-2097152 Len-19 TCP Segment of a reassembled POU] 144 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.272 Ack-271 Min-14688 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.273 Ack-30 Min-14688 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.273 Ack-30 Min-14689 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK) Seq.274 Ack-371 Min-14689 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK Seq.274 Ack-371 Min-14689 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK Seq.274 Ack-371 Min-14689 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028 ACK Seq.274 Ack-371 Min-14689 Len-0 145 23.1197420 141.77.168.39 TCP 69.57 + 53028					
123 18.082891 141.57.11.129 141.57.11.129 SMTP 56 CF User: cm5/doil 141.57.11.129 141.57.16.139 TCP 66 57 + 55028 [ACK] Seq-253 Ack+41 kin-14608 Lene0 127 18.082844 141.57.11.129 141.57.168.139 SMTP 72 S: 134 UBFC:04/cm26 141.57.11.129 141.57.168.39 141.57.11.129 TCP 54 55028 - 577 [ACK] Seq-41 Ack+27; kin-14608 Lene0 142 23.137388 141.57.11.129 141.57.168.39 TCP 69 567 + 55028 - 587 [ACK] Seq-41 Ack+27; kin-14608 Lene0 142 23.137388 141.57.11.129 141.57.168.39 TCP 69 567 + 55028 Ack-380 kin-14608 Lene0 142 23.137388 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 142 23.137388 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 142 23.137388 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 143 23.137388 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 143 23.137388 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 143 23.139738 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 144 23.13889 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 144 23.13889 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 145 23.139738 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 145 23.13889 141.57.11.129 141.57.168.39 TCP 69 57 + 55028 [ACK] Seq-27] Ack-48 kin-14608 Lene0 145 23.13889 141.57.11.129 141.57.168.39 STCP 69 57 55028 157 [ACK] Seq-28 Ack-58 kin-12097[52 Lene0 141.57.11.129 141.57.168.39 STCP 69 55028 2.57 [ACK] Seq-28 Ack-58 kin-12097[52 Lene0 141.57.11.129 141.57.168.39 STCP 69 55028 2.57 [ACK] Seq-28 Ack-58 kin-12097[52 Lene3 [TCP segment of a reassembled POU] 147 25.772754 141.57.168.39 141.57.11.129 TCP 69 55028 2.57 [ACK] Seq-28 Ack-58 kin-12097[52 Lene3 [TCP segment of a reassembled POU] 147 25.772754 141.57.11.129 141.57.168.39 TCP 69 57 5.5028 2.57 [ACK] Seq-28 Ack-58 ki					
129 18.68295					
127 136.62894 141.77.11.129 141.77.168.39 5MTP 72 5: 334 UBF.c250.cng6 1 141.77.161.39				TCP	
128 18.68333					
144 23.18644 141.37.11.129 141.37.11.129 TC 55 55288 + 587 [Ps], ACC] Seq.41 Ack-271 Min-A007152 Lens [TCP segment of a reassembled POU] 146 23.157368 141.37.11.129 141.37.11.129 TC 75 55288 + 587 [Ps], ACC] Seq.41 Ack-271 Min-A007152 Lens [TCP segment of a reassembled POU] 146 23.157469 141.37.11.129 TC 75 55288 + 587 [Ps], ACC] Seq.42 Ack-271 Min-A007152 Lens [TCP segment of a reassembled POU] 147 23.157469 141.37.11.129 TC 75 55288 + 587 [Ps], ACC] Seq.42 Ack-271 Min-A007152 Lens [TCP segment of a reassembled POU] 148 23.00007 141.37.11.129 TC 75 55288 + 587 [Ps], ACC] Seq.42 Ack-271 Min-A007152 Lens [TCP segment of a reassembled POU] 151 22.5757647 141.37.11.129 TC 75 55288 + 587 [ACC] Seq.42 Ack-270 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 553288 + 587 [ACC] Seq.42 Ack-280 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 553288 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 553288 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 553288 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [Ps], ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [ACC] Seq.40 Ack-300 Min-A007152 Lens [TCP segment of a reassembled POU] 152 25.00007 141.37.11.129 TC 75 55328 + 587 [ACC] Seq.4					
146 23.157388 141.37.11.129 141.37.168.39 TCP 69 587 + 35028 [ACK] Seq.271 Ack-421 Win-14688 Len-9 [TCP segment of a reassembled PDU] 141.37.168.39 141.37.11.129 TCP 69 587 + 35028 [ACK] Seq.271 Ack-621 Win-14688 Len-9 [TCP segment of a reassembled PDU] 142 23.157349 141.37.11.129 141.37.168.39 TCP 69 587 + 35028 [ACK] Seq.271 Ack-63 Win-14668 Len-9 [TCP segment of a reassembled PDU] 142 23.098034 141.37.11.129 141.37.168.39 STTP 69 587 + 35028 [ACK] Seq.271 Ack-63 Win-14668 Len-9 [TCP segment of a reassembled PDU] 151.253.270.30 Authentication successful 151.23.157349 141.37.11.129 TCP 54 55028 - 587 [ACK] Seq.273 Ack-63 Win-1269 TCP Len-9 [TCP segment of a reassembled PDU] 151.253.270.30 Authentication successful 151.23.157349 141.37.11.129 TCP 54 55028 - 587 [ACK] Seq.263 Ack-580 Win-1269 TCP Len-9 [TCP segment of a reassembled PDU] 151.253.270.30 Ack-580 Win-1269 Mack-580 Win-1269 Win-269 TCP segment of a reassembled PDU] 151.253.270 Ack-580 Win-1269 Mack-580 Win-1269 Win-269 Mack-580 Win-1269 Win-269 Mack-580 Win-1269 Win-269					
148, 23, 1574,09					
147 23.157494 141.77.11.129 141.79.168.39 TCP 69 587 = 35028 [ACK] Seq.271 Ack-61 Lin-14608 Len-0 148 23.988057 141.79.11.129 141.79.168.39 TCP 69 587 = 35028 [ACK] Seq.272 Ack-63 Lin-14608 Len-0 149 23.988054 141.79.11.129 141.79.168.39 STTP 69 587 = 35028 [ACK] Seq.273 Ack-63 Lin-14608 Len-0 151 23.957697 141.79.11.129 141.79.11.129 TCP 95 55028 = 587 [ACK] Seq.65 Ack-308 Lin-2897312 Len-0 172 23.958053 141.79.11.129 141.79.168.39 TCP 69 587 = 55028 = 587 [PS], ACK] Seq.65 Ack-508 Lin-2897312 Len-0 172 25.958258 141.79.11.129 141.79.168.39 TCP 69 55 55028 = 587 [PS], ACK] Seq.65 Ack-508 Lin-2897312 Len-0 172 25.958258 141.79.11.129 141.79.168.39 TCP 69 57 = 55028 EACK-508 Lin-2897312 Len-0 172 25.958258 141.79.11.129 141.79.168.39 TCP 69 57 = 55028 EACK-508 Lin-2897312 Len-0 172 29.758275 141.79.11.129 141.79.168.39 TCP 69 57 = 55028 EACK-508 Lin-2897312 Len-0 172 29.758275 141.79.11.129 141.79.11.129 STTP 59 5702 EACK-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 Ack-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 = 7502 [ACK] Seq.508 EACH-508 Lin-2897312 Len-0 172 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 EACK Seq.508 EACH-508 Lin-2897312 Len-0 173 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 EACK Seq.508 EACH-508 Lin-2897312 Len-0 173 29.757557 141.79.11.129 141.79.168.39 STTP 69 577 EACK Seq.508 EACH-508 Lin-2897312 Len-					
148 23.000807 141.77.11.29 141.77.11.129 5 5 (Fass: WWiDMATVMC)FSTg= 148 23.000807 141.77.11.129 141.77.11.129 5 6 6 587 + 3582 6 KM2 Seq. 27 km c					
140 23.988834 141.37.11.120 141.37.183.90 TCF 69 587 = 35028 [ACK] Seq=271.6.ccp3 lint-14688 [ex-en] [141.37.11.120 141.37.11.120 141.37.11.120 141.37.11.120 TCF 95 55028 = 587 [ACK] Seq=69 Ack-380 lint-2897132 Len=0 [141.37.11.120 Lint-14.137.11.120 TCF 95 55028 = 587 [PS], ACK] Seq=69 Ack-380 lint-2897132 Len=0 [17.00 seq=60 Ack-380 lint-2897132 Len=0 [17.00 se					
158 23.98265 141.37.11.129 141.37.168.39 5MTP 91.5: 235 2.7.0 Authentication successful 1512.959670 141.37.168.39 141.37.11.129 TC 95 53526 - 587 [CMS] Seq-69.06.4688 Uin-2097152 Len-0 [TCP segment of a reassembled PDU] 1712 29.108925 141.37.11.129 TC 95 53526 - 587 [CMS] Seq-69.06.4688 Uin-2097152 Len-1 [TCP segment of a reassembled PDU] 172 29.108925 141.37.11.129 TC 95 53526 - 587 [CMS] Seq-69.06.4688 Uin-2097152 Len-1 [TCP segment of a reassembled PDU] 172 29.108925 141.37.11.129 TC 95 53526 - 587 [CMS] Seq-69.06.4688 Uin-2097152 Len-1 [TCP segment of a reassembled PDU] 172 29.108925 141.37.11.129 141.37.11.129 TC 95 5326 - 587 [CMS] Seq-69.06.4688 Uin-2097152 Len-35 [TCP segment of a reassembled PDU] 172 29.702244 141.37.11.129 141.37.11.129 SMTP 56 C. Mall Fabrus (Access Seq. Access Seq. Acces					
151 23.9787674					
171; 23.68953					
172 29.189296 141.37.11.129 141.37.183.99 TCP 68 577 - 53828 [ACK] Seq-388 Ack-69 Min-144888 Lene 9 172 29.189296 141.37.11.129 141.37.183.99 TCP 68 537628 - 53762 PGPA, ACK] Seq-388 Ack-69 Min-14488 Lene 9 172 29.792284 141.37.11.129 141.37.118.39 TCP 68 53762 PGPA, ACK] Seq-388 Ack-69 Min-14468 Lene 9 172 29.792285 141.37.11.129 141.37.118.39 TCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14468 Lene 9 172 29.792757 141.37.11.129 141.37.183.99 TCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 172 29.792758 141.37.11.129 141.37.183.99 TCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 172 29.792759 141.37.11.129 141.37.183.99 TCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 172 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 172 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 172 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 173 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 173 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 173 29.792759 141.37.11.129 141.37.183.99 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 174 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14668 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 Lene 9 175 29.792759 141.37.11.129 141.37.183.90 STCP 68 537 - 53828 [ACK] Seq-388 Ack-69 Min-14688 [ACK] Seq-388 Ack-69 Min-14688 [ACK] Seq-388 Ack-69					
173 25.108294 141.37.11.129 141.37.11.129 TC 85.5528 + ST [PsH, ACK] Sep46 Ack-Sa0 Min-R0097152 Len-95 [TCP segment of a reassembled PDU] 172 25.702284 141.37.11.129 141.37.11.129 STIP 56 C: MRIL FRONT: cyssainhoffsannö9@heb.de> 172 25.772575 141.37.11.129 141.37.16.8.39 TCP 66 57 - 5326 [ACK] Sep380 Ack-193 Min-14680 Lenne 172 25.772575 141.37.11.129 141.37.168.39 STIP 66 57 - 5326 [ACK] Sep380 Ack-191 Min-14680 Lenne 172 25.772575 141.37.11.129 141.37.168.39 STIP 66 57 - 5326 [ACK] Sep380 Ack-191 Min-14680 Lenne 182 25.82253 141.37.16.129 141.37.168.39 STIP 66 57 - 576 [ACK] Sep-180 Ack-192 Min-14680 Lenne					
174 29.198915 141.37.11.129 141.37.168.39 TCP 69 587 + 53628 [ACK] Seq=308 Ack=99 Win-14686 Len=9 177 29.772284 141.37.11.129 141.37.11.129 SVITP 56 C: NAIL FROW: cyssarinboffsanrap@de-t.de> 178 29.772755 141.97.11.129 141.37.168.39 TCP 69 587 + 53628 [ACK] Seq=308 Ack=101 Win-14680 Len=0 179 29.772754 141.97.11.129 141.37.168.39 SVITP 66 S: 250 2.1.0 0k 181 28.282353 141.37.16.139 141.37.16.139 141.37.16.139 141.37.16.139 181 28.282353 141.37.16.139 141.37.16.139 141.37.16.139					
177 29.772284 141.37.11.129 141.37.11.129 STIP 56 C: PAIL FRON: cystasinorfframmöglæb.de 172 29.77275 411.37.11.129 141.37.168.39 TCP 66 587 + 35826 [ACX] 5eq=280 Act-121 Lith-1466B Lenn0 179 29.775547 141.37.11.129 141.37.168.39 STIP 66 5 259 2.1.0 0k 1812 29.82253 31 41.37.16.13 91 141.37.161.120 TCP 54 51262 - 587 [ACX] 5eq=181 Act-322 Lith-2697152 Lenn0					
178 29.772755 141.37.11.129 141.37.168.39 TCP 60 587 + 53628 [ACK] Seq=308 Ack=101 Win=14608 Len=0 179 29.773547 141.97.11.129 141.37.168.39 SMTP 68 5:260 2.1.0 Ok 128 128 29.82233 31 41.37.168.39 141.37.11.129 TCP 54 53628 + 557 [ACK] Seq=101 Ack=322 Win=2697152 Len=0 178 20 18 18 18 18 18 18 18 18 18 18 18 18 18					
179 29,779547 141,77,11.129 141,77,168,39 54TP 68 5: 259 2,1.6 OK 121 29,22233 141,77,168,39 141,77,168,39 54TP 69 55 250 2,1.6 OK 121 29,22233 141,77,168,39 141,77,168,39 141,77,168,39 141,77,168,39 141,77,68,39					
181 29.822533 141.37.168.39 141.37.11.129 TCP 54 53628 → 587 [ACK] Seq=101 Ack=322 Win=2097152 Len=0					
			141.37.6.164	TCP	3-3-3026 4-307 [R.K.] 560-101 (K.5-32 KK] [5-07/132 Len-0 5] [TCP Kep-Alive] 5153 + 445 [KK] [5-07/132 Len-0 5] [TCP Kep-Alive] 5153 + 445 [KK] [5-07/132 Len-0 5] [TCP Kep-Alive] 5153 + 445 [KK] [5-07/132 Len-0 101/132 [TCP KK]
189 31.679355 141.37.61.64 141.37.168.39 TCP 60 [TCP Keep-Alive ACK 445 > 51553 [ACK] 56q-1066 Ack-455 kin-257 (en-0					



- → SMTP Pakete in Klartext
- 6. Schreiben Sie noch eine Mail an ihren Email-Account. Verwenden Sie willkürliche EmailAdressen 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?
- → MAIL-FROM, "from" und "to" können beliebig gewählt werden. Lediglich RCPT-TO muss die richtige Zieladresse sein. Beim Empfänger wird schlussendlich nur das "from" Feld angezeigt, dieses kann man somit auch ganz einfach auf **service@paypal.de** oder Ähnliches setzen und gezielte Spam oder Phishing Mails verschicken.

3.2 SMTP in Python

Implementieren sie einen SMTP-Client in Python. Benutzen Sie aber nicht die in Python vorhandene Mail-Bibliotheken (smtplib) sondern programmieren Sie direkt auf Sockets. Hinweise:

- 1. Verwenden Sie die Python Bibliothek base64, um Login und Passwort in Base64-Code zu konvertieren.
- 2. Alle Daten werden im UTF-8-Format übertragen.
- 3. Alle gesendeten Zeilen müssen mit "\r\n" enden, damit der Mail-Server das Zeilenende erkennt. Auch diese Zeichen müssen "UTF-8"-codiert gesendet werden.
- → Siehe 3_smtp.py Datei.

4 Rechen-Server

Implementieren Sie einen einfachen Rechen-Server in Python. Der Rechen-Server soll in der Lage sein, die Summe, das Produkt, das Minimum und das Maximum von n Zahlen zu bestimmen. Die Kommunikation mit dem Server erfolgt über Sockets. Eine Anfrage hat das folgende Format:

<ID><Rechenoperation><N><z1><z2>...<zN>

ID ist ein unsigned Integer (4 Bytes) und dient als Identifikator der Aufgabe. Rechenoperation ist eine der Zeichenketten "Summe", "Produkt", "Minimum", "Maximum" und wird "UTF-8 kodiert" übertragen. Die Zahl N wird als Unsigned-Char-Wert (1 Byte) übertragen und gibt an, wie viele Zahlen folgen. Die Zahlen z1 bis zN werden als signed Integer (4 Bytes) übertragen. Verwenden Sie die Funktionen pack und unpack aus dem Modul struct, um die Nachrichten zu erzeugen. Sowohl struct.pack als auch .encode() liefern Bytes-Objekte, die einfach konkateniert werden können.

Das Ergebnis der Rechnung soll dem Client in folgendem Format zurückgeliefert werden:

<ID><Ergebnis>

Dabei ist ID definiert wie oben und Ergebnis soll ein signed Integer-Wert sein.

Implementieren Sie die Aufgabe mit Stream- und Datagram Sockets.

→ Siehe 4_rechenserver.py, 4_client_tcp.py, 4_client_udp.py

4.1 Lokale Kommunikation

Testen Sie die Skripte zunächst lokal, indem Sie für Client und Server die IP-Adresse 127.0.0.1 (localhost) verwenden. Starten sie dazu erst das Server-Skript und dann das Client-Skript jeweils in einem Windows-Cmd-Fenster. Nutzen Sie außerdem das Tool CurrPorts (https://www.nirsoft.net/utils/cports.html), um die aktiven Sockets zu überwachen.

Erklären Sie den Zusammenhang von ausgetauschten Paketen und Python-Code, indem Sie

- 1. für jedes gesendete Paket bestimmen, welcher Befehl in welchem Skript (Client/Server) dafür verantwortlich ist, dass das Paket gesendet wird
- 2. für jeden blockierenden Befehl bestimmen, die Ankunft welches Pakets dafür verantwortlich ist, dass die Ausführung des Befehls vervollständigt wird

No.	Time	Source	Destination	Protocol	Length Info
г	45 15.281959	127.0.0.1	127.0.0.1	TCP	56 53676 → 50000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
	46 15.282038	127.0.0.1	127.0.0.1	TCP	56 50000 → 53676 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
	47 15.282085	127.0.0.1	127.0.0.1	TCP	44 53676 → 50000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
	60 19.571068	127.0.0.1	127.0.0.1	TCP	62 53676 → 50000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=18
	61 19.571101	127.0.0.1	127.0.0.1	TCP	44 50000 → 53676 [ACK] Seq=1 Ack=19 Win=2619648 Len=0
	62 19.571385	127.0.0.1	127.0.0.1	TCP	52 50000 → 53676 [PSH, ACK] Seq=1 Ack=19 Win=2619648 Len=8
	63 19.571403	127.0.0.1	127.0.0.1	TCP	44 53676 → 50000 [ACK] Seq=19 Ack=9 Win=2619648 Len=0
	64 19.571811	127.0.0.1	127.0.0.1	TCP	44 53676 → 50000 [FIN, ACK] Seq=19 Ack=9 Win=2619648 Len=0
	65 19.571832	127.0.0.1	127.0.0.1	TCP	44 50000 → 53676 [ACK] Seq=9 Ack=20 Win=2619648 Len=0
	66 19.572333	127.0.0.1	127.0.0.1	TCP	44 50000 → 53676 [FIN, ACK] Seq=9 Ack=20 Win=2619648 Len=0
L	67 19.572367	127.0.0.1	127.0.0.1	TCP	44 53676 → 50000 [ACK] Seg=20 Ack=10 Win=2619648 Len=0

- No.45: CLIENT sock.connect((Server_IP,Server_Port)) sendet Paket
- No.47: CLIENT sock.connect((Server_IP,Server_Port)) beendet
- No.60: CLIENT sock.send(build.msg()) sendet Paket
- No.61: CLIENT sock.send(build.msg()) beendet
- No.62: SERVER sock.send(getAnswer(data)) sendet Paket
- No.63: SERVER sock.send(getAnswer(data)) beendet
- No.64: CLIENT sock.close() sendet Paket
- No.65: CLIENT sock.close() beendet
- No.66: SERVER sock.close() sendet Paket
- No.67: SERVER sock.close() beendet

4.2 Netzwerk-Kommunikation

Gehen Sie mit ihren Rechnern ins VPN. Konfigurieren Sie die IP-Adressen jetzt so, dass die Kommunikation im lokalen Netzwerk (im VPN) möglich ist.

Beantworten Sie die folgenden Fragen durch Experimente und unter Verwendung der Python-Hilfe:

- 1. Wie können Sie im Client Python-Skript die IP-Adresse und Port-Nummer des verwendeten lokalen Sockets bestimmen ("bestimmen" im Sinne von herausfinden)?
 - → ip, port = sock.getsockname()
- 2. Wann (in welcher Code-Zeile) und woher erhält ein Client seine IP-Adresse und Port-Nummer?
 - → sock.connect((Server_IP, Server_PORT)) #vom Server
- 3. Wie können Sie im Client-Skript die IP-Adresse und Port-Nummer des Sockets setzen?
 - → sock.bind((IP,port)) #wenn es festgelegt werden muss
- 4. Warum müssen Sie Timeouts verwenden und wie funktioniert try ... except? Mit welchem Befehl können Sie einen gemeinsamen Timeout für alle Sockets setzen?
 - → timeout: zeit die ein vorgang(z.B. threads) brauchen darf bis er mit einem fehler abgebrochen wird
 - \rightarrow socket.setdefaulttimeout(10)

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

→ nicht möglich

4.3 Unterstützungfür mehrere Clients

Erweitern sie den Server mit Threads, so dass dieser mit mehreren Clients gleichzeitig verbunden sein kann.

→ Siehe rechen_server.py

5 Port Scan

5.1 Beschreibung

In diesem Versuch führen wir einen Port-Scan durch, um herauszufinden, welche Ports auf einem Server geöffnet sind. Wir scannen zum einen die standardisierten Ports von 1-50 nach offenen TCP Ports. Wenn wir einen offenen Port finden, versuchen wir, eine Nachricht an diesen Port zu schicken. Weiterhin vermuten wir, dass auf dem Server ECHO-Dienste für die Übertragung mit TCP und UDP laufen.

Ist ein TCP-Port auf einem Server geöffnet, dann wird ein Verbindungsaufbau auf diesem Port akzeptiert. Ist der TCP-Port nicht offen, so antwortet der Server entweder nicht (Windows FehlerCode 10060) oder mit einem RST+ACK, in dem der Verbindungsaufbau zurückgewiesen wird (Windows Fehler-Code 10061).

Ist ein UDP Port auf einem Server geöffnet, so antwortet der Server entweder mit einer Nachricht oder überhaupt nicht. Die Reaktion hängt sowohl vom empfangenden Dienst als auch von der Nachricht selbst ab. Ist der UDP Port nicht geöffnet, so antwortet der Server entweder nicht oder mit einem ICMP Paket vom Typ 3, mit dem er mitteilt, dass das Ziel nicht erreichbar ist (Windows Fehler-Code 10054).

5.2 Versuch

5.2.1 TCP Port Scanner

Implementieren Sie ein Skript, das eine gegebene Anzahl von TCP-Ports auf einem gegebenen Server scannt und die offenen Ports zurückliefert. Führen Sie das Script für den Labor-Server 141.37.168.26 und Ports zwischen 1 und 50 durch. Zeichnen Sie die Kommunikation mit WireShark auf.

Starten Sie die einzelnen Port-Anfragen als Threads, um den Scan-Vorgang zu beschleunigen. Sie können einfach eine Funktion mit

```
t=Thread(target=<function>,args=(<arg>,))
```

starten. Achten Sie darauf, dass der Thread beendet werden kann. Einfach geht dies mittels eines global Flags Continue, das der Thread kontinuierlich abfragt und sich bei Continue==False beendet.

Hinweis: https://docs.python.org/3.4/library/threading.html

```
→ siehe 5_portscan_tcp.py
```

5.2.2 UDP Port Scanner

Erweitern Sie das Script, um auch UDP Ports zu scannen. Führen Sie das Script für den LaborServer 141.37.168.26 und Ports zwischen 1 und 50 durch. Zeichnen Sie die Kommunikation mit WireShark auf. Unterscheiden Sie Ports, auf denen Sie keine Antwort bekommen und Ports, auf denen Sie Fehlermeldung 10054 erhalten.

```
→ siehe 5_portscan_udp.py
```

5.3 Fragen

- 1. Geben Sie die Liste der offenen TCP und UDP Ports an.
 - → TCP: offene Ports: [7, 9, 13, 17, 19]
 - \rightarrow UDP: offene Ports: [7, 13, 17]
- 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.
- 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.
 - ightarrow Port 7 beim TCP script liefert keine Antwort zurück, da wir keine Nachricht schicken sondern nur eine Verbindung aufbauen
 - → Port 7 beim UDP script liefert den Inhalt der gesendeten Nachricht zurück