

## Lösungsblatt 3

5. Oktober 2020

### Aufgabe 4-1: Privilege Drop in C

Die Wandlung des Namens des Zielusers in numerische UID und GID muss vor dem `chroot` erfolgen, da dannach die Datei `/etc/passwd` in der diese Informationen abgelegt sind nicht mehr sichtbar ist. Deswegen wird der Privilege Drop in eine Vorbereitung `prep_priv_drop()` für diesen Lookup und die tatsächliche Ausführung des Privilege Drop `do_priv_drop()` aufgeteilt. In der Vorbereitung werden ausserdem die Gruppen des laufenden Prozesses schon auf die Gruppenzugehörigkeit des Ziel-Users für den Privilege Drop gesetzt.

Ausgabe der Benutzerinformationen vor und nach dem Privilege drop ergibt das folgende Bild:

```
root#
  getpwuid() returned the following info for your userid:
  pw_name   : root
  pw_uid    : 0
  pw_gid    : 0
  pw_dir    : /root
  pw_shell  : /bin/bash
getpwuid() returned the following info for your userid:
  pw_name   : vagrant
  pw_uid    : 1000
  pw_gid    : 1000
  pw_dir    : /home/vagrant
  pw_shell  : /bin/bash
root#
```

## Aufgabe 4-2: Chroot in C

Wichtig ist es, den `chroot` nach der Vorbereitung und vor dem Privilege Drop auszuführen:

```
1 ...  
2 // Prepare privilege drop  
3 user_s = prep_privilege_drop();  
4  
5 // Do chroot  
6 do_chroot();  
7  
8 // Do privilege drop  
9 // Note: This must always be done before the first external input.  
10 do_privilege_drop(user_s);  
11 ...
```

Zur Prüfung des gesetzten Root Directory können Sie dort eine Datei mit eindeutigem Namen ablegen und das Directory auflisten. Hier ist das Zielverzeichnis des Privilege Drop `/tmp/tst` in das die Datei `/tmp/tst/we_are_in_tmp_tst` gelegt wurde:

```
root# ./do_chroot  
..  
we_are_in_tmp_tst  
.  
root#
```

## Aufgabe 4-3: Privilege Separated Service

Siehe Programmcode der Beispiellösung.