**Malwareanalyse und Reverseengineering: Praktikumsaufgabe 2**

**2.1 Hacken!**

a, Zufallszahlen mit Hilfe der „rand“-Funktion erzeugen: (random.c)

#include <stdio.h>

#include <time.h>

void main()

{

//die aktuelle Zeit in Sekunden seit dem 1.1.1970.

srand(time(0));

int i;

for (i = 0; i < 10; i++)

//Zufallszahlen zwischen 0 und 99 (inklusive)

printf("%d \n", (rand() % 100));

}

b, Eigene C-Funktion namens „rand“ in einer neuen Datei: (unrandom.c)

int rand(){

return 42;

}

Übersetzen eines C-Programms in eine Shared-Library durch:

* gcc -shared -fPIC unrandom.c -o unrandom.so

c, Überladen der Systemfunktion „rand“ durch die selbstgeschriebene Funktion:

Exportiert die Funktionen einer C-Datei in die Shared Libraries und überschreibt bereits vorhandene Funktionen:

* export LD\_PRELOAD=$PWD/unrandom.so

d, Überladen der Funktion „localtime“ in der Systemdatei „date“ durch eine selbstgeschriebene Funktion: „localtime“:

#include <stdio.h>

#include <time.h>

struct tm \*localtime(const time\_t \* foo){

static struct tm test;

test.tm\_sec = 10;

test.tm\_min = 10;

test.tm\_hour = 10;

test.tm\_mday = 10;

test.tm\_mon = 10;

test.tm\_year = 1910;

test.tm\_wday = 1;

test.tm\_yday = 1;

test.tm\_isdst = 0;

return &test;

}

* gcc -shared -fPIC faketime.c -o faketime.so
* export LD\_PRELOAD=$PWD/unrandom.so