Übungen zu Systemnahe Programmierung in C

Abschnitt 10.8: Hands-On (Buffer Overflow)

02.07.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Hands-on: Buffer Overlow – Ablauf

- Programm `print_exam.c` ist durch Passwort geschützt
- Zusammen finden wir ein gravierendes Sicherheitsproblem
- … analysieren
- … und beheben es
- **Bei Bemerkungen oder Fragen:** Nutzt die *Hand heben* Funktionalität aus Zoom oder schreibt im Chat
  - ⇒ Der Moderator gibt euer Mikrofon frei
- Bitte erstellt und verteilt keine Aufzeichnungen!

Hands-on: Buffer Overflow – Ausgangssituation

- Passwortgeschütztes Programm

```
01 # Usage: ./print_exam <password>
02 ./print_exam spic
03 Correct Password
04 Printing exam...
```

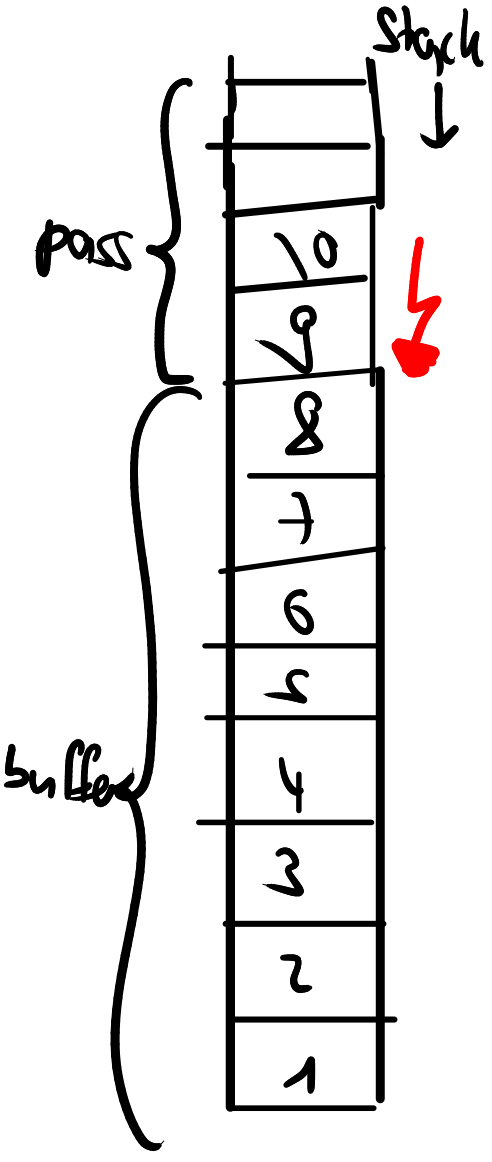
Hands-on: Buffer Overflow – Problemanalyse

- Passwortgeschütztes Programm

```
01 # Usage: ./print_exam <password>
02 ./print_exam spic
03 Correct Password
04 Printing exam...
```

- Ungeprüfte Benutzereingaben ⇒ Buffer Overflow

```
01 long check_password(const char *password) {
02   char buff[8];
03   long pass = 0;
04
05   strcpy(buff, password);
06   if(strcmp(buff, "spic") == 0) {
07     pass = 1;
08   }
09   return pass;
10 }
```



Hands-on: Buffer Overflow – Überlauf verhindern

```
01 long check_password(const char *password) {
02   char buff[8];
03   long pass = 0;
04
05   strcpy(buff, password);
06   if(strcmp(buff, "spic") == 0) {
07     pass = 1;
08   }
09   return pass;
10 }
```

- Mögliche Lösungen
  - Prüfen der Benutzereingabe
  - Dynamische Allokation des Puffers
  - Sichere Bibliotheksfunktionen verwenden ⇒ z.B. `strncpy(3)`