Betriebssysteme und Sicherheit, WS 2025/26

7. Aufgabenblatt – Sicherheit I

Besprechungszeitraum: 09.12. bis 12.12.

Aufgabe 7.1 Schutzziele

(a) Nennen und begründen Sie mögliche Schutzziele, die beim Versenden einer E-Mail relevant sein könnten.

(b) Nennen Sie (mindestens) ein mögliches Schutzziel, das die entsprechende Maßnahme adressieren soll.

1. Passwort
2. Hashwert
3. Checkbox: „AGB gelesen“
4. Redundanz

Aufgabe 7.2 Schwachstellen

Gegeben sei folgender C-Code:

```
#include <stdio.h>
#include <string.h>

void read_something() {
  char foo[4] = "foo";
  char bar[4];
  char baz[4] = "baz";

  printf("Gib eine Zeichenkette ein: ");
  gets(bar);

  printf("Variable foo: %.3s\n", foo);
  printf("Variable bar: %.3s\n", bar);
  printf("Variable baz: %.3s\n", baz);
}

int main() {
  read_something();
  printf("Programm ist fertig.\n");
  return 0;
}
```

Beantworten Sie die folgenden Fragen und begründen Sie Ihre Antwort. Gehen Sie dabei von einem 32-Bit-System aus.

Hinweis: Moderne C-Toolchains haben die unsichere Funktion `gets()` entfernt und aktivieren teils standardmäßig den Stack-Schutz (Canary). Um das gegebene Codebeispiel ohne die Sicherheitsfunktionen zu kompilieren, sollten Sie den folgenden Befehl verwenden und die Warnungen ignorieren.

```
gcc -std=c99 -fno-stack-protector -o program program.c
```

1. Wie sieht der Stackframe der Funktion `read_something` nach den C-Aufrufkonventionen aus? Nehmen Sie dafür an, dass sich die Rücksprungadresse in Adresse `0x1234` befindet.
2. Ist das Programm sicher?
3. Was passiert bei der Eingabe „`bar`“ (ohne Anführungszeichen)?
4. Was passiert bei der Eingabe „`bar_bar`“ (ohne Anführungszeichen)?
5. Was passiert bei der Eingabe „`bar_bar_bar`“ (ohne Anführungszeichen)?
6. Was passiert bei einer noch längeren Eingabe?
7. Wie kann man das Programm absichern?

Aufgabe 7.3 Malware Im Folgenden finden Sie Eigenschaften zur Verbreitung oder Funktionsweise von Malware. Klassifizieren Sie basierend auf dieser Beschreibung die Malware.

(a) Nach der Infektion beginnt die Software, Dateien auf dem Computer des Opfers zu verschlüsseln und verlangt eine Zahlung (normalerweise in Bitcoin) im Austausch für den Schlüssel der Verschlüsselung.

(b) Die Software tarnt sich als wünschenswerter Code oder Software, wie z. B. eine Banking-App oder ein Spiel, um in ein System einzudringen.

(c) Die Software kann sich selbst über ein Netzwerk übertragen, um andere Computer zu infizieren sowie sich selbst zu vervielfältigen.

(d) Die Software infiziert andere Programme und reproduziert sich damit selbst.

Aufgabe 7.4 Passwörter und Hash-Funktionen Nehmen Sie an, dass Sie sich Zugriff zu einer Datenbank mit den Nutzerpasswörtern einer populären Webseite verschafft haben. Sie haben herausgefunden, dass die Passwörter vor dem Abspeichern nach dem folgenden Prinzip gehasht wurden: Zunächst wird jeder Buchstabe in der Eingabe durch seine numerische Position im Alphabet ersetzt. Der Hash-Wert einer Eingabe ist dann die Summe dieser Zahlen modulo 1000.

(a) Hashen Sie die folgenden Wörter wie zuvor beschrieben.

`cause`, `race`, dog, `salt`

Nehmen Sie dafür folgendes an:

- Bei dem Eingabealphabet handelt sich um das lateinische Alphabet mit 26 Zeichen.
- Die Eingabe darf nur aus Kleinbuchstaben bestehen.

(b) Suchen Sie zwei Wortpaare mit dem gleichen Hash-Wert.

(c) Nennen Sie mindestens zwei Gründe, warum dieser Algorithmus nicht für sensible Daten wie Passwörter verwendet werden sollte.

Aufgabe 7.5 Passwortsicherheit Nehmen Sie an, dass bei einer Sicherheitsüberprüfung eines Systems die simulierte Angreifer:in eine Liste mit Passwort-Hashes bekommen hat.

(a) Wie könnte diese Liste verwendet werden, um sich Zugang zum System zu verschaffen?

(b) Wie lange dauert maximal ein Brute-Force Angriff, wenn man 64 Megahashes pro Sekunde berechnen kann und wenn Passwörter 12 Zeichen (nur lateinische Buchstaben) umfassen?

(c) Wie kann man es einer Angreifer:in erschweren, das Passwort zu erraten?