Aufgabe 1: Seitenkanalangriff

Datenschutz und Datensicherheit 20/21

Abgabe: Montag, 26. Oktober 2020, 14 Uhr

In dieser Aufgabe sollen sie einen sogenannten *Seitenkanalangriff* nutzen, um informiert und effektiv ein Passwort zu erraten. Dafür steht Ihnen das Programm `check` zur Verfügung. Sie wissen, dass das Passwort als einziges Kommandozeilenparameter erwartet wird und nur aus Groß- und Kleinbuchstaben (ASCII) besteht. Bei richtigem Passwort liefert der Programmaufruf von `check` den Return Code 0, ansonsten 1 um ein fehlerhaftes Passwort anzuzeigen. Dabei wird folgende Methode zum Überprüfen der Eingabe verwendet:

```
int check_input(char input[]) {
    if (strlen(input) != strlen(password)) {
        return fail();
    }

    for(int i=strlen(password); i>=0; --i) {
        usleep(100000); // brute-force protection

        if(password[i] != input[i]) {
            return fail();
        }
    }

    return success();
}
```

Die Abgabe erfolgt über Moodle und sollte aus einem Archiv bestehen. Darin sollten ggf. der Quelltext Ihrer Lösung für Aufgabe 1.2 und eine Datei mit Ihren restlichen Antworten/Beschreibungen (als `pdf` oder `txt`) enthalten sein. Das Archiv mit dem `check` Programm finden Sie auch nochmal unter: https://wosla.de/dsds/a01.tar.gz

Aufgabe 1.1 (3 Punkte)

Beschreiben Sie das Problem in der Funktion `check_input`, dass den Seitenkanal verursacht den Sie ausnutzen werden. Schätzen Sie die Komplexität Ihres Angriffs (wie viele Zeichenvergleiche sind notwendig) im schlechtesten Falle ein und vergleichen Sie diese mit dem Brute-Force Szenario.

Aufgabe 1.2 (5 Punkte)

Wie lautet das korrekte Passwort? Implementieren Sie dazu den von Ihnen beschriebenen Seitenkanalangriff in einer Programmiersprache Ihrer Wahl. Wenn Sie überhaupt nicht wissen, wo Sie starten sollen, ist die Ermittlung der Passwortlänge ein guter Anfangspunkt.

Aufgabe 1.3 (2 Punkte)

Ändern Sie die Methode `check_input` so ab, dass der von Ihnen beschriebene Angriff nicht mehr möglich ist und es keine Seitenkanäle mehr gibt. Können Sie die gewonnenen Erkenntnisse aus dieser Aufgabe irgendwie verallgemeinern?

Alternativen zu Aufgaben 1.2

Falls Sie die Vorgehensweise aus Aufgabe 1.2 zu einfach finden, oder sich in den Übungen langweilen, weil Sie alles eigentlich schon wissen, was ich erzähle: Es gibt auch andere Wege, an das Passwort zu kommen! Sie können z.B. auch mit einem Debugger das Programm während seiner Ausführung beobachten, oder das Passwort mittels Reverse Engineering ermitteln. Unter Linux könnten Sie sich hierzu mit `gdb` und dessen grundlegender Bedienung auseinandersetzen. Sie können sogar einen Bufferoverflow ausnutzen, um das Programm von sich aus zur Preisgabe des Passwortes zu bringen. Dazu hier ein Auszug, wie dass dann in etwa im Terminal aussieht:

```
./check `python3 -c 'print("A"*? + "\x??\x??\x??\x??")'`
Passwort inkorrekt.
Passwort '???????????' korrekt.
Segmentation fault
```

Oder finden Sie sogar noch eine andere Möglichkeit an das Passwort zu kommen?