Übungsaufgaben zur Vorlesung

IT-Sicherheit Grundlagen

Wolf Müller

Abgabetermin: Montag, 8. Mai 2023, 11:00 Uhr per Moodle
HU Berlin, Sommersemester 2023

Lab 1: Seitenkanalangriff (11 Punkte)

Für ein Programm wurde folgende Routine zur Überprüfung von Passwörtern geschrieben:

```
int compare_key() {
        unsigned int len = strlen(key_input);
        unsigned int correct_len = strlen(correct_key);
        int i, random, punishment = 0, false_key = 0;

        unsigned int seed = getpid() + key_input[0];
        srand(seed);

        if (len != correct_len)
        return 2;

        for (i = correct_len - 1; i >= 0; i--) {
                if (key_input[i] != correct_key[i]) {
                        punishment +=1; // against brute force attack
                        false_key = 1;
                }
        }
        random = (rand() % 5);
        // take that, now you won't be able to do a side-channel attack
        usleep( 800000 * (punishment + random));

        return false_key;
}
```

In dieser Aufgabe sollen Sie einen Seitenkanalangriff nutzen, um den in einem Programm enthaltenen Schlüssel effizient zu brechen. Ihnen steht dabei das Programm `crackme` als „*ELF 64-bit LSB executable*“ zur Verfügung, das die oben angegebene Funktion benutzt. Dieses Programm nimmt einen Kommandozeilenparameter entgegen und prüft, ob es sich dabei um das korrekte Passwort handelt, welches lediglich aus ASCII codierten Groß- und Kleinbuchstaben, sowie Dezimalziffern besteht.

Aufgabe 1.1: Analyse (3 Punkte)

Erklären Sie, über welchen Seitenkanal die oben abgebildete Routine angreifbar ist. Schätzen Sie die Komplexität des Angriffes im Vergleich zu einem reinen Brute-Force Angriff ab.

Aufgabe 1.2: Implementierung (5 Punkte)

Ermitteln Sie das Passwort des Programms. Sie können entweder den zuvor beschriebenen Seitenkanalangriff implementieren. Alternativ (das ist nur etwas für angehende Experten) ist es auch legitim, Methoden des Reverse Engineerings zu verwenden. Sollten Sie sich für dieses Vorgehen entscheiden, beschreiben Sie, wie Sie vorgegangen sind und wie Sie die eingebauten Gegenmaßnahmen umgangen haben.

Aufgabe 1.3: Prävention (3 Punkte)

Ändern Sie die den Quelltext der Funktion (ideal C, notfalls Java oder Pseudocode) `compare_key` ab, so dass keine Seitenkanäle mehr vorhanden sind.

Hinweise

- Die Abgabe erfolgt über Moodle.
- Die Abgabe erfolgt als **Gruppenabgabe**. In den Gruppen G01, . . . , G25 können sich maximal **zwei** Teilnehmer*innen, was auch die angestrebte Gruppengröße ist, zusammenfinden. Bitte informieren Sie mich rechtzeitig, wenn Sie aus triftigen Gründen weitere Gruppen benötigen.
- Ihre Abgabe sollte aus einem tar-Archiv bestehen, welches sowohl ihren Quelltext (Skript- oder Programmiersprache) für Ihre Implementierung des Seitenkanalangriffs (falls sie die Aufgabe nicht per Reverse Engineering gelöst haben), als auch eine Datei mit den Antworten (pdf oder txt) auf die oben genannten Fragen enthält.
- Das Binary von `crackme`[1] finden Sie unter: `https://www2.informatik.hu-berlin.de/sar/Itsec/uebung_ssl/crackme`.
- Gegebenenfalls müssen Sie die Zugriffsrechte von `crackme` anpassen, bevor Sie das Binary ausführen können.
- Das Referenzsystem zur Bearbeitung der Aufgabe ist `gruenau2`.
- Wenn Sie Methoden des Reverse Engineerings nutzen wollen, könnte Software, wie das auf den `gruenau[1-8]`-Rechnern bereits installierte IDA Pro in der Kommandozeilenversion `idal64` oder in der grafischen Version `idaq64`, vielleicht aber auch die von der NSA genutzte Open Source Software Ghidra oder radare2 mit zugehöriger Anleitung nützlich sein.

[1] `SHA256(crackme)=5492b33b1a54370b4ba12899cc71ebace51b407cc21a531c6931d64cde1675f8`