Lehrstuhl für Systemsicherheit
Prof. Dr. Thorsten Holz

Vorlesung Betriebssystemsicherheit
Wintersemester 2016/2017
Ruhr-Universität Bochum

2. Übung

21. November 2016
Abgabe der Hausaufgaben: per Moodle bis zum 28. November 2016, 16:00 Uhr
Bei Fragen und Problemen können Sie sich per E-Mail/Moodle an uns wenden.

Aufgaben

In diser Übung sollen Sie sich mit den Grundlagen von Sicherheit unter Linux vertraut machen. Als Arbeitsumgebung stellen wir eine virtuelle Maschine (VM) zur Verfügung und Sie können die Aufgaben in der VM lösen. Den Download-Link erhalten Sie innerhalb von Moodle, die Einrichtung der VM müssen Sie selbständig durchführen. Eventuell benötigte Quelldateien zum Lösen der Aufgaben finden Sie im Home-Verzeichnis in der VM bzw. in Moodle. Bitte geben Sie Ihre Lösungen per Moodle ab, die korrigierten Aufgaben erhalten Sie ebenfalls dort.

Aufgabe 1: Race Conditions (1 + 2 + 2 Punkt)

1. Erläutern Sie kurz in zwei bis drei Sätzen die Problematik von Race Conditions und beschreiben Sie ein mögliches Angriffsszenario.

2. Analysieren Sie das Programm `toctou` und beschreiben Sie die im Programm enthaltene Schwachstelle. Wie kann ein Angreifer diese Schwachstelle ausnutzen und sich einen *root* Zugriff zum System verschaffen? Geben Sie den kommentierten Quellcode ihrer Lösung inklusive einer kurzen Beschreibung in Moodle ab.

   Hinweis: Der Quelltext des Programms ist in der Datei `toctou.c` verfügbar. Eventuell können Sie symbolische Links zur Lösung der Aufgabe verwenden.

3. Wie kann die Schwachstelle verhindert werden? Beschreiben Sie eine Möglichkeit, wie der Code aus dem vorherigen Aufgabenteil abgeändert werden kann. Ihre Lösung kann auch spezifisch für Linux sein. Geben Sie den kommentierten Quellcode Ihrer Lösung ab.

Aufgabe 2: LD_PRELOAD (1 + 2 + 2 Punkte)

1. Erläutern Sie kurz in zwei bis drei Sätzen die Problematik, die sich aus der Umgebungsvariablen LD_PRELOAD ergibt und beschreiben Sie ein mögliches Angriffsszenario.

2. Erstellen Sie eine Shared Object Library (`.so` Datei), um die Funktion `readdir()/readdir64()` so zu manipulieren, dass alle Dateien, die das Wort *malware* enthalten, versteckt werden. Testen Sie Ihre Library mit dem Befehl `ls`, Dateien mit dem Namen *malware* sollen nicht mehr angezeigt werden. Geben Sie den kommentierten Quellcode in Moodle ab.

3. Erstellen Sie eine Shared Object Library (`.so` Datei), um die Funktion `gethostbyname()` so zu manipulieren, dass die Adresse `rub.de` immer als `127.0.0.1` aufgelöst wird. Alle anderen Hostnamen müssen weiterhin korrekt aufgelöst werden. Sie sollten das Programm `resolve.c` aus den Übungsmaterialien verwenden, um ihre Library zu testen. Geben Sie den kommentierten Quellcode in Moodle ab.

Hinweis: Informationen zur Linux-API finden sie unter `http://linux.die.net/man/`. Eine Beispiel-Implementierung für `malloc()` finden Sie in der Datei `tracemalloc.c` in der VM. Sie können diese Datei mit den folgenden Befehlen kompilieren und laden:

```
$ gcc -shared -fPIC -ldl -o tracemalloc.so tracemalloc.c
$ export LD_PRELOAD=./tracemalloc.so
```

Aufgabe 3: Passwort Cracking (optional, keine Punkte)

In der virtuellen Maschine existieren verschiedene Benutzer, eine shadow-Datei ist auch per Moodle verfügbar. Versuchen Sie mittels John the Ripper (`http://www.openwall.com/john/`) oder einem ähnlichen Tool, so viele Passwörter wie möglich zu knacken. Geben Sie per Moodle eine Liste mit Benutzernamen sowie dem dazugehörigen Passwort ab.

Hinweis: Innerhalb von weniger als zwei Tagen sollten Sie auf einem modernen Rechner etwa 10 Passwörtern herausfinden können.