**System Security**
Übungsblatt 3

# Hackerpraktikum

FAU Erlangen-Nürnberg, Lehrstuhl für Informatik 1

### Privilege Escalation (Aufgabe 1)

Auf dem System `ssh://10.0.23.31` befindet sich unter `/usr/local/bin/` das Programm `ping`, das eine Schwachstelle enthält. Der Quellcode befindet sich im Verzeichnis `/usr/local/src/`.

1. Verschaffen Sie sich `root`-Rechte auf dem System. (2 P.)
2. Erstellen Sie einen Patch mittels `diff(1)`, der die Sicherheitslücke schließt ohne dabei an Funktionalität zu verlieren. (1 P.)

3 P.

### Sandbox (Aufgabe 2)

Schreiben Sie eine `LD_PRELOAD` basierte Sandbox, die es Ihnen erlaubt ein unbekanntes Programm in einer (vermeintlich) sichereren Umgebung zu starten, indem es Bibliotheks-Aufrufe wie `read` und `write` überschreibt. Ihre Sandbox soll verhindern, dass kritische Systemdateien, ausgelesen bzw. manipuliert werden. Die Konfiguration der Sandbox soll dabei (dynamisch) über White- bzw. Blacklists erfolgen. (3 P.).
Eine `LD_PRELOAD` basierte Sandbox ist unsicher, da man relativ einfach aus ihr ausbrechen kann. Nennen Sie zwei Möglichkeiten, um dies zu erreichen. (1 P.) 4 P.

### Backdoor (Aufgabe 3)

In dieser Aufgabe erstellen Sie eine User-Mode Backdoor, die folgende Anforderungen erfüllt:

1. Erlaubt den entfernten Zugriff mit `root`-Rechten.
2. Kommuniziert über das ICMP-Protokoll.
3. Ist so konzipiert, dass Sie nur von Ihnen verwendet werden kann.
4. Ist auch nach einem Neustart noch funktionsfähig.
5. Verbindet sich zum Angreifer über eine Reverse Shell.

Ferner entwickeln Sie ein Programm/Script, das die Kommunikation mit der Backdoor regelt und bei Ausführung eine `root`-Shell mit dem infiltrierten System öffnet.
*Verzichten Sie auf die Installation zusätzlicher Bibliotheken auf dem Targetsystem.* 4 P.

**Rootkit (Aufgabe 4)**
Schreiben Sie ein Kernel-Mode Rootkit, das Dateien, Prozesse und Netzwerkverbindungen eines konfigurierbaren Programms (z.B. Ihrer Backdoor) versteckt. Das Rootkit soll einen Neustart überleben und sich sowie das Nutzprogramm durch entsprechende Konfiguration vor folgenden User-Mode-Programmen verstecken:

`lsmod(8)` (1 P.) `ls(1)` (2 P.) `ps(1)` (2 P.) `lastlog(8)` (2 P.) `netstat(1)` (2 P.)

Die Konfiguration erfolgt dabei im zugehörigen `Makefile` über LKM-Parameter, die beim Laden des Moduls definiert werden. Alternativ können Sie die Konfiguration auch dynamisch mit Hilfe eines Kommunikationsprogramms umsetzen.

**Hinweis:** Implementieren Sie ihr Rootkit als LKM und nutzen Sie Standard-Methoden wie das Hooken der Syscall-Tabelle und das Löschen der Module/Prozesse aus den Kernel-Strukturen. Sie können `strace(1)` verwenden, um herauszufinden, welche Systemaufrufe ein bestimmtes Program nutzt.
**Hinweis:** Das alleinige Hooken des `write(2)` System Calls in Verbindung mit Pattern Matching ist auf Grund damit einhergehender Performanceeinbußen zur Lösung der Aufgabe unzureichend. 9 P.

Folgende Bedingungen gelten für Ihre Lösungen:

- Aufgaben 1 und 3 müssen auf `10.0.23.31` lauffähig sein.
- Die Aufgaben 2 und 4 müssen *zumindest* unter Ubuntu 16.04 LTS lauffähig sein.
- Wenn nötig enthalten Ihre Lösungen ein `Makefile`, das ihren Source Code kompiliert und etwaige Abhängigkeiten auflöst.
- Dokumentieren Sie die nötigen Schritte zur Installation Ihrer Programme.

$3 + 4 + 4 + 9 =$ **20 Punkte**