**Network Security**
Übungsblatt 2

# Hackerpraktikum

FAU Erlangen-Nürnberg, Lehrstuhl für Informatik 1

### Service Fingerprinting (Aufgabe 1)

Im Folgenden entwickeln Sie Programme um HTTP und FTP Dienste *automatisch* zu erkennen. Beachten Sie, dass die von Ihnen erstellten Tools ohne die Verwendung von `nmap` auskommen müssen. Entwickeln Sie dazu zwei Programme, die jeweils die folgenden Aufgabenstellungen erfüllen. Jedes Programm sollte als ersten und einzigen Parameter die IP-Adresse des Zielsystems übergeben bekommen.

1. Auf Host `10.0.23.15` laufen 5 verschiedene HTTP-Daemons auf 5 verschiedenen Ports. Um welche Ports und Daemons (wenn möglich mit Versionsnummer) handelt es sich? (2 P.)

2. Auf Host `10.0.23.15` laufen zusätzlich 4 FTP Server.
   - Ports: 210, 2100, 2121, 21000
   - Daemons: `vs-ftpd`, `pro-ftpd`, `py-ftpd`, `pure-ftpd`

   Welcher Daemon lauscht hinter welchem Port? Untersuchen Sie die genannten FTP-Daemons dazu nach Eigenheiten ihrer Implementierung. (2 P.)

4 P.

### Sniffing (Aufgabe 2)

Auf Host `10.0.23.19` läuft ein HTTP-Server für den ein SSL-verschlüsselter Netzwerkmitschnitt existiert. Für die Implementierung der SSL-Bibliothek des Servers ist eine kritische Sicherheitslücke aufgetaucht.

- Identifizieren Sie die Sicherheitslücke und nutzen Sie diese aus um an den privaten Schlüssel des Servers zu gelangen.
- Entschlüsseln Sie mit dem Schlüssel den Mitschnitt des Netzwerkverkehrs.
- Suchen Sie im Netzwerkverkehr nach einer Benutzername-Passwort-Kombination für die Website `https://10.0.23.19/squirrelmail`.
- Loggen Sie sich auf `https://10.0.23.19/squirrelmail` ein und untersuchen Sie welche Emails verschickt bzw. empfangen wurden.
- Mit welcher Einstellung hätte verhindert werden können, dass aufgezeichnete SSL-Verbindungen beim Auftauchen einer Sicherheitslücke nachträglich entschlüsselt werden können?

**Hinweis:** Auf `https://faui1-hpa.cs.fau.de/material/heartbleed.pcap` finden Sie den Mitschnitt der SSL-Verbindung. Zur Entschlüsselung des Mitschnitts können Sie Wireshark (Achtung! Version $\geq$ 2.2.0) verwenden. Zur Ausnutzung der Sicherheitslücke auf dem Server schreiben Sie bitte ihr eigenes Programm – auch wenn es sich um eine bekannte Schwachstelle handeln sollte, zur der bereits Exploits existieren. 5 P.

### Spoofing (Aufgabe 3)

Ihr Opfer ist Systemadministrator und hat ein Aktualisierungsskript für sein System erstellt (`update.sh`).

1. Bauen Sie ein eigenes Debian-Paket. Nach dem Vorbild von `dnsspoof` soll ein Programm namens `debianspoof` erstellt werden, sodass bei der nächsten Ausführung des Update-Skriptes im Netzwerk ihr Paket installiert wird. Ihr Opfer sollte während seiner Arbeit den Angriff nicht bemerken. (3 P.)
2. Das Opfer benutzt oft `wireshark` als `root`-Benutzer, um Netzwerkprobleme zu beheben. Nutzen Sie dieses Wissen aus, um die Datei `/root/hacked.txt` zu erstellen. (2 P.)
3. Wie könnte der Angriff verhindert werden? Nennen Sie zwei Beispiele. (1 P.)

**Hinweis:** Das Update-Skript finden Sie im StudOn. 6 P.

### Denial of Service (Aufgabe 4)

Schreiben Sie ein Programm, mit dem man messen kann, wie viele SSL/TLS-Handshakes ein Server maximal pro Zeiteinheit bewältigen kann. Testen Sie Ihr Programm gegen `https://10.0.23.14/`, aber *nicht* gegen Server außerhalb des VPNs.

**Hinweis:** Beachten Sie, dass das Ziel der Aufgabe ein *Denial of Service* ist, so dass sich keine weiteren Clients mehr verbinden können. Benutzen Sie dazu nicht `THC-SSL-DOS`. Arbeiten Sie mit einem steigenden Parallelisierungsgrad und stellen Sie in jedem Thread erst eine Verbindung her, bevor Sie die Handshakes parallel starten. Das Ziel der Aufgabe ist es nicht, den Systemload auf dem Webinterface in die Höhe zu treiben; dieser Wert dient Ihnen lediglich als Orientierung. 5 P.

$4 + 5 + 6 + 5 =$ **20 Punkte**