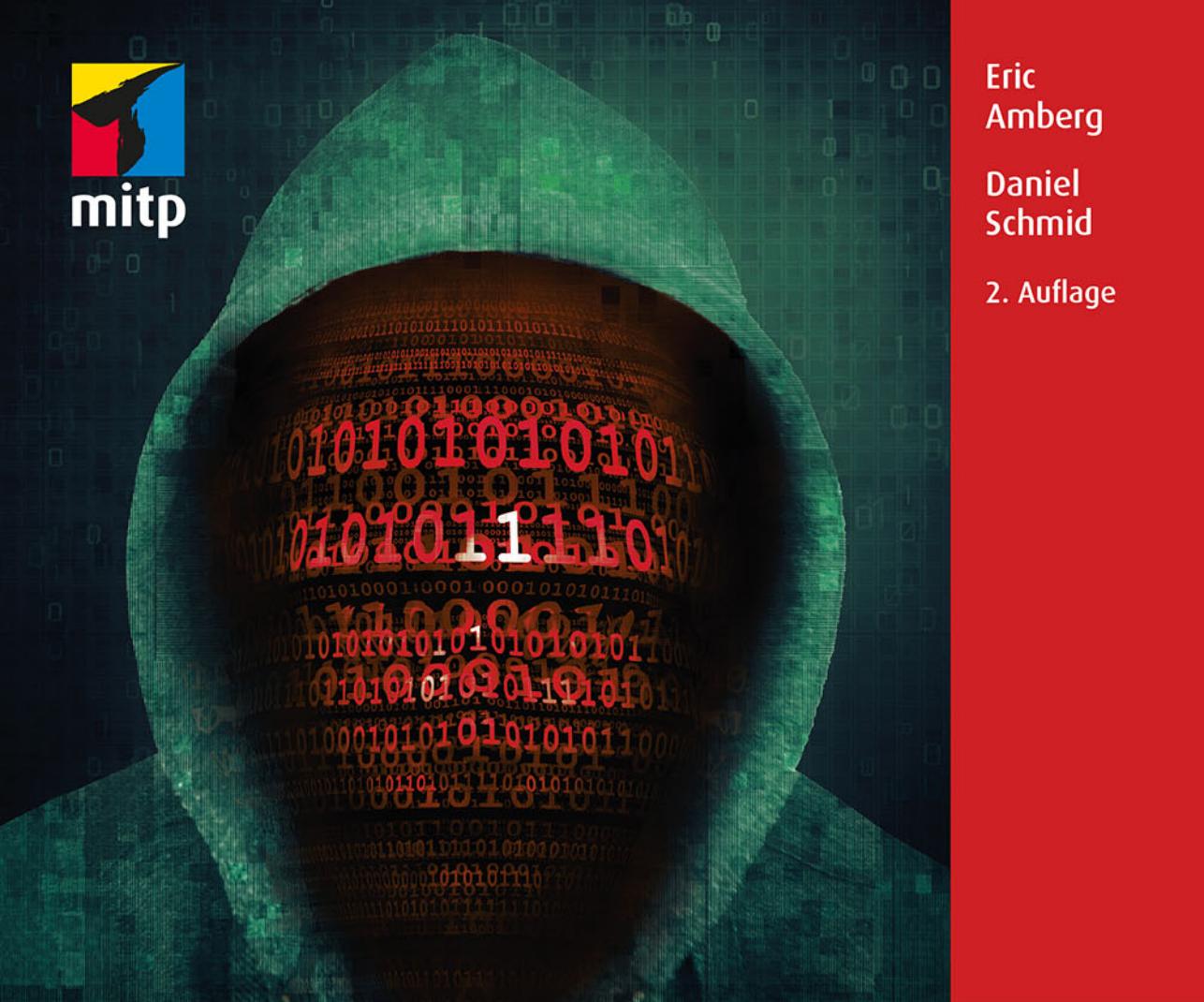




Eric  
Amberg  
Daniel  
Schmid  
2. Auflage



# Hacking

Der umfassende Praxis-Guide

Inkl. Prüfungsvorbereitung zum CEHv11

# Inhaltsverzeichnis

<b>Einleitung</b> .....	29
<b>Über die Autoren</b> .....	35
<b>Danksagung</b> .....	36
<b>Teil I Grundlagen und Arbeitsumgebung</b> .....	37
<b>1 Grundlagen Hacking und Penetration Testing</b> .....	41
1.1 Was ist Hacking? .....	42
1.2 Die verschiedenen Hacker-Typen .....	43
1.3 Motive und Absichten eines Hackers .....	45
1.3.1 Das Motiv .....	45
1.3.2 Ziel des Angriffs .....	46
1.4 Ethical Hacking .....	47
1.5 Der Certified Ethical Hacker (CEHv11) .....	49
1.5.1 Was steckt dahinter? .....	49
1.5.2 Die CEHv11-Prüfung im Detail .....	50
1.6 Die Schutzziele: Was wird angegriffen? .....	51
1.6.1 Vertraulichkeit .....	51
1.6.2 Integrität .....	53
1.6.3 Verfügbarkeit .....	55
1.6.4 Authentizität und Nicht-Abstreitbarkeit .....	56
1.6.5 Die Quadratur des Kreises .....	56
1.7 Systematischer Ablauf eines Hacking-Angriffs .....	58
1.7.1 Phasen eines echten Angriffs .....	58
1.7.2 Unterschied zum Penetration Testing .....	60
1.8 Praktische Hacking-Beispiele .....	62
1.8.1 Angriff auf den Deutschen Bundestag .....	62
1.8.2 Stuxnet – der genialste Wurm aller Zeiten .....	63
1.8.3 Angriff auf heise.de mittels Emotet .....	63
1.9 Zusammenfassung und Prüfungstipps .....	64
1.9.1 Zusammenfassung und Weiterführendes .....	64
1.9.2 CEH-Prüfungstipps .....	64
1.9.3 Fragen zur CEH-Prüfungsvorbereitung .....	65
<b>2 Die Arbeitsumgebung einrichten</b> .....	67
2.1 Virtualisierungssoftware .....	68
2.1.1 Software-Alternativen .....	69
2.1.2 Bereitstellung von VirtualBox .....	70

## Inhaltsverzeichnis

2.2	Die Laborumgebung in der Übersicht . . . . .	71
2.3	Kali Linux . . . . .	72
2.3.1	Einführung . . . . .	72
2.3.2	Download von Kali Linux als ISO-Image . . . . .	73
2.3.3	Kali Linux als VirtualBox-Installation . . . . .	74
2.3.4	Kali Linux optimieren . . . . .	79
2.4	Windows 10 als Hacking-Plattform . . . . .	83
2.4.1	Download von Windows 10 . . . . .	83
2.4.2	Windows-10-Installation in VirtualBox . . . . .	84
2.4.3	Windows 10 – Spyware inklusive . . . . .	85
2.4.4	Gasterweiterungen installieren . . . . .	85
2.5	Übungsumgebung und Zielscheiben einrichten . . . . .	86
2.5.1	Metasploitable . . . . .	87
2.5.2	Die Netzwerkumgebung in VirtualBox anpassen . . . . .	90
2.5.3	Multifunktionsserver unter Linux . . . . .	92
2.5.4	Windows XP und andere Betriebssysteme . . . . .	93
2.5.5	Eine Windows-Netzwerkumgebung aufbauen . . . . .	93
2.6	Zusammenfassung und Weiterführendes . . . . .	94
<b>3</b>	<b>Einführung in Kali Linux . . . . .</b>	<b>95</b>
3.1	Ein erster Rundgang . . . . .	95
3.1.1	Überblick über den Desktop . . . . .	96
3.1.2	Das Startmenü . . . . .	99
3.1.3	Der Dateimanager . . . . .	101
3.1.4	Systemeinstellungen und -Tools . . . . .	103
3.2	Workshop: Die wichtigsten Linux-Befehle . . . . .	104
3.2.1	Orientierung und Benutzerwechsel . . . . .	105
3.2.2	Von Skripts und Dateiberechtigungen . . . . .	107
3.2.3	Arbeiten mit Root-Rechten . . . . .	109
3.2.4	Das Dateisystem und die Pfade . . . . .	112
3.2.5	Dateien und Verzeichnisse erstellen, kopieren, löschen etc. . . . .	113
3.2.6	Dateien anzeigen . . . . .	114
3.2.7	Dateien finden und durchsuchen . . . . .	115
3.2.8	Die Man-Pages: Hilfe zur Selbsthilfe . . . . .	118
3.2.9	Dienste starten und überprüfen . . . . .	119
3.3	Die Netzwerk-Konfiguration anzeigen und anpassen . . . . .	121
3.4	Software-Installation und -Update . . . . .	124
3.4.1	Die Paketlisten aktualisieren . . . . .	124
3.4.2	Installation von Software-Paketen . . . . .	125
3.4.3	Software suchen . . . . .	126
3.4.4	Entfernen von Software-Paketen . . . . .	126
3.5	Zusammenfassung und Prüfungstipps . . . . .	127
3.5.1	Zusammenfassung und Weiterführendes . . . . .	127
3.5.2	CEH-Prüfungstipps . . . . .	127
3.5.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	127

<b>4</b>	<b>Anonym bleiben und sicher kommunizieren . . . . .</b>	129
4.1	Von Brotkrumen und Leuchtpuren . . . . .	129
4.2	Proxy-Server – schon mal ein Anfang . . . . .	131
4.2.1	Grundlagen – so arbeiten Proxys . . . . .	131
4.2.2	Einen Proxy-Server nutzen . . . . .	132
4.2.3	Öffentliche Proxys in der Praxis . . . . .	133
4.2.4	Vor- und Nachteile von Proxy-Servern . . . . .	135
4.2.5	Proxy-Verwaltung mit FoxyProxy . . . . .	136
4.3	VPN, SSH und Socks – so bleiben Black Hats anonym . . . . .	136
4.3.1	Virtual Private Networks (VPN) . . . . .	137
4.3.2	SSH-Tunnel . . . . .	139
4.3.3	SOCKS-Proxy . . . . .	141
4.3.4	Kaskadierung für höchste Anonymität und Vertraulichkeit . . . . .	145
4.3.5	Proxifier – Für unwillige Programme . . . . .	146
4.4	Deep Web und Darknet – im Untergrund unterwegs . . . . .	146
4.4.1	Wo geht es bitte zum Untergrund? . . . . .	146
4.4.2	Das Tor-Netzwerk . . . . .	148
4.4.3	Das Freenet Project . . . . .	153
4.4.4	Die Linux-Distribution Tails . . . . .	154
4.5	Anonym mobil unterwegs . . . . .	156
4.5.1	Mobile Proxy-Tools und Anonymizer . . . . .	156
4.6	Sonstige Sicherheitsmaßnahmen . . . . .	157
4.6.1	System säubern mit dem CCleaner . . . . .	158
4.6.2	G-Zapper: Cookies unter Kontrolle . . . . .	159
4.7	Zusammenfassung und Prüfungstipps . . . . .	159
4.7.1	Zusammenfassung und Weiterführendes . . . . .	159
4.7.2	CEH-Prüfungstipps . . . . .	160
4.7.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	161
<b>5</b>	<b>Kryptografie und ihre Schwachstellen . . . . .</b>	163
5.1	Einführung in die Krypto-Algorithmen . . . . .	164
5.1.1	Alice und Bob ... und Mallory . . . . .	164
5.1.2	Algorithmen und Schlüssel . . . . .	165
5.1.3	Das CrypTool – Kryptografie praktisch erfahren . . . . .	166
5.2	Die symmetrische Verschlüsselung . . . . .	167
5.2.1	Grundlagen der symmetrischen Verfahren . . . . .	167
5.2.2	Verschlüsselung im alten Rom: Die Cäsar-Chiffre . . . . .	168
5.2.3	Strom- und Blockchiffre . . . . .	168
5.2.4	Vor- und Nachteile von symmetrischen Algorithmen . . . . .	169
5.2.5	Wichtige symmetrische Algorithmen . . . . .	169
5.2.6	Symmetrische Verschlüsselung in der Praxis . . . . .	172
5.3	Die asymmetrische Verschlüsselung . . . . .	175
5.3.1	Wo liegt das Problem? . . . . .	175
5.3.2	Der private und der öffentliche Schlüssel . . . . .	175
5.3.3	Der Schlüsselaustausch . . . . .	176

## Inhaltsverzeichnis

5.3.4	Authentizitätsprüfung . . . . .	178
5.3.5	Wichtige asymmetrische Algorithmen . . . . .	179
5.4	Hash-Algorithmen . . . . .	181
5.4.1	Ein digitaler Fingerabdruck . . . . .	181
5.4.2	Integritätsprüfung mit Hashwerten . . . . .	182
5.4.3	Wichtige Hash-Algorithmen . . . . .	185
5.5	Digitale Signaturen . . . . .	188
5.5.1	Das Prinzip der digitalen Signatur . . . . .	188
5.5.2	Wichtige Verfahren der digitalen Signatur . . . . .	189
5.6	Public-Key-Infrastrukturen (PKI) . . . . .	190
5.6.1	Das Prinzip von PKI . . . . .	190
5.6.2	Digitale Zertifikate . . . . .	191
5.6.3	Zertifikate und PKI in der Praxis . . . . .	192
5.6.4	Zertifikatssperrlisten und OCSP . . . . .	195
5.7	Virtual Private Networks (VPN) . . . . .	197
5.7.1	IPsec-VPNs . . . . .	198
5.7.2	SSL-VPNs . . . . .	200
5.8	Angriffe auf kryptografische Systeme . . . . .	201
5.8.1	Methodologie der Kryptoanalyse . . . . .	201
5.8.2	Der Heartbleed-Angriff . . . . .	204
5.8.3	Des Poodles Kern – der Poodle-Angriff . . . . .	205
5.9	Kryptotrojaner und Ransomware . . . . .	206
5.9.1	WannaCry . . . . .	206
5.9.2	Petya . . . . .	207
5.9.3	Locky . . . . .	208
5.9.4	Schutz- und Gegenmaßnahmen . . . . .	208
5.10	Zusammenfassung und Prüfungstipps . . . . .	209
5.10.1	Zusammenfassung und Weiterführendes . . . . .	209
5.10.2	CEH-Prüfungstipps . . . . .	209
5.10.3	Frage zur CEH-Prüfungsvorbereitung . . . . .	209
<b>Teil II</b>	<b>Informationsbeschaffung . . . . .</b>	<b>213</b>
6	<b>Informationsbeschaffung – Footprinting &amp; Reconnaissance . . . . .</b>	<b>217</b>
6.1	Ich will hacken, wozu die langweilige Informationssuche? . . . . .	218
6.1.1	Worum geht es bei der Informationsbeschaffung? . . . . .	219
6.1.2	Welche Informationen sind relevant? . . . . .	219
6.2	Suchmaschinen und Informationsportale nutzen . . . . .	221
6.2.1	Reguläre Suchmaschinen . . . . .	221
6.2.2	Netcraft: Nach öffentlichen und zugriffsbeschränkten Seiten suchen . . . . .	222
6.2.3	WayBack Machine – das Internet-Archiv . . . . .	223
6.2.4	Shodan . . . . .	224
6.2.5	Map-Anbieter: Mal von oben betrachtet . . . . .	225
6.2.6	Personen-Suchmaschinen . . . . .	226

6.2.7	Jobsuchmaschinen als Informationsquelle . . . . .	226
6.2.8	Arbeitgeber-Bewertungsportale . . . . .	227
6.3	Google-Hacking . . . . .	227
6.3.1	Was steckt dahinter? . . . . .	227
6.3.2	Wichtige Suchoperatoren . . . . .	228
6.3.3	Die Google Hacking Database (GHDB) . . . . .	228
6.4	Social-Media-Footprinting . . . . .	229
6.4.1	Wo suchen wir? . . . . .	230
6.4.2	Was suchen wir? . . . . .	230
6.4.3	Wie suchen wir? . . . . .	230
6.5	Technische Analysen . . . . .	231
6.5.1	Whois . . . . .	231
6.5.2	DNS – Das Domain Name System . . . . .	233
6.5.3	E-Mail-Footprinting . . . . .	237
6.5.4	Website-Footprinting . . . . .	239
6.5.5	Dokumente analysieren mit Metagoofil . . . . .	240
6.6	Recon-ng – das Web-Reconnaissance-Framework . . . . .	241
6.6.1	Die ersten Schritte mit Recon-ng . . . . .	241
6.6.2	Ein Modul installieren und laden . . . . .	243
6.6.3	Wie geht es weiter? . . . . .	245
6.7	Maltego – Zusammenhänge visualisieren . . . . .	245
6.7.1	Einführung in Maltego . . . . .	245
6.7.2	Maltego starten . . . . .	246
6.7.3	Mit Maltego arbeiten . . . . .	247
6.7.4	Der Transform Hub . . . . .	250
6.8	Gegenmaßnahmen gegen Footprinting . . . . .	251
6.9	Zusammenfassung und Prüfungstipps . . . . .	251
6.9.1	Zusammenfassung und Weiterführendes . . . . .	251
6.9.2	CEH-Prüfungstipps . . . . .	252
6.9.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	252
7	<b>Scanning – das Netzwerk unter der Lupe . . . . .</b>	255
7.1	Scanning – Überblick und Methoden . . . . .	255
7.1.1	Die Scanning-Phase . . . . .	256
7.1.2	Ziel des Scanning-Prozesses . . . . .	256
7.1.3	Scanning-Methoden . . . . .	256
7.2	TCP/IP-Essentials . . . . .	257
7.2.1	Das OSI-Netzwerk-Referenzmodell . . . . .	257
7.2.2	ARP, Switch & Co. – Layer-2-Technologien . . . . .	259
7.2.3	Das Internet Protocol (IPv4) . . . . .	259
7.2.4	Das Internet Control Message Protocol (ICMP) . . . . .	260
7.2.5	Das User Datagram Protocol (UDP) . . . . .	261
7.2.6	Das Transmission Control Protocol (TCP) . . . . .	262
7.3	Nmap – DER Portscanner . . . . .	263
7.3.1	Host Discovery . . . . .	264

7.3.2	Normale Portscans . . . . .	267
7.3.3	Zu scannende Ports festlegen . . . . .	269
7.3.4	Besondere Portscans . . . . .	270
7.3.5	Dienst- und Versionserkennung . . . . .	271
7.3.6	Betriebssystem-Erkennung . . . . .	272
7.3.7	Firewall/IDS-Vermeidung (Evasion) . . . . .	273
7.3.8	Ausgabe-Optionen . . . . .	274
7.3.9	Die Nmap Scripting Engine (NSE) . . . . .	275
7.3.10	Weitere wichtige Optionen . . . . .	276
7.3.11	Zenmap . . . . .	277
7.4	Scannen mit Metasploit . . . . .	277
7.4.1	Was ist Metasploit? . . . . .	277
7.4.2	Erste Schritte mit Metasploit (MSF) . . . . .	278
7.4.3	Nmap in Metasploit nutzen . . . . .	281
7.5	Weitere Tools und Verfahren . . . . .	283
7.5.1	Paketerstellung und Scanning mit hping3 . . . . .	283
7.5.2	Weitere Packet-Crafting-Tools . . . . .	285
7.5.3	Banner Grabbing mit Telnet und Netcat . . . . .	285
7.5.4	Scannen von IPv6-Netzwerken . . . . .	287
7.6	Gegenmaßnahmen gegen Portscanning und Banner Grabbing . . . . .	288
7.7	Zusammenfassung und Prüfungstipps . . . . .	289
7.7.1	Zusammenfassung und Weiterführendes . . . . .	289
7.7.2	CEH-Prüfungstipps . . . . .	290
7.7.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	290
8	<b>Enumeration – welche Ressourcen sind verfügbar?</b> . . . . .	293
8.1	Was wollen wir mit Enumeration erreichen? . . . . .	293
8.2	NetBIOS- und SMB-Enumeration . . . . .	294
8.2.1	Die Protokolle NetBIOS und SMB . . . . .	294
8.2.2	Der Enumeration-Prozess . . . . .	296
8.3	SNMP-Enumeration . . . . .	301
8.3.1	SNMP-Grundlagen . . . . .	302
8.3.2	SNMP-Agents identifizieren . . . . .	304
8.3.3	Enumeration-Tools nutzen . . . . .	305
8.4	LDAP-Enumeration . . . . .	310
8.4.1	LDAP- und AD-Grundlagen . . . . .	310
8.4.2	Der Enumeration-Prozess . . . . .	312
8.5	SMTP-Enumeration . . . . .	314
8.5.1	SMTP-Grundlagen . . . . .	314
8.5.2	Der Enumeration-Prozess . . . . .	315
8.6	NTP-Enumeration . . . . .	317
8.6.1	Funktionsweise von NTP . . . . .	317
8.6.2	Der Enumeration-Prozess . . . . .	318
8.7	DNS-Enumeration . . . . .	319
8.7.1	NFS-Enumeration . . . . .	324

8.8	8.7.2 Weitere Enumeration-Techniken . . . . .	326
8.9	8.9 Schutzmaßnahmen gegen Enumeration . . . . .	326
8.9	8.9 Zusammenfassung und Prüfungstipps . . . . .	328
	8.9.1 Zusammenfassung und Weiterführendes . . . . .	328
	8.9.2 CEH-Prüfungstipps . . . . .	329
	8.9.3 Fragen zur CEH-Prüfungsvorbereitung . . . . .	329
9	<b>9 Vulnerability-Scanning und Schwachstellenanalyse</b> . . . . .	331
9.1	9.1 Was steckt hinter Vulnerability-Scanning? . . . . .	331
	9.1.1 Vulnerabilities und Exploits . . . . .	332
	9.1.2 Common Vulnerabilities and Exposures (CVE) . . . . .	332
	9.1.3 CVE- und Exploit-Datenbanken . . . . .	333
	9.1.4 Vulnerability-Scanner . . . . .	335
9.2	9.2 Vulnerability-Scanning mit Nmap . . . . .	336
	9.2.1 Die Kategorie »vuln« . . . . .	336
	9.2.2 Die passenden Skripts einsetzen . . . . .	337
9.3	9.3 Nessus . . . . .	339
	9.3.1 Installation von Nessus . . . . .	339
	9.3.2 Vulnerability-Scanning mit Nessus . . . . .	341
	9.3.3 Nessus versus OpenVAS . . . . .	345
9.4	9.4 Rapid 7 Nexpose . . . . .	345
9.5	9.5 Vulnerability-Scanning in der Praxis . . . . .	346
	9.5.1 Vulnerability-Assessments . . . . .	346
	9.5.2 Einsatz von Vulnerability-Scannern im Ethical Hacking . . . . .	348
	9.5.3 Credential Scan vs. Remote Scan . . . . .	349
	9.5.4 Verifizieren der Schwachstelle . . . . .	349
	9.5.5 Exploits zum Testen von Schwachstellen . . . . .	350
	9.5.6 Spezialisierte Scanner . . . . .	350
9.6	9.6 Zusammenfassung und Prüfungstipps . . . . .	351
	9.6.1 Zusammenfassung und Weiterführendes . . . . .	351
	9.6.2 CEH-Prüfungstipps . . . . .	351
	9.6.3 Fragen zur CEH-Prüfungsvorbereitung . . . . .	352
	<b>Teil III Systeme angreifen . . . . .</b>	355
10	<b>10 Password Hacking . . . . .</b>	361
10.1	10.1 Zugriffsschutz mit Passwörtern und anderen Methoden . . . . .	362
10.2	10.2 Angriffsvektoren auf Passwörter . . . . .	363
10.3	10.3 Password Guessing und Password Recovery . . . . .	364
	10.3.1 Grundlagen des Password Guessings . . . . .	365
	10.3.2 Default-Passwörter . . . . .	366
	10.3.3 Password Recovery unter Windows . . . . .	369
	10.3.4 Password Recovery für Linux . . . . .	374
	10.3.5 Password Recovery auf Cisco-Routern . . . . .	375

## Inhaltsverzeichnis

10.4	Die Windows-Authentifizierung . . . . .	377
10.4.1	Die SAM-Datenbank . . . . .	377
10.4.2	LM und NTLM . . . . .	378
10.4.3	Kerberos . . . . .	379
10.4.4	NTLM-Hashes auslesen mit FGdump . . . . .	383
10.5	Die Linux-Authentifizierung . . . . .	385
10.5.1	Speicherorte der Login-Daten . . . . .	385
10.5.2	Passwort-Hashes unter Linux . . . . .	386
10.5.3	Der Salt – Passwort-Hashes »salzen« . . . . .	386
10.5.4	Wie gelangen wir an die Passwort-Hashes? . . . . .	387
10.6	Passwort-Hashes angreifen . . . . .	389
10.6.1	Angriffsvektoren auf Passwort-Hashes . . . . .	389
10.6.2	Pass the Hash (PTH) . . . . .	392
10.6.3	Wortlisten erstellen . . . . .	394
10.6.4	L0phtcrack . . . . .	398
10.6.5	John the Ripper . . . . .	400
10.6.6	Cain & Abel . . . . .	402
10.7	Online-Angriffe auf Passwörter . . . . .	402
10.7.1	Grundlegende Problematik . . . . .	402
10.7.2	Medusa . . . . .	403
10.7.3	Hydra . . . . .	405
10.7.4	Ncrack . . . . .	406
10.8	Distributed Network Attack (DNA) . . . . .	408
10.8.1	Funktionsweise . . . . .	408
10.8.2	ElcomSoft Distributed Password Recovery . . . . .	409
10.9	Schutzmaßnahmen gegen Password Hacking . . . . .	409
10.10	Zusammenfassung und Prüfungstipps . . . . .	410
10.10.1	Zusammenfassung und Weiterführendes . . . . .	410
10.10.2	CEH-Prüfungstipps . . . . .	411
10.10.3	Frage zur CEH-Prüfungsvorbereitung . . . . .	412
11	<b>Shells und Post-Exploitation</b> . . . . .	413
11.1	Remote-Zugriff mit Shell und Backdoor . . . . .	413
11.1.1	Einführung in Shells und Backdoors . . . . .	414
11.1.2	Netcat und Ncat – Einführung . . . . .	416
11.1.3	Grundlegende Funktionsweise von Netcat und Ncat . . . . .	417
11.1.4	Eine Bind-Shell bereitstellen . . . . .	421
11.1.5	Eine Reverse-Shell bereitstellen . . . . .	422
11.1.6	Wo stehen wir jetzt? . . . . .	424
11.2	Grundlagen Privilegien-Eskalation . . . . .	424
11.2.1	Vertikale Rechteerweiterung . . . . .	424
11.2.2	Horizontale Rechteerweiterung . . . . .	425
11.2.3	Rechte von Programmen . . . . .	425
11.3	Mit Privilegien-Eskalation zur Root-Shell . . . . .	426
11.3.1	Reverse-Shell durch DistCC-Exploit . . . . .	426

11.3.2	Bereitstellung eines Post-Exploits . . . . .	428
11.3.3	Mit Metasploit-Multi-Handler zur Root-Shell. . . . .	431
11.4	Meterpreter – die Luxus-Shell für Hacker . . . . .	432
11.4.1	Exploits und Payload . . . . .	433
11.4.2	Einführung in Meterpreter . . . . .	433
11.4.3	Meterpreter-Shell in der Praxis . . . . .	435
11.4.4	Eine Meterpreter-Shell für Windows erstellen . . . . .	437
11.4.5	Externe Module in Meterpreter laden . . . . .	440
11.5	Empire – Das Powershell-Post-Exploitation-Framework . . . . .	442
11.5.1	Das Szenario . . . . .	442
11.5.2	Bereitstellung von Empire . . . . .	443
11.5.3	Grundlagen: Listener, Stager, Agents . . . . .	444
11.5.4	Empire in Aktion: Module nutzen . . . . .	447
11.6	Verteidigungsmaßnahmen gegen Privilegien-Eskalation . . . . .	449
11.7	Zusammenfassung und Prüfungstipps . . . . .	450
11.7.1	Zusammenfassung und Weiterführendes . . . . .	450
11.7.2	CEH-Prüfungstipps . . . . .	451
11.7.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	451
<b>12</b>	<b>Mit Malware das System übernehmen . . . . .</b>	<b>453</b>
12.1	Malware-Grundlagen . . . . .	454
12.1.1	Typische Malware-Kategorien . . . . .	454
12.1.2	Wie gelangt Malware auf das Opfer-System? . . . . .	456
12.1.3	Eine selbst erstellte Malware . . . . .	458
12.2	Viren und Würmer . . . . .	459
12.2.1	Was ist ein Computervirus? . . . . .	459
12.2.2	Was ist ein Computerwurm? . . . . .	461
12.2.3	Einen Makro-Virus erstellen . . . . .	462
12.3	Trojanische Pferde in der Praxis . . . . .	466
12.3.1	Trojaner-Typen . . . . .	466
12.3.2	Einen Trojaner selbst bauen . . . . .	468
12.3.3	Viren- und Trojaner-Baukästen . . . . .	471
12.4	Malware tarnen und vor Entdeckung schützen. . . . .	473
12.4.1	Grundlagen der Tarnung von Payload . . . . .	473
12.4.2	Encoder einsetzen . . . . .	476
12.4.3	Payload mit Hyperion verschlüsseln . . . . .	479
12.4.4	Das Veil-Framework . . . . .	480
12.4.5	Shellter AV Evasion . . . . .	480
12.4.6	Fileless Malware . . . . .	481
12.5	Rootkits . . . . .	483
12.5.1	Grundlagen der Rootkits . . . . .	483
12.5.2	Kernel-Rootkits . . . . .	484
12.5.3	Userland-Rootkits . . . . .	484
12.5.4	Rootkit-Beispiele . . . . .	485
12.5.5	Rootkits entdecken und entfernen . . . . .	485

## Inhaltsverzeichnis

12.6	Covert Channel . . . . .	486
12.6.1	ICMP-Tunneling . . . . .	487
12.6.2	NTFS Alternate Data Stream (ADS) . . . . .	490
12.7	Keylogger und Spyware . . . . .	492
12.7.1	Grundlagen . . . . .	492
12.7.2	Keylogger und Spyware in der Praxis . . . . .	492
12.8	Advanced Persistent Threat (APT) . . . . .	497
12.8.1	Wie funktioniert ein APT? . . . . .	497
12.8.2	Ablauf eines APT-Angriffs . . . . .	498
12.8.3	Zielgruppen von APT-Angriffen . . . . .	498
12.9	Schutzmaßnahmen gegen Malware . . . . .	499
12.10	Zusammenfassung und Prüfungstipps . . . . .	499
12.10.1	Zusammenfassung und Weiterführendes . . . . .	499
12.10.2	CEH-Prüfungstipps . . . . .	500
12.10.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	500
<b>13</b>	<b>Malware-Erkennung und -Analyse . . . . .</b>	<b>503</b>
13.1	Grundlagen der Malware-Analyse . . . . .	503
13.1.1	Statische Malware-Analyse . . . . .	504
13.1.2	Dynamische Malware-Analyse . . . . .	507
13.2	Verdächtiges Verhalten analysieren . . . . .	507
13.2.1	Virencheck durchführen . . . . .	508
13.2.2	Prozesse überprüfen . . . . .	512
13.2.3	Netzwerkaktivitäten prüfen . . . . .	515
13.2.4	Die Windows-Registrierung checken . . . . .	520
13.2.5	Autostart-Einträge unter Kontrolle . . . . .	524
13.2.6	Windows-Dienste checken . . . . .	526
13.2.7	Treiber überprüfen . . . . .	528
13.2.8	Integrität der Systemdateien prüfen . . . . .	530
13.2.9	Datei-Integrität durch Prüfsummen-Check . . . . .	531
13.2.10	System-Integrität mit Tripwire sichern . . . . .	533
13.3	Sheep-Dip-Systeme . . . . .	534
13.3.1	Einführung . . . . .	534
13.3.2	Aufbau eines Sheep-Dip-Systems . . . . .	535
13.4	Schutz durch Sandbox . . . . .	536
13.4.1	Sandboxie . . . . .	536
13.4.2	Cuckoo . . . . .	538
13.5	Aufbau einer modernen Anti-Malware-Infrastruktur . . . . .	539
13.5.1	Relevante Komponenten . . . . .	540
13.5.2	Komponenten der Anti-Malware-Infrastruktur . . . . .	540
13.6	Allgemeine Schutzmaßnahmen vor Malware-Infektion . . . . .	542
13.7	Zusammenfassung und Prüfungstipps . . . . .	543
13.7.1	Zusammenfassung und Weiterführendes . . . . .	543
13.7.2	CEH-Prüfungstipps . . . . .	544
13.7.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	545

<b>14</b>	<b>Steganografie . . . . .</b>	547
14.1	Grundlagen der Steganografie . . . . .	547
14.1.1	Wozu Steganografie? . . . . .	547
14.1.2	Ein paar einfache Beispiele . . . . .	548
14.1.3	Klassifikation der Steganografie . . . . .	549
14.2	Computergestützte Steganografie . . . . .	553
14.2.1	Daten in Bildern verstecken . . . . .	553
14.2.2	Daten in Dokumenten verstecken . . . . .	558
14.2.3	Weitere Cover-Datenformate . . . . .	559
14.3	Steganalyse und Schutz vor Steganografie . . . . .	560
14.3.1	Methoden der Steganalyse . . . . .	560
14.3.2	Steganalyse-Tools . . . . .	561
14.3.3	Schutz vor Steganografie . . . . .	561
14.4	Zusammenfassung und Prüfungstipps . . . . .	562
14.4.1	Zusammenfassung und Weiterführendes . . . . .	562
14.4.2	CEH-Prüfungstipps . . . . .	563
14.4.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	563
<b>15</b>	<b>Spuren verwischen . . . . .</b>	565
15.1	Auditing und Logging . . . . .	565
15.1.1	Die Windows-Protokollierung . . . . .	566
15.1.2	Die klassische Linux-Protokollierung . . . . .	568
15.2	Spuren verwischen auf einem Windows-System . . . . .	571
15.2.1	Das Windows-Auditing deaktivieren . . . . .	571
15.2.2	Windows-Ereignisprotokolle löschen . . . . .	573
15.2.3	Most Recently Used (MRU) löschen . . . . .	575
15.2.4	Zeitstempel manipulieren . . . . .	578
15.2.5	Clearing-Tools . . . . .	582
15.3	Spuren verwischen auf einem Linux-System . . . . .	583
15.3.1	Logfiles manipulieren und löschen . . . . .	583
15.3.2	Systemd-Logging in Journald . . . . .	586
15.3.3	Zeitstempel manipulieren . . . . .	586
15.3.4	Die Befehlszeilen-Historie löschen . . . . .	588
15.4	Schutz vor dem Spuren-Verwischen . . . . .	589
15.5	Zusammenfassung und Prüfungstipps . . . . .	590
15.5.1	Zusammenfassung und Weiterführendes . . . . .	590
15.5.2	CEH-Prüfungstipps . . . . .	591
15.5.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	591
<b>Teil IV</b>	<b>Netzwerk- und sonstige Angriffe . . . . .</b>	595
<b>16</b>	<b>Network Sniffing mit Wireshark &amp; Co. . . . .</b>	599
16.1	Grundlagen von Netzwerk-Sniffern . . . . .	599
16.1.1	Technik der Netzwerk-Sniffer . . . . .	599

16.1.2	Wireshark und die Pcap-Bibliotheken . . . . .	601
16.2	Wireshark installieren und starten . . . . .	601
16.2.1	Installation unter Linux . . . . .	601
16.2.2	Installation unter Windows . . . . .	602
16.2.3	Der erste Start . . . . .	603
16.3	Die ersten Schritte mit Wireshark . . . . .	604
16.3.1	Grundeinstellungen . . . . .	604
16.3.2	Ein erster Mitschnitt . . . . .	606
16.4	Mitschnitt-Filter einsetzen . . . . .	607
16.4.1	Analyse eines TCP-Handshakes . . . . .	608
16.4.2	Der Ping in Wireshark . . . . .	609
16.4.3	Weitere Mitschnittfilter . . . . .	610
16.5	Anzeigefilter einsetzen . . . . .	611
16.5.1	Eine HTTP-Sitzung im Detail . . . . .	612
16.5.2	Weitere Anzeigefilter . . . . .	614
16.6	Passwörter und andere Daten ausspähen . . . . .	615
16.6.1	FTP-Zugangsdaten ermitteln . . . . .	616
16.6.2	Telnet-Zugangsdaten identifizieren . . . . .	617
16.6.3	SSH – sicherer Schutz gegen Mitlesen . . . . .	619
16.6.4	Andere Daten ausspähen . . . . .	621
16.7	Auswertungsfunktionen von Wireshark nutzen . . . . .	622
16.8	Tcpdump und TShark einsetzen . . . . .	624
16.8.1	Tcpdump – der Standard-Sniffer für die Konsole . . . . .	624
16.8.2	TShark – Wireshark auf der Konsole . . . . .	627
16.9	Zusammenfassung und Prüfungstipps . . . . .	629
16.9.1	Zusammenfassung und Weiterführendes . . . . .	629
16.9.2	CEH-Prüfungstipps . . . . .	629
16.9.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	630
17	<b>Lauschangriffe &amp; Man-in-the-Middle . . . . .</b>	633
17.1	Eavesdropping und Sniffing für Hacker . . . . .	633
17.1.1	Eavesdropping und Wiretapping . . . . .	634
17.1.2	Sniffing als Angriiffsvektor . . . . .	634
17.2	Man-in-the-Middle (MITM) . . . . .	635
17.2.1	Was bedeutet Man-in-the-Middle? . . . . .	636
17.2.2	Was erreichen wir durch einen MITM-Angriff? . . . . .	637
17.3	Active Sniffing . . . . .	637
17.3.1	Mirror-Ports: Ein Kabel mit drei Enden . . . . .	638
17.3.2	Aus Switch mach Hub – MAC-Flooding . . . . .	638
17.3.3	Auf dem Silbertablett: WLAN-Sniffing . . . . .	640
17.3.4	Weitere physische Abhörmöglichkeiten . . . . .	641
17.4	Die Kommunikation für MITM umleiten . . . . .	641
17.4.1	Physische Umleitung . . . . .	641
17.4.2	Umleitung über aktive Netzwerk-Komponenten . . . . .	642
17.4.3	Umleiten mit ARP-Spoofing . . . . .	643

17.4.4	ICMP-Typ 5 Redirect .....	643
17.4.5	DNS-Spoofing oder DNS-Cache-Poisoning .....	644
17.4.6	Manipulation der hosts-Datei.....	646
17.4.7	Umleiten via DHCP-Spoofing.....	647
17.5	Die Dsniff-Toolsammlung .....	648
17.5.1	Programme der Dsniff-Suite .....	648
17.5.2	Abhören des Netzwerk-Traffics .....	649
17.5.3	MITM mit arpspoof .....	650
17.5.4	Die ARP-Tabelle des Switches mit macof überfluten .....	653
17.5.5	DNS-Spoofing mit dnspoof .....	653
17.5.6	Dsniff.....	656
17.6	Man-in-the-Middle-Angriffe mit Ettercap .....	657
17.6.1	Einführung in Ettercap.....	657
17.6.2	DNS-Spoofing mit Ettercap .....	659
17.7	Schutz vor Lauschangriffen & MITM .....	667
17.8	Zusammenfassung und Prüfungstipps.....	669
17.8.1	Zusammenfassung und Weiterführendes .....	669
17.8.2	CEH-Prüfungstipps .....	670
17.8.3	Fragen zur CEH-Prüfungsvorbereitung .....	670
<b>18</b>	<b>Session Hijacking .....</b>	<b>673</b>
18.1	Grundlagen des Session Hijackings .....	673
18.1.1	Wie funktioniert Session Hijacking grundsätzlich?.....	674
18.1.2	Session-Hijacking-Varianten .....	674
18.2	Network Level Session Hijacking.....	675
18.2.1	Die TCP-Session im Detail.....	676
18.2.2	Entführen von TCP-Sessions.....	678
18.2.3	Eine Telnet-Session entführen.....	680
18.2.4	Weitere Hijacking-Varianten auf Netzwerk-Ebene .....	685
18.3	Application Level Session Hijacking .....	686
18.3.1	Die Session-IDs.....	686
18.3.2	Die Session-ID ermitteln .....	687
18.3.3	Sniffing/Man-in-the-Middle.....	688
18.3.4	Die Session-ID erraten – das Prinzip .....	688
18.3.5	WebGoat bereitstellen .....	689
18.3.6	Die Burp Suite – Grundlagen und Installation .....	692
18.3.7	Burp Suite als Intercepting Proxy .....	693
18.3.8	Der Burp Sequencer – Session-IDs analysieren.....	697
18.3.9	Entführen der Session mithilfe der Session-ID .....	700
18.3.10	Man-in-the-Browser-Angriff.....	707
18.3.11	Weitere Angriffsformen .....	709
18.4	Gegenmaßnahmen gegen Session Hijacking.....	711
18.4.1	Session Hijacking entdecken.....	711
18.4.2	Schutzmaßnahmen .....	712

18.5	Zusammenfassung und Prüfungstipps .....	714
18.5.1	Zusammenfassung und Weiterführendes.....	714
18.5.2	CEH-Prüfungstipps .....	715
18.5.3	Fragen zur CEH-Prüfungsvorbereitung.....	715
<b>19</b>	<b>Firewalls, IDS/IPS und Honeypots einsetzen und umgehen.....</b>	<b>717</b>
19.1	Firewall-Technologien .....	717
19.1.1	Netzwerk- und Personal-Firewalls .....	718
19.1.2	Filtertechniken und Kategorisierung der Netzwerk-Firewalls .....	719
19.2	Firewall-Szenarien .....	723
19.2.1	DMZ-Szenarien .....	723
19.2.2	Failover-Szenarien .....	725
19.3	Firewalls umgehen .....	726
19.3.1	Identifikation von Firewalls .....	726
19.3.2	IP-Adress-Spoofing .....	727
19.3.3	Was wirklich funktioniert .....	728
19.4	Intrusion-Detection- und -Prevention-Systeme .....	729
19.4.1	Einführung in Snort .....	732
19.5	Intrusion-Detection-Systeme umgehen .....	736
19.5.1	Injection/Insertion .....	736
19.5.2	Evasion .....	737
19.5.3	Denial-of-Service-Angriff (DoS) .....	738
19.5.4	Obfuscation .....	738
19.5.5	Generieren von False Positives .....	738
19.5.6	Fragmentation .....	739
19.5.7	TCP Session Splicing .....	740
19.5.8	Weitere Evasion-Techniken .....	740
19.6	Honeypots .....	741
19.6.1	Grundlagen und Begriffsklärung .....	741
19.6.2	Kategorisierung der Honeypots .....	742
19.6.3	KFSensor – ein Honeypot in der Praxis .....	745
19.6.4	Honeypots identifizieren und umgehen .....	749
19.6.5	Rechtliche Aspekte beim Einsatz von Honeypots .....	750
19.7	Zusammenfassung und Prüfungstipps .....	751
19.7.1	Zusammenfassung und Weiterführendes .....	751
19.7.2	CEH-Prüfungstipps .....	752
19.7.3	Fragen zur CEH-Prüfungsvorbereitung .....	752
<b>20</b>	<b>Social Engineering .....</b>	<b>755</b>
20.1	Einführung in das Social Engineering .....	755
20.1.1	Welche Gefahren birgt Social Engineering? .....	756
20.1.2	Verlustangst, Neugier, Eitelkeit – die Schwachstellen des Systems Mensch .....	756
20.1.3	Varianten des Social Engineerings .....	759
20.1.4	Allgemeine Vorgehensweise beim Social Engineering .....	761

20.2	Human Based Social Engineering . . . . .	761
20.2.1	Vortäuschen einer anderen Identität . . . . .	762
20.2.2	Shoulder Surfing & Co. . . . .	764
20.2.3	Piggybacking und Tailgaiting . . . . .	765
20.3	Computer Based Social Engineering . . . . .	766
20.3.1	Phishing . . . . .	766
20.3.2	Pharming . . . . .	766
20.3.3	Spear Phishing . . . . .	767
20.3.4	Drive-by-Downloads . . . . .	768
20.3.5	Gefälschte Viren-Warnungen . . . . .	769
20.4	Das Social-Engineer Toolkit (SET) . . . . .	770
20.4.1	Einführung in SET . . . . .	770
20.4.2	Praxisdemonstration: Credential Harvester . . . . .	772
20.4.3	Weitere Angriffe mit SET . . . . .	775
20.5	So schützen Sie sich gegen Social-Engineering-Angriffe . . . . .	776
20.6	Zusammenfassung und Prüfungstipps . . . . .	778
20.6.1	Zusammenfassung und Weiterführendes . . . . .	778
20.6.2	CEH-Prüfungstipps . . . . .	779
20.6.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	779
<b>21</b>	<b>Hacking-Hardware . . . . .</b>	<b>781</b>
21.1	Allgemeines und rechtliche Hinweise zu Spionage-Hardware . . . . .	782
21.2	Angriffsvektor USB-Schnittstelle . . . . .	782
21.2.1	Hardware Keylogger . . . . .	783
21.2.2	USB Rubber Ducky . . . . .	784
21.2.3	Bash Bunny . . . . .	786
21.2.4	Digispark . . . . .	788
21.2.5	USBNinja . . . . .	789
21.2.6	Mouse Jiggler . . . . .	790
21.3	Weitere Hacking-Gadgets . . . . .	790
21.3.1	VideoGhost . . . . .	790
21.3.2	Packet Squirrel . . . . .	791
21.3.3	LAN Turtle . . . . .	792
21.3.4	Throwing Star LAN Tap . . . . .	792
21.3.5	Software Defined Radio . . . . .	793
21.3.6	Crazyradio PA . . . . .	793
21.3.7	WiFi Pineapple . . . . .	794
21.3.8	Proxmark 3 . . . . .	795
21.3.9	ChameleonMini . . . . .	795
21.4	Raspberry Pi als Hacking-Kit . . . . .	795
21.5	Gegenmaßnahmen . . . . .	797
21.6	Zusammenfassung und Prüfungstipps . . . . .	799
21.6.1	Zusammenfassung und Weiterführendes . . . . .	799
21.6.2	CEH-Prüfungstipps . . . . .	800
21.6.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	800

<b>22</b>	<b>DoS- und DDoS-Angriffe</b>	803
22.1	DoS- und DDoS-Grundlagen	803
22.1.1	Was ist ein Denial-of-Service-Angriff?	804
22.1.2	Warum werden DoS- und DDoS-Angriffe durchgeführt?	804
22.1.3	Kategorien der DoS/DDoS-Angriffe	805
22.2	DoS- und DDoS-Angriffstechniken	805
22.2.1	UDP-Flood-Angriff	806
22.2.2	ICMP-Flood-Angriff	806
22.2.3	Smurf-Angriff	807
22.2.4	Syn-Flood-Angriff	808
22.2.5	Fragmentation-Angriff	811
22.2.6	Slowloris-Angriff	812
22.2.7	Permanenter Denial-of-Service (PDoS)	813
22.2.8	Distributed-Reflected-Denial-of-Service-Angriff (DRDoS)	814
22.3	Botnetze – Funktionsweise und Betrieb	815
22.3.1	Bots und deren Einsatzmöglichkeiten	816
22.3.2	Aufbau eines Botnetzes	816
22.3.3	Wie gelangen Bots auf die Opfer-Systeme?	818
22.3.4	Mobile Systeme und IoT	819
22.3.5	Botnetze in der Praxis	819
22.3.6	Verteidigung gegen Botnetze und DDoS-Angriffe	820
22.4	DoS-Angriffe in der Praxis	822
22.4.1	SYN- und ICMP-Flood-Angriff mit hping3	823
22.4.2	DoS-Angriff mit Metasploit	825
22.4.3	DoS-Angriff mit SlowHTTPTest	827
22.4.4	Low Orbit Ion Cannon (LOIC)	828
22.5	Verteidigung gegen DoS- und DDoS-Angriffe	830
22.5.1	Allgemeiner Grundschutz	830
22.5.2	Schutz vor volumetrischen DDoS-Angriffen	831
22.6	Zusammenfassung und Prüfungstipps	832
22.6.1	Zusammenfassung und Weiterführendes	832
22.6.2	CEH-Prüfungstipps	833
22.6.3	Fragen zur CEH-Prüfungsvorbereitung	833
<b>Teil V</b>	<b>Web-Hacking</b>	835
<b>23</b>	<b>Web-Hacking – Grundlagen</b>	839
23.1	Was ist Web-Hacking?	839
23.2	Architektur von Webanwendungen	840
23.2.1	Die Schichten-Architektur	840
23.2.2	Die URL-Codierung	841
23.2.3	Das Hypertext Transfer Protocol (HTTP)	842
23.2.4	Cookies	845

23.2.5	HTTP vs. HTTPS . . . . .	845
23.2.6	Webservices und -technologien . . . . .	846
23.3	Die gängigsten Webserver: Apache, IIS, nginx . . . . .	851
23.3.1	Apache HTTP Server . . . . .	851
23.3.2	Internet Information Services (IIS) . . . . .	853
23.3.3	nginx . . . . .	855
23.4	Typische Schwachstellen von Webservern und -anwendungen . . . . .	856
23.4.1	Schwachstellen in Webserver-Plattformen . . . . .	856
23.4.2	Schwachstellen in der Webanwendung . . . . .	857
23.5	Reconnaissance für Web-Hacking-Angriffe . . . . .	858
23.5.1	Footprinting und Scanning . . . . .	858
23.5.2	Web-Firewalls und Proxys entlarven . . . . .	860
23.5.3	Hidden Content Discovery . . . . .	860
23.5.4	Website-Mirroring . . . . .	863
23.5.5	Security-Scanner . . . . .	863
23.6	Praxis-Szenario: Einen Apache-Webserver mit Shellshock hacken . . . . .	866
23.6.1	Die Laborumgebung präparieren . . . . .	866
23.6.2	Den Angriff durchführen . . . . .	868
23.7	Praxis-Szenario 2: Angriff auf WordPress . . . . .	869
23.7.1	WordPress-VM bereitstellen . . . . .	870
23.7.2	WordPress scannen und Enumeration . . . . .	874
23.7.3	User-Hacking . . . . .	876
23.8	Zusammenfassung und Prüfungstipps . . . . .	876
23.8.1	Zusammenfassung und Weiterführendes . . . . .	876
23.8.2	CEH-Prüfungstipps . . . . .	877
23.8.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	877
24	<b>Web-Hacking – OWASP Top 10 . . . . .</b>	879
24.1	Einführung in die OWASP-Projekte . . . . .	879
24.2	WebGoat & Co – virtuelle Sandsäcke für das Web-Hacking-Training . . . . .	883
24.2.1	WebGoat . . . . .	883
24.2.2	Mutillidae II . . . . .	884
24.2.3	bWAPP . . . . .	885
24.2.4	DVWA . . . . .	886
24.2.5	OWASP Broken Web Application . . . . .	887
24.2.6	Web Security Dojo . . . . .	887
24.2.7	Vulnhub und Pentesterlab . . . . .	888
24.3	Die OWASP Top 10 in der Übersicht . . . . .	888
24.4	A1 – Injection . . . . .	889
24.4.1	Kategorien von Injection-Angriffen . . . . .	889
24.4.2	Beispiel für einen Injection-Angriff . . . . .	890
24.5	A2 – Fehler in der Authentifizierung . . . . .	892
24.5.1	Grundlagen . . . . .	892
24.5.2	Identitätsdiebstahl durch Token-Manipulation . . . . .	893
24.5.3	Schutzmaßnahmen . . . . .	896

## Inhaltsverzeichnis

24.6	A3 – Verlust der Vertraulichkeit sensibler Daten . . . . .	896
24.6.1	Welche Daten sind betroffen? . . . . .	896
24.6.2	Angriffsszenarien . . . . .	897
24.6.3	Schutzmaßnahmen . . . . .	898
24.7	A4 – XML External Entities (XXE) . . . . .	899
24.7.1	XML-Entities . . . . .	899
24.7.2	Ein Beispiel für einen XXE-Angriff . . . . .	900
24.7.3	Schutzmaßnahmen . . . . .	901
24.8	A5 – Fehler in der Zugriffskontrolle . . . . .	902
24.8.1	Unsichere direkte Objektreferenzen . . . . .	902
24.8.2	Fehlerhafte Autorisierung auf Anwendungsebene . . . . .	904
24.8.3	Schutzmaßnahmen . . . . .	907
24.9	A6 – Sicherheitsrelevante Fehlkonfiguration . . . . .	907
24.9.1	Typische Fehlkonfigurationen . . . . .	907
24.9.2	Directory Browsing . . . . .	908
24.9.3	Schutzmaßnahmen . . . . .	910
24.10	A7 – Cross-Site-Scripting (XSS) . . . . .	910
24.10.1	Wie funktioniert XSS? . . . . .	911
24.10.2	Ein einfaches XSS-Beispiel . . . . .	911
24.10.3	XSS-Varianten . . . . .	913
24.10.4	Ein Beispiel für Stored XSS . . . . .	915
24.10.5	Exkurs: Cross-Site-Request-Forgery (CSRF) . . . . .	917
24.10.6	Schutzmaßnahmen . . . . .	918
24.11	A8 – Unsichere Deserialisierung . . . . .	919
24.11.1	Was bedeutet Serialisierung von Daten? . . . . .	919
24.11.2	Wie wird die Deserialisierung zum Problem? . . . . .	920
24.11.3	Schutzmaßnahmen . . . . .	920
24.12	A9 – Nutzung von Komponenten mit bekannten Schwachstellen . . . . .	921
24.12.1	Wo liegt die Gefahr und wer ist gefährdet? . . . . .	921
24.12.2	Verwundbare JavaScript-Bibliotheken aufdecken mit Retire.js . . . . .	921
24.12.3	Schutzmaßnahmen . . . . .	922
24.13	A10 – Unzureichendes Logging & Monitoring . . . . .	923
24.13.1	Herausforderungen beim Logging & Monitoring . . . . .	923
24.13.2	Sind unserer Systeme gefährdet? . . . . .	924
24.14	Zusammenfassung und Prüfungstipps . . . . .	925
24.14.1	Zusammenfassung und Weiterführendes . . . . .	925
24.14.2	CEH-Prüfungstipps . . . . .	925
24.14.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	926
25	<b>SQL-Injection</b> . . . . .	929
25.1	Mit SQL-Injection das Login austricksen . . . . .	930
25.1.1	Der grundlegende Ansatz . . . . .	930
25.1.2	Anmeldung als gewünschter Benutzer . . . . .	933
25.1.3	Clientseitige Sicherheit . . . . .	934

25.2	Daten auslesen mit SQL-Injection . . . . .	936
25.2.1	Manipulation eines GET-Requests . . . . .	937
25.2.2	Informationen über die Datenbank auslesen . . . . .	938
25.2.3	Die Datenbank-Tabellen identifizieren . . . . .	940
25.2.4	Spalten und Passwörter auslesen . . . . .	942
25.3	Fortgeschrittene SQL-Injection-Techniken . . . . .	943
25.3.1	Einführung in Blind SQL-Injection . . . . .	944
25.3.2	Codieren des Injection-Strings . . . . .	946
25.3.3	Blind SQLi: Eins oder null? . . . . .	949
25.3.4	Time based SQL-Injection . . . . .	950
25.4	SQLMap – automatische Schwachstellensuche . . . . .	952
25.4.1	SQLi-CheatSheets . . . . .	952
25.4.2	Einführung in SQLMap . . . . .	953
25.4.3	Weitere Analysen mit SQLMap . . . . .	958
25.5	Schutzmaßnahmen vor SQLi-Angriffen . . . . .	960
25.6	Zusammenfassung und Prüfungstipps . . . . .	961
25.6.1	Zusammenfassung und Weiterführendes . . . . .	961
25.6.2	CEH-Prüfungstipps . . . . .	961
25.6.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	962
<b>26</b>	<b>Web-Hacking – sonstige Injection-Angriffe . . . . .</b>	<b>965</b>
26.1	Command-Injection . . . . .	965
26.1.1	Einführung in Command-Injection-Angriffe . . . . .	966
26.1.2	Command-Injection in der Praxis . . . . .	966
26.1.3	Schutzmaßnahmen vor Command-Injection-Angriffen . . . . .	968
26.2	LDAP-Injection . . . . .	969
26.2.1	Die LDAP-Infrastruktur bereitstellen . . . . .	969
26.2.2	Ein erster Injection-Angriff . . . . .	973
26.2.3	LDAP-Injection mit der BurpSuite vereinfachen . . . . .	975
26.2.4	LDAP-Injection-Discovery . . . . .	976
26.2.5	Discovery-Automatisierung mit Hilfe der BurpSuite . . . . .	977
26.2.6	Flexibilität und Geduld sind gefragt . . . . .	981
26.2.7	Schutz vor LDAP-Injection-Angriffen . . . . .	982
26.3	File-Injection . . . . .	982
26.3.1	Directory-Traversal-Angriffe . . . . .	983
26.3.2	File-Upload-Angriffe . . . . .	985
26.3.3	Local File Inclusion versus Remote File Inclusion . . . . .	987
26.4	Zusammenfassung und Prüfungstipps . . . . .	991
26.4.1	Zusammenfassung und Weiterführendes . . . . .	991
26.4.2	CEH-Prüfungstipps . . . . .	991
26.4.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	991
<b>27</b>	<b>Buffer-Overflow-Angriffe . . . . .</b>	<b>993</b>
27.1	Wie funktioniert ein Buffer-Overflow-Angriff? . . . . .	993
27.1.1	Das Grundprinzip . . . . .	994

27.1.2	Welche Anwendungen sind verwundbar? . . . . .	994
27.1.3	Funktionsweise des Stacks . . . . .	995
27.1.4	Register . . . . .	995
27.2	Ein Buffer-Overflow-Angriff in der Praxis . . . . .	997
27.2.1	SLmail-Exploit . . . . .	997
27.2.2	Die Laborumgebung . . . . .	997
27.2.3	Der Immunity Debugger . . . . .	999
27.2.4	Fuzzing . . . . .	1002
27.2.5	Einen eindeutigen String erstellen . . . . .	1006
27.2.6	Den EIP lokalisieren . . . . .	1008
27.2.7	Den Shellcode platzieren . . . . .	1008
27.2.8	Bad Characters identifizieren . . . . .	1010
27.2.9	Grundüberlegung: Wohin soll der EIP zeigen? . . . . .	1012
27.2.10	Mona und die Module . . . . .	1012
27.2.11	Die Anweisung JMP ESP auffinden . . . . .	1013
27.2.12	Den Programmablauf über den EIP steuern . . . . .	1015
27.2.13	Den Shellcode erstellen und ausführen . . . . .	1017
27.3	Heap-Overflow-Angriffe . . . . .	1021
27.3.1	Der Heap . . . . .	1021
27.3.2	Heap Overflow versus Stack Overflow . . . . .	1022
27.3.3	Use-after-free . . . . .	1022
27.3.4	Heap Spraying . . . . .	1022
27.4	Schutzmaßnahmen gegen Buffer-Overflow-Angriffe . . . . .	1023
27.4.1	Address Space Layout Randomization (ASLR) . . . . .	1023
27.4.2	Data Execution Prevention (DEP) . . . . .	1024
27.4.3	SEHOP und SafeSEH . . . . .	1024
27.4.4	Stack Canary . . . . .	1024
27.4.5	Wie sicher sind die Schutzmaßnahmen? . . . . .	1025
27.5	Zusammenfassung und Prüfungstipps . . . . .	1026
27.5.1	Zusammenfassung und Weiterführendes . . . . .	1026
27.5.2	CEH-Prüfungstipps . . . . .	1027
27.5.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	1027
<b>Teil VI</b>	<b>Angriffe auf WLAN und Next-Gen-Technologien . . . . .</b>	<b>1029</b>
28	<b>WLAN-Hacking . . . . .</b>	<b>1033</b>
28.1	WLAN-Grundlagen . . . . .	1033
28.1.1	Frequenzen und Kanäle . . . . .	1034
28.1.2	Der IEEE-802.11-Standard . . . . .	1035
28.1.3	Infrastruktur . . . . .	1036
28.1.4	Verbindungsaufbau . . . . .	1039
28.1.5	Verschlüsselungsmethoden . . . . .	1042
28.2	Setup für das WLAN-Hacking . . . . .	1045
28.2.1	Die WLAN-Hacking-Plattform . . . . .	1045

28.2.2	Der richtige WLAN-Adapter . . . . .	1046
28.2.3	Den Monitor Mode aktivieren . . . . .	1046
28.3	WLAN-Scanning und -Sniffing . . . . .	1048
28.3.1	Scanning . . . . .	1049
28.3.2	WLAN-Sniffing . . . . .	1049
28.3.3	Hidden SSIDs aufspüren . . . . .	1051
28.4	Angriffe auf WLAN . . . . .	1053
28.4.1	Denial of Service durch Störsender . . . . .	1053
28.4.2	Deauthentication-Angriff . . . . .	1053
28.4.3	Angriff auf WEP . . . . .	1055
28.4.4	Angriff auf WPA/WPA2 . . . . .	1058
28.4.5	Angriff auf WPA3 . . . . .	1060
28.4.6	Angriff auf WPS . . . . .	1061
28.4.7	MAC-Filter umgehen . . . . .	1064
28.4.8	WLAN-Passwörter auslesen . . . . .	1066
28.4.9	Standard-Passwörter . . . . .	1068
28.4.10	Captive Portals umgehen . . . . .	1069
28.5	Rogue Access Points . . . . .	1071
28.5.1	Fake-Access-Point bereitstellen . . . . .	1072
28.5.2	WLAN-Phishing . . . . .	1074
28.6	Schutzmaßnahmen . . . . .	1076
28.6.1	Allgemeine Maßnahmen . . . . .	1077
28.6.2	Fortgeschrittene Sicherheitsmechanismen . . . . .	1078
28.7	Zusammenfassung und Prüfungstipps . . . . .	1079
28.7.1	Zusammenfassung und Weiterführendes . . . . .	1079
28.7.2	CEH-Prüfungstipps . . . . .	1080
28.7.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	1080
29	<b>Mobile Hacking . . . . .</b>	1083
29.1	Grundlagen . . . . .	1083
29.1.1	Mobile Betriebssysteme . . . . .	1083
29.1.2	Apps und App-Stores . . . . .	1085
29.2	Angriffe auf mobile Geräte . . . . .	1087
29.2.1	Schutzziele . . . . .	1087
29.2.2	Angriffsvektoren . . . . .	1088
29.2.3	OWASP Mobile Top 10 . . . . .	1090
29.3	Mobile Hacking in der Praxis . . . . .	1091
29.3.1	Android über den PC . . . . .	1091
29.3.2	Android-Rooting . . . . .	1095
29.3.3	Jailbreaking iOS . . . . .	1101
29.3.4	SIM-Unlock . . . . .	1103
29.3.5	Hacking-Tools für Android . . . . .	1103
29.3.6	Android-Tojamer erstellen . . . . .	1106
29.3.7	Angriffe auf iOS . . . . .	1112
29.3.8	Spyware für mobile Geräte . . . . .	1112

29.4	Bring Your Own Device (BYOD) . . . . .	1113
29.4.1	BYOD-Vorteile . . . . .	1113
29.4.2	BYOD-Risiken . . . . .	1114
29.4.3	BYOD-Sicherheit . . . . .	1115
29.5	Mobile Device Management (MDM) . . . . .	1115
29.6	Schutzmaßnahmen . . . . .	1117
29.7	Zusammenfassung und Prüfungstipps . . . . .	1119
29.7.1	Zusammenfassung und Weiterführendes . . . . .	1119
29.7.2	CEH-Prüfungstipps . . . . .	1120
	29.7.3 Fragen zur CEH-Prüfungsvorbereitung . . . . .	1120
<b>30</b>	<b>IoT- und OT-Hacking und -Security</b> . . . . .	<b>1123</b>
30.1	Das Internet of Things . . . . .	1123
30.1.1	Was ist das Internet of Things? . . . . .	1124
30.1.2	Was umfasst das Internet of Things? . . . . .	1124
30.1.3	Die grundlegende Sicherheitsproblematik von IoT-Geräten . . . . .	1125
30.2	IoT-Technik – Konzepte und Protokolle . . . . .	1125
30.2.1	IoT-Betriebssysteme . . . . .	1126
30.2.2	IoT-Kommunikationsmodelle . . . . .	1126
30.2.3	IoT-Übertragungstechnologien . . . . .	1128
30.2.4	IoT-Kommunikationsprotokolle . . . . .	1130
30.3	Schwachstellen von IoT-Systemen . . . . .	1131
30.3.1	OWASP Top 10 IoT 2018 . . . . .	1131
30.3.2	Angriffsvektoren auf IoT-Systeme . . . . .	1133
30.4	IoT-Angriffszenarien . . . . .	1136
30.4.1	Rolling-Code-Angriff . . . . .	1136
30.4.2	Mirai – Botnet und DDoS-Angriffe . . . . .	1138
30.4.3	Lokale Angriffe über die UART-Schnittstelle . . . . .	1139
30.4.4	Command-Injection via Web-Frontend . . . . .	1140
30.4.5	Der BlueBorne-Angriff . . . . .	1141
30.4.6	Angriffe auf ZigBee-Geräte mit Killerbee . . . . .	1142
30.4.7	Angriffe auf Firmware . . . . .	1143
30.5	Weitere Angriffsformen auf IoT-Ökosysteme . . . . .	1144
30.5.1	Exploit Kits . . . . .	1144
30.5.2	IoT-Suchmaschinen . . . . .	1144
30.6	OT-Hacking . . . . .	1146
30.6.1	OT-Grundlagen und -Konzepte . . . . .	1146
30.6.2	Konvergenz von IT und OT . . . . .	1147
30.6.3	Das Purdue-Modell . . . . .	1148
30.6.4	OT-Sicherheitsherausforderungen . . . . .	1149
30.6.5	OT-Schwachstellen und Bedrohungen . . . . .	1150
30.6.6	OT-Malware . . . . .	1151
30.6.7	OT-Hackingtools und -Enumeration . . . . .	1152
30.6.8	Schutzmaßnahmen vor OT-Angriffen . . . . .	1153
30.7	Schutzmaßnahmen vor IoT-Angriffen . . . . .	1154

30.8	Zusammenfassung und Prüfungstipps . . . . .	1156
30.8.1	Zusammenfassung und Weiterführendes . . . . .	1156
30.8.2	CEH-Prüfungstipps . . . . .	1156
30.8.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	1156
<b>31</b>	<b>Angriffe auf die Cloud . . . . .</b>	<b>1159</b>
31.1	Grundlagen des Cloud Computings . . . . .	1159
31.1.1	Was ist eigentlich »die Cloud?« . . . . .	1160
31.1.2	Cloud-Service-Modelle . . . . .	1161
31.1.3	Deployment-Modelle für die Cloud . . . . .	1162
31.1.4	Große Cloud-Anbieter . . . . .	1164
31.2	Wichtige Cloud-Technologien . . . . .	1165
31.2.1	Virtualisierung . . . . .	1165
31.2.2	Container-Technologien . . . . .	1166
31.2.3	Docker . . . . .	1168
31.2.4	Kubernetes . . . . .	1171
31.2.5	Schwachstellen von Container-Technologien . . . . .	1172
31.2.6	Serverless Computing . . . . .	1173
31.2.7	Schwachstellen von Serverless Computing . . . . .	1174
31.2.8	Weitere Cloud-Dienstleistungen . . . . .	1174
31.3	Bedrohungen der Sicherheit und Integrität in der Cloud . . . . .	1174
31.3.1	Kontrollverlust . . . . .	1175
31.3.2	Unsichere Cloud-Infrastruktur . . . . .	1175
31.3.3	Missbrauchs-Risiken beim Cloud-Anbieter . . . . .	1177
31.3.4	Unsichere Kommunikation mit der Cloud . . . . .	1177
31.3.5	Unzureichende Zugangskontrolle . . . . .	1179
31.3.6	Cloud Computing für Hacker . . . . .	1180
31.3.7	Übersicht und Zusammenfassung . . . . .	1180
31.4	Angriffe auf Cloud-Infrastrukturen . . . . .	1181
31.4.1	Zugangsdaten ermitteln . . . . .	1181
31.4.2	Persistenten Zugang sichern . . . . .	1182
31.4.3	Malware einschleusen . . . . .	1182
31.4.4	Unsichere Voreinstellungen ausnutzen . . . . .	1183
31.4.5	Cryptojacking . . . . .	1183
31.4.6	Zugang über Federation Services . . . . .	1184
31.4.7	Angriffsvektor Webanwendung . . . . .	1185
31.5	Cloud-Security-Tools . . . . .	1185
31.5.1	Security-Tools des Cloud-Anbieters . . . . .	1185
31.5.2	Drittanbieter-Security-Software . . . . .	1185
31.5.3	Pentest-Simulation mit CloudGoat und Pacu . . . . .	1186
31.6	Zusammenfassung und Prüfungstipps . . . . .	1187
31.6.1	Zusammenfassung und Weiterführendes . . . . .	1187
31.6.2	CEH-Prüfungstipps . . . . .	1189
31.6.3	Fragen zur CEH-Prüfungsvorbereitung . . . . .	1189

<b>32</b>	<b>Durchführen von Penetrationstests</b>	1191
32.1	Begriffsbestimmung Penetrationstest	1191
32.1.1	Was bedeutet »Penetrationstest« eigentlich?	1192
32.1.2	Wozu einen Penetrationstest durchführen?	1192
32.1.3	Penetrationstest vs. Security Audit vs. Vulnerability Assessment	1193
32.1.4	Arten des Penetrationstests	1194
32.2	Rechtliche Bestimmungen	1195
32.2.1	In Deutschland geltendes Recht	1196
32.2.2	US-amerikanisches und internationales Recht	1197
32.3	Vorbereitung und praktische Durchführung des Penetrationstests	1199
32.3.1	Die Beauftragung	1199
32.3.2	Methodik der Durchführung	1201
32.3.3	Praxistipps	1203
32.4	Der Pентest-Report	1206
32.4.1	Dokumentation während des Pentests	1206
32.4.2	Was umfasst der Pентest-Report?	1207
32.4.3	Aufbau des Pентest-Reports	1208
32.5	Abschluss und Weiterführendes	1210
32.5.1	Das Abschluss-Meeting	1211
32.5.2	Weiterführende Tätigkeiten	1211
32.6	Zusammenfassung und Prüfungstipps	1211
32.6.1	Zusammenfassung und Weiterführendes	1211
32.6.2	CEH-Prüfungstipps	1212
32.6.3	Fragen zur CEH-Prüfungsvorbereitung	1213
<b>A</b>	<b>Lösungen</b>	1215
	<b>Stichwortverzeichnis</b>	1229

# Einleitung

Sie suchen nach einem strukturierten, umfassenden Praxishandbuch zum Thema »Ethical Hacking und Penetration Testing«? Prima, dann sind Sie hier genau richtig! In diesem Buch lernen Sie die Vorgehensweisen und Techniken professioneller Hacker und Penetration-Tester kennen und erlernen das Handwerk von der Pike auf. Durch viele Schritt-für-Schritt-Anleitungen, die Sie selbst in Ihrem Hacking-Labor nachvollziehen können, erleben Sie die Hacking-Techniken quasi live und in der Praxis. Hier ist Mitmachen angesagt!

Dieses Buch versteht sich zum einen als Praxisleitfaden für einen fundierten Einstieg in die Welt der Hacker und Penetration-Tester. Zum anderen sind die Inhalte an das Curriculum des Certified-Ethical-Hacker-Examens (CEHv11) des EC-Council angelehnt, sodass Sie dieses Werk als zusätzliche Ressource für die Prüfungsvorbereitung nutzen können. Bitte beachten Sie hierzu, dass es bestimmte Voraussetzungen für die Prüfungszulassung gibt, die wir Ihnen im ersten Kapitel erläutern.

Das CEH-Examen unterliegt ständigen Aktualisierungen, die naturgemäß nicht im bereits gedruckten Buch berücksichtigt werden können. Im Buch-Memberbereich auf [www.hacking-akademie.de/buch/member](http://www.hacking-akademie.de/buch/member) versuchen wir aber, immer zeitnah aktualisierte Informationen bereitzustellen. Die Zugangsdaten zum Memberbereich finden Sie am Ende dieser Einleitung.

## Für wen ist dieses Buch geeignet?

Dieses Buch ist für Sie geeignet, wenn Sie sich praxisorientiert und umfassend mit den Themen Hacking und Penetration Testing beschäftigen möchten. Die Zielgruppe umfasst insbesondere:

- Angehende Ethical Hacker und Penetration-Tester
- System- und Netzwerkadministratoren mit Fokus auf IT-Sicherheit
- Verantwortliche im Bereich IT-Security
- Interessierte Power-User

Auch wenn Sie sich durch einfaches Durchlesen des Buches bereits einen guten Überblick über das Thema verschaffen können, ist der Inhalt eher dazu konzipiert, tief in die Materie einzutauchen, und fordert Sie mit konkreten praktischen Beispielen zum Mitmachen auf. Dies erfordert bei Ihnen auf diesem Level auch ein ordentliches Maß an Engagement und Eigeninitiative. Aber genau so lernen Sie die Methoden nicht nur in der Theorie, sondern direkt in der praktischen Umsetzung.

Die Inhalte bauen an einigen Stellen aufeinander auf, sodass das Buch für ein umfassendes Verständnis Kapitel für Kapitel durchgearbeitet werden sollte. Natürlich eignet es sich darüber hinaus auch als Nachschlagewerk, da zu allen Inhalten, die für das Verständnis eines Themas benötigt werden, entsprechende Verweise zu den jeweiligen Stellen im Buch vorhanden sind.

## Für wen ist dieses Buch nicht geeignet?

Auch wenn Sie in diesem Buch sehr viele Hacking-Tools kennenlernen werden, so möchten wir an dieser Stelle doch klar betonen, dass das Buch nicht für Scriptkiddies gedacht ist, die mit ein paar wenigen Klicks coole Hacks zaubern und ihre Freunde beeindrucken wollen. Leser, die ohne viel Hintergrundwissen und Engagement ein paar oberflächliche Tricks lernen wollen, finden sicher andere Literatur interessanter.

Andersherum geht es hier auch nicht darum, versierten Profis, die bereits tief in den Themen stecken, den letzten Schliff zu geben. Zu jedem Thema, das das Buch aufgreift, lassen sich eigene Bücher schreiben. Auch wenn die Seitenzahl sehr groß ist, können wir zu vielen Themen nicht mehr als einen fundierten, praxisnahen Einstieg bieten.

## Was werden Sie hier lernen?

In diesem Buch geht es um Ethical Hacking und Penetration Testing. Wir werden diese Begriffe noch detaillierter beschreiben. Vom Grundsatz handelt es sich darum, die Perspektive des Angreifers einzunehmen, um die Schwachstellen von Computersystemen und -netzwerken aufzudecken. Dabei haben wir unter dem Strich das Ziel, die IT-Systeme sicherer zu machen. Es geht also nicht darum, die gefundenen Schwachstellen für die eigene Bereicherung zu nutzen, sondern darum, dem Auftraggeber die Möglichkeit zu geben, diese zu beseitigen. Anders ausgedrückt, bilden wir Sie hier zu einem »gutartigen« Hacker aus. Die Vorgehensweise, Technologien und eingesetzten Tools sind jedoch weitgehend dieselben, wie sie von bösartigen Hackern verwendet werden. Diese lernen Sie damit also ebenfalls kennen. Es ist wie so oft: Nicht die Werkzeuge bestimmen darüber, ob sie etwas verbessern oder Schaden anrichten, sondern derjenige, der sich diese Werkzeuge zunutze macht und einsetzt.

Hacking ist einerseits sehr kreativ und individuell, andererseits gibt es aber auch eine sinnvolle Vorgehensweise mit verschiedenen Phasen, die in fast jedem professionellen Hacking-Angriff enthalten sind. Sie erfahren, welche das sind und wie die einzelnen Phasen ablaufen. Viele Hacking-Tätigkeiten bauen aufeinander auf, andere kommen nur in bestimmten Szenarien zum Tragen. Wir haben in diesem Buch fast alle relevanten und gängigen Bereiche abgedeckt: angefangen vom simplen Passwort-Hacking über diverse Web-Hacking-Szenarien bis hin zu Mobile- und IoT-Hacking. Für alle Angriffsformen werden effektive Verteidigungsmaßnahmen aufgelistet, so dass Sie Ihre Kunden dabei unterstützen können, die gefundenen Schwachstellen zu beheben.

Der Fokus in diesem Buch liegt allerdings auf den Angriffstechniken. Sie erhalten zum einen fundierte Hintergrundinformationen zur Vorgehensweise und zu den Hacking-Techniken und zum anderen viele Praxisszenarien, in denen Sie Ihr neues Wissen praktisch einsetzen können. Nachdem Sie dieses Buch durchgearbeitet und die Szenarien praktisch nachvollzogen haben, sind Sie auf dem besten Weg zu einem fähigen Ethical Hacker und Penetration-Tester. Im Anschluss sind Sie in der Lage, Ihre Fähigkeiten eigenständig weiterzuentwickeln und mit zusätzlichen Informationsquellen Ihr Know-how zu vertiefen. Zudem erhalten Sie eine wertvolle Ressource für die Vorbereitung auf das CEHv11-Examen, mit dem Sie Ihre Karriere als Ethical Hacker effektiv voranbringen können.

# Inhaltsübersicht

Das Buch ist in sechs Teile untergliedert. Nachfolgend stellen wir Ihnen den Inhalt kurz vor, damit Sie sich ein Bild verschaffen können.

## Teil I – Grundlagen und Arbeitsumgebung

Hier erfahren Sie zunächst in Kapitel 1, welche Hacker-Typen es gibt und welche Ziele diese verfolgen. Wichtig ist dabei auch der rechtliche Aspekt, den wir natürlich ebenfalls betrachten. In Kapitel 2 bauen wir gemeinsam die Arbeitsumgebung für unser Hacking-Labor auf, das Sie im Laufe des gesamten Buches nutzen können. In Kapitel 3 lernen Sie Ihr wichtigstes Arbeitsgerät namens Kali Linux kennen.

Kapitel 4 widmet sich der Anonymität im Internet und der Methoden, deren sich die Hacker bedienen, um anonym zu bleiben. In Kapitel 5 betrachten wir mit der Kryptografie eines der wichtigsten Konzepte im Rahmen der IT-Sicherheit, wobei kryptografische Systeme in der Praxis auch immer wieder Angriffen ausgesetzt sind.

## Teil II – Informationsbeschaffung

Im zweiten Teil beschäftigen wir uns mit der Informationsbeschaffung. Zunächst lernen Sie in Kapitel 6 die passive Datensammlung. In Kapitel 7 nehmen wir das Netzwerk unter die Lupe mithilfe von Netzwerk-Scannern wie z.B. Nmap. Kapitel 8 enthält Techniken und Wege für den Enumeration-Prozess, bei dem wir versuchen, aus verschiedenen Netzwerk-Diensten so viele Informationen zu extrahieren wie möglich.

Mit dem Vulnerability-Scanning in Kapitel 9 werden wir dann bereits aggressiver und suchen gezielt nach Schwachstellen. Die Schwachstellenanalyse behandeln wir ebenfalls in diesem Kapitel.

## Teil III – Systeme angreifen

Nun geht es daran, Systeme konkret zu hucken. Wir beginnen in Kapitel 10 mit dem klassischen Password-Hacking und betrachten diverse Wege, um an Login-Daten zu gelangen. Mit der Privilegien-Eskalation in Kapitel 11 zielen wir darauf ab, unserer Rechte zu erweitern, wenn wir einen nicht-privilegierten Zugang zu den Zielsystemen erlangt haben.

Die Kapitel 12 und 13 beschäftigen sich mit Malware. Zum einen lernen Sie, wie Malware Computersysteme angreift, und erfahren dabei auch, wie Sie selbst Trojaner und ähnliche bösartige Software erstellen können. Zum anderen betrachten wir die Malware-Analyse, also Wege, um Malware aufzuspüren und zu beseitigen.

In Kapitel 14 erfahren Sie, wie Sie mithilfe von Steganografie Dateien und Informationen unberichtet und versteckt transportieren können. Kapitel 15 befasst sich mit dem Verwischen von Spuren. Dies ist ein elementarer Bestandteil eines Hacking-Prozesses, wenn der Angreifer unentdeckt bleiben möchte.

## Teil IV – Netzwerk- und sonstige Angriffe

Der Übergang zu diesem Teil ist fließend. In Kapitel 16 schauen wir mit Wireshark & Co. hinter die Kulissen der Netzwerk-Kommunikation. Hier lernen Sie, wie Sie Passwörter und Login-Vorgänge mitschneiden und ganze Sessions analysieren können. Dies führt wie von selbst zu Kapitel 17, in dem es um Lauschangriffe und Man-in-the-Middle-Angriffe geht.

Mit Session-Hijacking kann ein Angreifer eine etablierte und authentifizierte Session von ahnungslosen Benutzern übernehmen und spart sich so die Eingabe von Zugangsdaten. Wie das geht, erfahren Sie in Kapitel 18.

In Kapitel 19 lernen Sie die wichtigsten Security-Systeme kennen, denen sich ein Angreifer gegenübersieht. Hierzu gehören neben Firewalls insbesondere Intrusion-Detection- bzw. -Prevention-Systeme sowie Honeypots.

Den Abschluss dieses vierten Teils bilden drei eher anders geartete Angriffsmethoden. In Kapitel 20 werfen wir einen Blick hinter die Kulissen des Social Engineerings. Mit dieser Technik greifen wir nicht die Computersysteme selbst an, sondern bedienen uns psychologischer Tricks, um die Benutzer der Systeme auszutricksen und an Informationen zu gelangen. Kapitel 21 präsentiert Ihnen gängige Hacking-Hardware. Hier lernen Sie zum Beispiel, wie Sie einen Keylogger installieren oder ein Hacking-Kit für die Hosentasche auf einem Raspberry Pi einrichten können. Last, but not least beschäftigen wir uns in Kapitel 22 mit DoS- und DDoS-Angriffen. Diese destruktive Angriffsform ist im Internet weit verbreitet und kann auch im Rahmen von größer angelegten Angriffen nützlich sein, um bestimmte Systeme außer Gefecht zu setzen, die den Angriff evtl. verhindern könnten.

## Teil V – Web-Hacking

Einer der wichtigsten Angriffsvektoren ist der Angriff auf Webanwendungen. Daher haben wir diesem Thema einen breiten Raum eingeräumt. In Kapitel 23 lernen Sie zunächst die Grundlagen der Web-Kommunikation und -Technologien und erfahren, wie Angriffe auf Webserver und -anwendungen grundsätzlich funktionieren.

Kapitel 24 führt Sie in die Welt der *OWASP Top 10* ein, OWASP steht für *Open Web Application Security Project*. Dabei handelt es sich um die zehn gängigsten Angriffsvektoren auf Webanwendungen. In diesem Kapitel erfahren Sie die daraus resultierenden Angriffe in Theorie und Praxis. Kapitel 25 greift den wichtigsten Punkt der OWASP Top 10 heraus und betrachtet den Angriffsvektor SQL-Injection von allen Seiten. In Kapitel 26 ergänzen Sie Ihr Wissen zu Injection-Angriffen und wir betrachten weitere Formen wie Command-Injection, Code-Injection oder LFI und RFI.

Den Abschluss dieses Teils bildet eine sehr gängige Form des Angriffs auf Software, die zwar häufig bei Webanwendungen zum Einsatz kommt, aber nicht auf diese beschränkt ist. Die Rede ist von Buffer-Overflow-Angriffen, die Sie in Kapitel 27 kennenlernen. Dort gehen wir ein umfassendes Praxisbeispiel durch, sodass Sie Ihren eigenen Buffer-Overflow-Angriff durchführen können.

## Teil VI – Angriffe auf WLAN und Next-Gen-Technologien

Nun kommen wir zum letzten Teil des Buches, in dem wir uns zunächst mit der Thematik der mobilen Geräte beschäftigen. Im Kapitel 28 lernen Sie alles rund um WLAN-Hacking. Welchen Angriffsvektoren Smartphones und Tablets ausgesetzt sind, erfahren Sie in Kapitel 29. Kapitel 30 führt Sie in die Welt des IoT-Hackings ein, das immer wichtiger wird, da das Internet of Things seinen Siegeszug unaufhaltsam fortsetzt und die internetfähigen Alltagsgegenstände oft angreifbar sind. Mit dem Thema Cloud-Security schließen wir das Themenspektrum dieses Buches in Kapitel 31 ab.

An dieser Stelle haben Sie ein fundiertes Verständnis für Hacking-Methoden und -Technologien sowie für gängige Hacking-Tools. Zudem haben Sie zu allen Angriffsmethoden und -vektoren die effektivsten Gegenmaßnahmen kennengelernt und sind in der Lage, Kunden bzw. Auftraggeber hinsichtlich der Absicherung ihrer Systeme fundiert zu beraten.

Um dieser Tätigkeit einen Rahmen zu geben, existieren Penetrationstests. Das letzte Kapitel dieses Buches erläutert detailliert die Vorgehensweise bei einem Penetrationstest und gibt viele Tipps und Hinweise für angehende Penetration-Tester.

## Aktualität der Inhalte

Als wir dieses Buch vor über vier Jahren begonnen hatten, war uns nicht einmal im Ansatz klar, auf was wir uns einlassen würden! Es sollte unser bisher umfangreichstes Buchprojekt werden, da der Inhalt ständigen Änderungen und Anpassungen unterworfen ist. Als wir das Buch inhaltlich einmal fertiggestellt hatten, konnten wir sozusagen von vorn anfangen und mussten viele Stellen überarbeiten, vieles ergänzen und einiges streichen, da es keine Gültigkeit mehr hatte. Fast die Hälfte des Buches wurde in der Zwischenzeit inhaltlich überarbeitet, um es an den aktuellen Stand anzupassen.

Mittlerweile wurde das Buch für die 2. Auflage erneut an vielen Stellen überarbeitet, um es für die aktuelle Zertifizierung zum CEHv11 zu aktualisieren. Und auch hier mussten wir an diversen Stellen veraltete Tools und Beschreibungen anpassen.

Aufgrund dieser Erfahrung haben wir einen wichtigen Hinweis an Sie als Leser: Wir haben viel Herzblut in dieses Buch investiert. Alle Anleitungen wurden mit größtmöglicher Sorgfalt erstellt und mehrfach getestet. Leider können die Anleitungen jedoch immer nur den Stand zum Zeitpunkt der Erstellung darstellen. Programme, Webseiten und Prozesse unterliegen in der IT-Welt ständiger Weiterentwicklung und Veränderung. Daher kann und wird es passieren, dass vereinzelt Programme nicht mehr so funktionieren wie beschrieben, Webseiten anders aussehen als im Buch abgedruckt und Inhalte unter Umständen nicht mehr in der Form zur Verfügung stehen wie beschrieben. Wir bitten hierfür um Verständnis und motivieren Sie, in derartigen Fällen selbstständig nach Lösungen zu suchen.

Denn das ist Hacking: neue Wege gehen, Dinge anders machen, um zu neuen Ergebnissen zu gelangen. Hacking erfordert Kreativität, Neugier und eine gute Portion Beharrlichkeit, da Hacker die Computersysteme und Software nicht in der vom Hersteller oder Entwickler erwarteten Art und Weise nutzen und daher mit dem Unerwarteten umgehen müssen.

## Die Webseite zum Buch

Obwohl dieses Buch bereits sehr umfangreich ist, mussten wir aus Platzgründen diverse Inhalte auslagern. An vielen Stellen im Buch verweisen wir auf die jeweiligen Dokumente mit ergänzenden Informationen, die unter [www.hacking-akademie.de/buch/member](http://www.hacking-akademie.de/buch/member) verfügbar sind. Sie stehen exklusiv für Sie als Leser zur Verfügung und sind zugangsgeschützt. Geben Sie das Passwort **h4ckm3mber** ein, um in den Buch-Memberbereich zu gelangen und hier auf alle zusätzlichen Inhalte zugreifen zu können. In diesem Zusammenhang stellen wir auch eine Errata-Seite bereit, in der alle bekannten Fehler bzw. Updates zu den Inhalten erfasst sind. Falls Sie Fehler melden oder anderweitiges Feedback geben wollen, freuen wir uns darüber. Dies können Sie an [buch@hacking-akademie.de](mailto:buch@hacking-akademie.de) schicken.

Noch ein Hinweis zur Online-Learning-Plattform Hacking-Akademie: Hier bieten wir als Ergänzung zum Buch eine umfassende Ausbildung zum Ethical Hacker und Penetration-Tester an. Mit praxisorientierten Videolektionen und eigener Laborumgebung erhalten Sie hier die Möglichkeit, Ihre Hacking- und Security-Skills systematisch auf- und auszubauen.

## Worauf warten Sie noch?

Jetzt liegt es an Ihnen! Haben Sie das Zeug zu einem fähigen Hacker? Sie benötigen ein hohes Maß an Motivation und Neugier, Disziplin und Geduld. Hacking lernt man nicht von heute auf morgen. Hacking umfasst grundsätzlich die gesamte Palette der IT-Systeme und -Anwendungen.

Wer hier jenseits des Scriptkiddie-Niveaus erfolgreich sein möchte, beschreitet einen langen, spannenden Weg, auf dem er sehr viel lernen, aber auch immer wieder an seine Grenzen stoßen wird. Wir freuen uns, wenn wir Sie bei Ihrem Einstieg in die spannende Welt des Hackings und Penetration Testings ein Stück weit begleiten und unterstützen können.

Jetzt bleibt nur eins: Gehen Sie den ersten Schritt, beginnen Sie Ihren Weg! Bauen Sie noch heute Ihr Hacking-Labor auf und starten Sie Ihre Karriere als Ethical Hacker!

Herzliche Grüße,  
Eric Amberg und Daniel Schmid

# Über die Autoren



**Eric Amberg** ist selbstständiger Experte für IT-Netzwerke und -Sicherheit und hat in den letzten 20 Jahren zahlreiche Projekte aller Größenordnungen durchgeführt. Seine große Leidenschaft ist die Wissensvermittlung, die er in Büchern, Magazinen und insbesondere Videotrainings stets praxisnah und lebendig präsentiert. Eric verfügt über diverse Zertifizierungen, unter anderem CEH, CISSP, CCNP Security, LPIC-2 und ist zertifizierter Cisco-Trainer (CSI # 34318).



**Daniel Schmid** ist bei einem großen Energiekonzern im Bereich Netzwerke und Security tätig. Als Projektleiter für diverse große, teils internationale Projekte hat er in über 10 Jahren viel Erfahrung in der Planung und Implementation sicherheitskritischer Infrastruktur gesammelt und hat dabei seine Leidenschaft für das Thema »Hacking und Penetration Testing« entdeckt.

Eric und Daniel haben bereits viele gemeinsame Projekte erfolgreich umgesetzt und sind die Gründer der Hacking-Akademie ([hacking-akademie.de](https://hacking-akademie.de)).

# Danksagung

Dieses Buch war ein echtes Mammut-Projekt, das ohne die Unterstützung von vielen Menschen nicht zu diesem bemerkenswerten Ergebnis geführt hätte. Daher möchten sich die Autoren Eric und Daniel bei allen Beteiligten herzlich für den großartigen Einsatz und die fantastische Unterstützung bedanken.

Unser besonderer Dank gilt unseren unermüdlichen Testlesern Anton Perchermeier, Martin Meirl, Markus Bauer und Timo Scheidemantel. Mit euren umfassenden, kritischen und fundierten Rückmeldungen habt ihr die hohe Qualität dieses Buchs erst ermöglicht. Wir schätzen uns glücklich, Profis aus dem IT-Security-Umfeld wie euch als engagierte Testleser zu haben. Dank euch ist der Inhalt des Buchs noch einmal deutlich aufgewertet worden.

Auch an Sabine Schulz vom mitp-Verlag geht ein herzliches Dankeschön! Liebe Sabine, Du hast während der langen Entstehungszeit dieses Buchs stets zu uns gehalten und trotz vieler Verzögerungen immer mit Verständnis reagiert – das ist alles andere als selbstverständlich, hat aber auch dazu beigetragen, dass wir uns noch mehr Mühe mit dem Buch gegeben haben, damit sich die Wartezeit auch wirklich gelohnt hat.

Man sagt, hinter jedem erfolgreichen Mann steht eine starke Frau. Ob der Spruch allgemein noch zeitgemäß ist, sei dahingestellt – auf uns trifft er auf jeden Fall zu. Ohne dass unsere Partnerinnen uns den Rücken freigehalten hätten und sehr tolerant mit der vielen Zeit umgegangen wären, in der wir am Buch-Manuskript gesessen haben, wäre dieses Buchprojekt nicht realisierbar gewesen. Unser ganz besonderer Dank gilt daher unseren Ehefrauen Kati und Rocío. Ihr habt uns dabei so großartig unterstützt und mit viel Verständnis und Geduld in den letzten Jahren auf die zusätzliche Arbeitslast reagiert, die uns das Buch auferlegt hat. Nur mit Eurer Hilfe konnte dieses Buch entstehen!

Berlin und Stuttgart, 30. November 2021

Eric und Daniel

# Grundlagen Hacking und Penetration Testing

Hacker sind die Bösen! Hacker sind darauf aus, möglichst viel Schaden anzurichten und bedrohen das Internet und jeden Rechner, der daran angeschlossen ist! Also gilt es, Hackern möglichst schnell und nachhaltig das Handwerk zu legen ...

Okay, Schluss damit! Die obige Aussage ist natürlich Unsinn! Tatsache ist, dass wir Hackern diverse geniale Programme und Tools verdanken. Kennen Sie Linux? Nun, wer nicht? Wissen Sie, wer es entwickelt hat? Linus Torvalds, ein finnischer Student, der sich nicht damit abfinden wollte, dass AT&T den Quellcode zu UNIX nicht freigeben wollte und ein System benötigte, das besser auf seine Anforderungen zugeschnitten war. Daraus entstand Linux (Linus+X). Und auch wenn die meisten »Rechtschaffenen« unter uns Torvalds einen »Entwickler« nennen würden, so versteht er sich selbst doch als »Hacker«.

Es gibt also jede Menge Begrifflichkeiten zu unterscheiden. In diesem Kapitel legen wir die Grundlagen für Ihr Verständnis von Hacking und Penetration Testing. Sie lernen insbesondere Folgendes:

- Was ist Hacking?
- Verschiedene Hacker-Typen
- Motive und Absichten eines Hackers
- Was bedeutet Ethical Hacking?
- Die Zertifizierung zum Ethical Hacker (CEH)
- Die Schutzziele
- Wie funktioniert ein Penetrationstest?
- Hacking-Beispiele

In diesem ersten Kapitel beschäftigen wir uns mit den Grundlagen des Hackings. Damit Sie verstehen, was ein Hacker überhaupt ist und wo das Wort Hacking herkommt. Sie werden zudem erfahren, welche verschiedenen Hacker-Typen es gibt und wie die Ziele der Hacker aussehen. Sie lernen, was sich hinter dem *Ethical Hacking* verbirgt und warum Sie sich diesen Ehrencodex zu Eigen machen sollten.

Darüber hinaus betrachten wir auch die andere Seite. Die Schutzziele geben Aufschluss darüber, gegen welche Gefahren wir uns schützen wollen. Letztlich geht es darum, Computersysteme und -netzwerke sicherer zu machen. Der Weg ist also das Hacking, das Ziel jedoch, die IT-Sicherheit zu erhöhen. Daher werden wir ein großes Augenmerk auf den Schutz der gefundenen Schwachstellen und Angriffsvektoren legen.

Ein *Ethical Hacker* betreibt seine Tätigkeit regelmäßig im Rahmen eines beauftragten Penetrations- tests. Sie lernen, wie ein solcher Test aufgebaut ist, welchen Klärungsbedarf es mit dem Auftraggeber gibt und wie ein Hacker bzw. Penetrationstester vorgeht.

Den Abschluss dieses Kapitels liefern einige bekannte Hacking-Beispiele, die Ihnen schon einmal einen gewissen Bezug zur Realität zeigen. Im Laufe dieses Buches lernen Sie noch viele weitere Möglichkeiten kennen, wie Computersysteme angegriffen werden können. Dabei gehen wir auch immer wieder auf bereits bekannte Angriffe ein und beschreiben diese.

## 1.1 Was ist Hacking?

In der heutigen Zeit von Informationstechnologien und Vernetzung spricht man von einem »Hacker«, wenn es um eine Person geht, die sich Zugriffe zu Netzwerken, Systemen und Anwendungen verschafft. Ohne dass der Besitzer der jeweiligen Einrichtungen das beabsichtigt hat. Doch das war nicht schon immer so.

Wo kommt denn dieses Wort überhaupt her und was ist denn Hacking eigentlich? Der Begriff »Hacking« kommt aus einer Zeit, in der nicht Netzwerke und Computersysteme im Fokus standen. Denn damit hatte der Begriff erst mal gar nichts zu tun. Es ging vielmehr darum, sich so intensiv mit einer bestimmten Technik zu beschäftigen, dass man einen Weg findet, scheinbar Unmögliches machbar zu machen. Auf Deutsch hätte man das Wort »Tüftler« verwendet.

Ein Hacker war jemand, der mithilfe von ein paar Streichhölzern, einem Gummi und einem Bleistift einen Fernseher bauen kann. Oder war das MacGyver? :-) Spaß beiseite. Tatsächlich war ein Hacker ursprünglich einfach nur jemand, der sich sehr intensiv mit einer Technologie auseinandergesetzt hat, um sie zu begreifen, für sich nutzbar zu machen und ggf. zu verbessern. Ein Hacker ist nichts Bedrohliches oder Böses an sich. Dieser Ruf kam erst später durch die Medien und als es die ersten Einbrüche in fremde Systeme gab. Heutzutage hat ein Hacker in der Öffentlichkeit kein gutes Ansehen, man verbindet den Begriff in der Regel mit einem Verbrecher, der gegen das Gesetz handelt. Doch das stimmt so nicht zwangsläufig.

Aber wie kommt denn nun dieses Bild vom Hacker, der in fremde Computersysteme eindringt und allerlei Schaden anrichtet, zustande? Nun, zweifelsfrei haben Hacker eines gemeinsam: Sie sind neugierige Menschen, die neue Wege suchen, insbesondere mit Computersystemen zu arbeiten! Und einige von ihnen sind scharf auf Informationen. Dabei ist es zunächst einmal zweitrangig, ob ein Computersystem diese Informationen freiwillig bereitstellt oder nicht. Im Gegenteil versprechen gut geschützte Computer und Netzwerke sogar interessantere Informationen – proportional steigend zu den Schutzmaßnahmen.

Und so waren es natürlich auch gerade die Hacker mit ihrem tiefgreifenden Wissen über Computersysteme und -netzwerke, die, oftmals aus purer Neugier, Wege in diese Systeme gesucht und gefunden haben. In vielen Fällen wurden die gefundenen Schwachstellen dem jeweiligen Eigentümer bekannt gemacht und die möglicherweise gefundenen Daten und Informationen gar nicht verwendet – es ging nur um die Machbarkeit eines Einbruchs.

Aber wie es so ist, nutzen nicht alle ihr außerordentliches Wissen, um Gutes zu tun, diese Welt sicherer zu machen oder interessante Software unentgeltlich zur Verfügung zu stellen. Stattdessen unterliegen sie der Verlockung, ihr Expertenwissen für sich selbst zu nutzen, um sich zu bereichern.

Und genau hier grenzen sich die einzelnen Hacker-Typen voneinander ab. Denn der traditionelle Hacker im oben beschriebenen Sinne möchte keinesfalls in einen Topf mit diesen Kriminellen geworfen werden. Daher wird der »böse« Hacker auch generell als »Cracker« bezeichnet. Doch dies ist nur eine sehr globale Kategorisierung. Für eine fundierte Unterscheidung derjenigen, die sich mit dem Thema »Hacking« intensiver beschäftigen, müssen wir etwas weiter in die Tiefe gehen und neben der Motivation auch die Qualität der Tätigkeit betrachten.

## 1.2 Die verschiedenen Hacker-Typen

Bestimmt kennen Sie aus diversen Blockbustern die schwarzen Gestalten, die hinter einer Wand von Bildschirmen sitzen und nur von den kryptischen, grünen Zeichen beleuchtet werden, die über die Monitore rasen. Auch wenn dieses gängige Klischee tatsächlich durchaus vereinzelt bedient wird und einige Zeitgenossen auf diese Art arbeiten, gibt es doch auch ganz andere Inkarnationen der Hacker-Zunft.

Es finden sich nämlich genauso Hacker, die mit Anzug und Krawatte bei namhaften Firmen ein- und ausgehen, um deren Sicherheit zu testen. Diese Leute haben auch eine Hacking-Ausbildung, nutzen ihr Wissen allerdings nicht, um Schaden anzurichten, sondern um genau davor zu schützen – man nennt sie auch Penetrationstester bzw. kurz: Pentester. Tatsächlich gibt es aber auch böse Jungs, die Anzug und Krawatte tragen. In bestimmten Situationen gilt: Kleider machen Leute. Und wer z.B. in einer Bank ein Computer-Terminal haken möchte, tut gut daran, optisch nicht aufzufallen. Auch für das *Social Engineering*, bei dem Informationen über Menschen anstatt über Technik gewonnen werden, ist das Auftreten oft ein wichtiger Aspekt. Näheres hierzu finden Sie in Kapitel 20 *Social Engineering*.

Nachfolgend eine Übersicht über die wichtigsten Hacker-Klassifikationen.

### Scriptkiddies

Sie haben wenig Grundwissen und versuchen, mithilfe von Tools in fremde Systeme einzudringen. Dabei sind diese Tools meist sehr einfach über eine Oberfläche zu bedienen. Die Motivation ist meistens Spaß und die Absichten sind oft krimineller Natur. Oftmals möchten Scriptkiddies mit ihren Aktionen Unruhe stiften. Die Angriffe sind meist ohne System und Strategie. Viele Hacker starten ihre Karriere als Scriptkiddie, nutzen die Tools zunächst mit wenig Erfahrung, lernen aus dem Probieren, entwickeln sich weiter und finden dadurch einen Einstieg in die Szene.

### Black Hats

Diese Gattung Hacker beschreibt am ehesten die Hacker, die man aus den Medien kennt. Hier redet man von Hackern mit bösen Absichten. Sie haben sehr gute Kenntnisse und greifen bewusst und strukturiert Unternehmen, Organisationen oder Einzelpersonen an, um diesen Schaden zuzufügen. Die Ziele der Black Hats sind vielfältig und reichen vom einfachen Zerstören von Daten bis hin zum Diebstahl von wertvollen Informationen, wie Kontodaten oder Unternehmensgeheimnissen. In manchen Fällen reicht es den Black Hats auch, wenn sie erfolgreich die Server ihres Opfers lahmlegen und damit Sabotage verüben.

### White Hats

Einen *White Hat Hacker* nennt man oft auch einen *Ethical Hacker*. Er nutzt das Wissen und die Tools eines Hackers, um zu verstehen, wie Black Hats bei ihren Angriffen vorgehen. Im Gegensatz zum Black Hat will der White Hat jedoch die betreffenden Systeme letztlich vor Angriffen besser schützen und testet daher die Schwachstellen aktiv aus. Damit hat ein White Hat Hacker grundsätzlich keine bösen Absichten, im Gegenteil, er unterstützt die Security-Verantwortlichen der jeweiligen Organisation. White Hat Hacker oder Ethical Hacker versuchen im Anschluss an ihre Hacking-Tätigkeit, herauszufinden, welche Sicherheitslücken es gibt, und geben eine Anleitung dazu, diese möglichst effizient zu schließen.

## **Penetrationstester (Pentester)**

Zu den White Hat Hackern gehören auch die sogenannten Penetrationstester. Hier steht grundsätzlich ein Auftrag im Hintergrund eines Angriffs. Pentester werden angeheuert, um ein bestimmtes System auf Herz und Nieren zu testen. Hier wird sehr systematisch nach Schwachstellen gesucht. Ein Penetrationstester hat eine ausdrückliche Genehmigung für sein Tun. Am Ende seiner Arbeit steht ein Bericht zur Verfügung, in dem alle gefundenen Schwachstellen dem Auftraggeber aufgezeigt werden. Dieser hat dann die Möglichkeit, die Lücken zu schließen, bevor die Black Hats ihr Glück versuchen ...

## **Grey Hats**

Genauso wie die Farbe Grau zwischen Schwarz und Weiß liegt, so liegen die Grey Hats zwischen den Black und den White Hat Hackern. Mal haben sie gute, mal schlechte Absichten. Je nachdem was ihnen gerade lukrativ erscheint. Ein Grey Hat ist nicht grundsätzlich böse, nimmt es mit der Ethik aber auch nicht unbedingt so genau.

## **Cyber-Terroristen**

Dies sind organisierte Gruppen, die sich gegen bestimmte Dinge auflehnen und mithilfe des Internets und seiner Technologien Angriffe durchführen. Dabei versuchen sie, möglichst viel Schaden anzurichten. In vielen Fällen ist ihr Tun politisch oder auch religiös motiviert.

## **Staatlich unterstützte Hacker**

Hierbei handelt es sich um Hacker, die im Auftrag einer Regierung agieren. Sie wurden speziell ausgebildet und versuchen, als Agenten beispielsweise an geheime Informationen zu kommen. Das Einsatzgebiet kann der Kampf gegen den Terror sein oder auch das Sammeln von Informationen über einen Gegner in Konfliktsituationen. Insbesondere die USA, Russland und China sind hier sehr aktiv.

## **Suicide Hacker**

Der CEH (Certified Ethical Hacker) beschreibt hier eine Ausprägung des Hackings, bei dem der Angreifer ohne Rücksicht auf Verluste vorgeht und dabei auch sich selbst der Gefahr aussetzt, entdeckt zu werden. Dabei handelt es sich ggf. nicht wirklich um Profis, sondern eher um Verzweiflungstäter, die jedoch aufgrund ihrer Kompromisslosigkeit kurzfristig hocheffektiv ihre Ziele erreichen können.

## **Hacktivisten**

Werden Systeme, insbesondere Webserver, im Internet gehackt, um auf politische Inhalte hinzuweisen und zu protestieren, sprechen wir von *Hacktivismus* oder *Hacktivisten*. Dabei werden in der Regel die originalen Webinhalte durch eigene Inhalte ersetzt. Diesen Prozess nennt man auch *defacen* (von engl. *Face* = Gesicht). Weitere Methoden der Hacktivisten sind *Denial-of-Service-Angriffe* und *E-Mail-Spamming*. Die bekannteste Hacktivist-Gruppe kennen Sie vielleicht sogar schon, die Rede ist von *Anonymous*.

Oft ist es nicht einfach, zwischen den verschiedenen Typen zu unterscheiden. Ein Black Hat Hacker kann genauso auch ab und zu ein Hacktivist sein und ein White Hat arbeitet oft auch als Penetrationstester. Wichtig ist, zu wissen, dass nicht alle Hacker dieselben Absichten haben und es Hacker mit unterschiedlichsten Motiven gibt. Gutes Stichwort ...

## 1.3 Motive und Absichten eines Hackers

Egal, ob White oder Black Hat Hacker: Die Tools, die Techniken, die Vorgehensweise und auch das Wissen ist annähernd dasselbe. Unterschieden wird darin, welche Motive und Absichten ein Hacker hat.

### 1.3.1 Das Motiv

Fragen Sie einen Hacker (oder Cracker) danach, könnten Sie typischerweise folgende Antworten erhalten:

#### Ich möchte mich an jemandem rächen!

Rache ist kein seltenes Motiv, ob es der alte Arbeitgeber ist, der einen entlassen hat, eine Firma, mit der man Probleme hatte, oder gar die/der Ex-Partnerin/Partner. Das Ziel des Hacking-Angriffs besteht darin, jemandem Schaden zuzufügen, dem man nicht wohlgesonnen ist.

#### Ich möchte damit Geld verdienen!

Wer das Hacking beherrscht, dem stehen viele Türen offen. Gute White Hat Hacker sind gefragt – egal, ob sie als Security-Spezialist um die Sicherheit eines Unternehmens bemüht sind oder großen Organisationen Penetrationstests anbieten. Das White Hat Hacking ist durchaus lukrativ. Aber auch Black Hat Hacker kommen an ihr Geld, meistens allerdings durch illegale Weise wie Erpressung oder Datendiebstahl. Im Zweifel werden sie für ihre Aktivitäten von anderen bezahlt, in deren Auftrag sie ein bestimmtes Ziel verfolgen.

#### Ich möchte Spaß haben!

Keine Frage, Hacking macht Spaß, das werden Sie noch früh genug merken. Diese Mischung von Nervenkitzel und Erfolgserlebnis nach einem gelungenen Angriff ist sehr reizvoll. Daher gibt es viele Menschen, die sich das Hacking zum Hobby gemacht haben, eben weil es Spaß macht. Auch hier kann die Waage zur einen oder zur anderen Seite ausschlagen: Entweder nutzen Sie Ihr Wissen, um anderen zu helfen oder ihnen zu schaden ...

#### Ich möchte jemanden ausspionieren!

Nicht gerade die feine Art, aber es finden sich immer wieder gute Gründe, um einen Menschen, ein Unternehmen oder eine Institution auszuspionieren. Den klassischen Job eines Privat-Detektivs übernimmt in diesem Fall der Hacker. Die umfangreichsten Informationen finden sich heutzutage nicht mehr in Aktenschränken, sondern auf den Festplatten der Computer einer Person oder Institution. Daher ist der Einsatz von Hacking-Methoden sehr vielversprechend, um an sensible Informationen zu gelangen.

#### Ich möchte etwas bewegen!

Auch Aktivismus ist oft ein Motiv zum Hacken – daher der bereits oben beschriebene Begriff *Hacktivismus*. Es gibt eine Vielzahl von Angriffen auf politische Parteien bzw. Länder, Bewegungen und Firmen. Man muss hierzu heutzutage nicht mehr auf die Straße gehen, der Protest kann auch virtuell stattfinden, wie wir bereits weiter oben dargelegt haben.

## Ich möchte im Mittelpunkt stehen!

Meldungen über Hacking-Angriffe sind aus den Medien kaum noch wegzudenken. Möchten Sie auch mal in der Zeitung stehen? Dazu ist nur ein richtiger Angriff an der richtigen Stelle notwendig. Natürlich wäre es nicht gut, wenn Sie Ihren Namen unter einem Fahndungsfoto stehen sehen. Meist verbergen sich Hacker daher hinter Pseudonymen oder Gruppen. Bekannte Hacking-Gruppen sind zum Beispiel *Anonymous*, *AntiSec* oder *LulzSec*.

### 1.3.2 Ziel des Angriffs

Warum ein Hacker einen Angriff ausführt, haben wir also geklärt; stellt sich noch die Frage, was er genau vorhat. Welche Absichten können also hinter einem Hacking-Angriff stecken? Betrachten wir die wichtigsten:

#### Datendiebstahl

Der Angreifer ist auf geheime Daten seiner Opfer aus, er möchte an Informationen kommen. Daher geht er gezielt auf die Suche nach bestimmten Dateien oder Datensätzen. Die Daten können dann gewinnbringend weiterverkauft, gegen das Opfer verwendet oder erst gegen ein Lösegeld wieder freigegeben werden.

#### Manipulation

Auch hier sucht der Angreifer nach Daten, aber nicht, um diese an sich zu bringen, sondern um sie zu verändern. Das kann insbesondere bei finanziellen Transaktionen teilweise gravierende Folgen haben. Stellen Sie sich einmal vor, das Komma auf Ihrem monatlichen Gehaltszettel wäre um eine Stelle nach rechts verschoben ... und nun stellen Sie sich Ihren Arbeitgeber vor. Wo es Gewinner gibt, existieren immer auch Verlierer!

#### Erpressung

Mit gestohlenen oder manipulierten Daten kann der Angreifer das Opfer natürlich auch erpressen: Zahlt der Betroffene nicht die geforderte Summe, so werden z.B. Firmen-Interna veröffentlicht oder ein zentrales System lahmgelegt.

Eine Variante hierzu ist der Einsatz von *Ransomware*. Dabei werden die Daten des Opfers verschlüsselt und der Schlüssel nur gegen Zahlung eines Geldbetrags (engl. Ransom) übermittelt.

#### Rechte erweitern

In den meisten Fällen steckt dahinter die Absicht, den Angriff effektiv fortzuführen. Es wird versucht, an möglichst viele Rechte und Privilegien zu gelangen, um damit eine möglichst umfassende Kontrolle über das Zielsystem zu bekommen. Stellen Sie sich vor, Sie melden sich als normaler Benutzer an einem System an und erlangen durch Hacking-Methoden Administrator-Privilegien. Von diesem Moment an stehen Ihnen alle Türen offen, sodass Sie z.B. neue Software installieren oder die Systemkonfiguration ändern können. Somit ist die Rechte-Erweiterung (auch als *Privilegien-Eskalation* bzw. gängiger *Privilege Escalation* bekannt) selten Selbstzweck, sondern in der Regel Mittel zum Zweck.

## Unerlaubt etwas steuern

Viele Systeme haben die Aufgabe, etwas zu steuern. Denken Sie hierbei an Verkehrsleitrechner, Sicherheitszentralen, Maschinensteuerungen usw. Hat man sich einmal in die Sicherheitszentrale eingehackt, spart man sich das Brecheisen. Ist es z.B. einem Hacker möglich, sich in die Kontrollsysteme eines Kernkraftwerks zu hacken, kann das fatale Folgen bis hin zum Super-GAU haben. Sie halten das für weit hergeholt? Dann warten Sie mal ab, bis Sie die perfiden Methoden von *Stuxnet* kennengelernt haben, einer Wurmsoftware, die wir Ihnen in Abschnitt 1.8.2 dieses Kapitels vorstellen.

## Geld stehlen

Viele Angriffe finden auch auf Banken und Geldautomaten statt. Das Ziel der Begierde ist der schnöde Mammon – also Geld. Mal ehrlich: Haben Sie nicht auch schon davon geträumt, einen Geldautomaten so zu manipulieren, dass er unbegrenzt Geld ausspuckt? Wir zeigen Ihnen ... NICHT, wie es geht! Aber es gibt Techniken und Methoden, um sich zu bereichern, auch ohne den Bankautomaten aus dem Fundament zu reißen. In einigen Fällen werden Bankautomaten mit veralteter (und damit anfälliger) Software, wie z.B. Windows XP betrieben. Über Remote-Zugriff ist es möglich, entsprechende Schadsoftware zu installieren, um damit die Bankautomaten zu manipulieren.

Darüber hinaus ist es natürlich auch durch die Manipulation von Kontenbewegungen und Finanzsoftware möglich, Geld auf das eigene Konto auf den Bahamas transferieren zu lassen. Wie Sie feststellen, ist dieses Hacking-Ziel in der Regel durch Manipulation zu erreichen, die wir weiter oben bereits grundlegend als übergeordnetes Hacking-Ziel ausgemacht haben.

## Ruf ruinieren

Wie Sie schon wissen, können die Motive für Hacking auch Rache oder Aktivismus sein. Die Absicht, einen Ruf zu ruinieren, kann auf verschiedene Art und Weise umgesetzt werden. Eine Möglichkeit besteht darin, einen erfolgreichen Angriff bekannt werden zu lassen. Stellen Sie sich z.B. vor, in den Medien wird von einem erfolgreichen Hacking-Angriff auf eine Bank berichtet. Das richtet großen Image-Schaden an.

## Zugang/Service blockieren

Eine der häufigsten Angriffsformen ist der *Denial-of-Service-Angriff* (DoS). Dabei versucht der Angreifer, das Opfer-System oder -Netzwerk derartig zu überlasten, dass der angebotene Dienst (in der Regel Webanwendungen) nicht mehr für reguläre Anfragen oder Zugriffe erreichbar ist. DoS-Angriffe kommen in ganz verschiedenen Varianten vor. Im Internet wird häufig ein *Distributed-Denial-of-Service-Angriff* (DDoS) durchgeführt, wobei Hunderte oder sogar Tausende Systeme zentral gesteuert werden und synchronisiert einen Angriff starten (sogenannte Botnetze).

## 1.4 Ethical Hacking

Sie lernen in diesem Buch eine ganze Menge über das Hacking. Dieses Wissen können Sie für die verschiedensten Zwecke einsetzen. An dieser Stelle möchten wir jedoch noch einmal ganz ausdrücklich an Ihren ethischen Kompass appellieren!

## Was du nicht willst, das man dir tu' ...

Das Ziel dieses Buches ist *offensive IT-Sicherheit*. Das bedeutet, dass Sie als jemand, der sich mit den Methoden und Techniken der bösen Jungs (und Mädels) auskennt, Ihr Wissen nutzen, um die Sicherheit von Computersystemen zu erhöhen, indem Sie deren Schwachstellen aufdecken und helfen, diese zu beseitigen. Dies wird als *Ethical Hacking* bezeichnet. Es dient ausschließlich der Sicherheit von Computersystemen und bezeichnet den verantwortungsvollen Umgang mit dem Know-how des Hackings.

Als Ethical Hacker verpflichten Sie sich, Schaden von Computersystemen abzuwenden und niemals absichtlich zu verursachen. Sie handeln nach dem Motto: »Was du nicht willst, das man dir tu', das füg' auch keinem anderen zu!«

Lernen Sie so viel über das Hacking wie möglich und seien Sie immer neugierig – doch die Freiheit des einen hört dort auf, wo die Freiheit des anderen eingeschränkt wird! Greifen Sie niemals ohne schriftliche Genehmigung und eindeutige Auftragsklärung fremde Systeme an. Das Wissen über theoretische und praktische Hacking-Technologien verpflichtet. So wie ein Kampfsportler seine Fähigkeiten nur im Ring bzw. auf der Matte und nicht auf der Straße anwenden darf, so bleibt ein Ethical Hacker immer im ethischen und rechtlichen Rahmen des Erlaubten. Gutes Stichwort, dazu gibt es noch etwas Wichtiges zu erläutern.

## Der Hacker-Paragraf

Im Jahr 2007 wurde im Rahmen der »Strafvorschriften zur Bekämpfung der Computerkriminalität« der Paragraf 202c des Strafgesetzbuches (StGB) eingeführt. Er lautet folgendermaßen:

- (1) *Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er*
  - 1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder*
  - 2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist, herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu zwei Jahren oder mit Geldstrafe bestraft.*
- (2) *§ 149 Abs. 2 und 3 gilt entsprechend.*

Das umfasst grundsätzlich auch die Hacker-Tools, deren sich nicht nur die bösen Jungs, sondern auch Administratoren und Sicherheitsbeauftragte bedienen, um die Sicherheit von Computersystemen und -netzwerken zu erhöhen. Bevor Sie jetzt jedoch aus rechtlichen Bedenken dieses Buch zuschlagen und sich dem Fernsehprogramm widmen, dürfen wir Sie beruhigen: Auch wenn der Wortlaut hier leider sehr schwammig ist und eine weitgefasste Auslegung zulassen würde, so dient der Paragraf seinem Inhalt nach nur der Vereitelung von Straftaten.

Die bisherige Rechtsprechung zeigt, dass die Verwendung dieser Tools zur Erhöhung der Sicherheit von IT-Infrastrukturen keine Strafverfolgung nach sich zieht. Dennoch bleibt eine gewisse rechtliche Unsicherheit. Der entsprechende Wikipedia-Artikel ist sehr aufschlussreich und einen Blick wert: [https://de.wikipedia.org/wiki/Vorbereiten\\_des\\_Ausspähens\\_und\\_Abfangens\\_von\\_Daten](https://de.wikipedia.org/wiki/Vorbereiten_des_Ausspähens_und_Abfangens_von_Daten). Sichern Sie sich beim Hacking bzw. Penetration Testing in fremden Umgebungen immer schriftlich und umfangreich ab, indem Sie Art und Umfang Ihrer Tätigkeit (bzw. des Penetrations-Tests) ganz genau beschreiben und anschließend auch ausführlich dokumentieren.

## 1.5 Der Certified Ethical Hacker (CEHv11)

Dieses Buch versteht sich als eine fundierte, praxisorientierte Einführung in das Thema »Ethical Hacking«. Es ist an die Inhalte der Prüfung zum *Certified Ethical Hacker* (CEHv11) angepasst und stellt somit eine wertvolle Ressource für Ihre Vorbereitung auf das Examen dar. Auch wenn der Fokus nicht primär auf der Prüfungsvorbereitung liegt, werden wir im Laufe des Buches immer wieder Hinweise zur Prüfung geben. An dieser Stelle möchten wir Ihnen einmal kurz den CEH vorstellen.

### 1.5.1 Was steckt dahinter?

Der *Certified Ethical Hacker* ist eine herstellerunabhängige Zertifizierung, die vom EC-Council ([www.eccouncil.org](http://www.eccouncil.org)) entwickelt und angeboten wird. Dahinter verbirgt sich eine Organisation, die sich auf Zertifizierungen im Hacking- und Security-Bereich spezialisiert hat.

Der CEH ist mittlerweile in der Version 11 verfügbar. Er stellt eine anspruchsvolle Basiszertifizierung für angehende Ethical Hacker und Penetrationstester dar, die durch weitergehende Zertifizierungen ergänzt wird. So steht seit dem CEHv10 optional eine ergänzende CEH-Practical-Zertifizierung zur Verfügung. Dabei handelt es sich um eine praktische Prüfung, bei der der Kandidat seine Hacking-Kenntnisse in einer praxisnahen Laborumgebung unter Beweis stellen muss. Inzwischen führen diese beiden Prüfungen zusammen zum *CEH Master*, um den Mehrwert hervorzuheben.

Wer sich darüber hinaus noch weiter in den professionellen Bereich begeben möchte, kann über den *EC-Council Certified Penetration Testing Professional* (CPENT) den nächsten Schritt gehen und auch die Expert-Level-Zertifizierung zum *Licensed Penetration Tester* (LPT) absolvieren. Mittlerweile bietet das EC-Council eine Vielzahl von Zertifizierungen und Zertifizierungspfaden an.

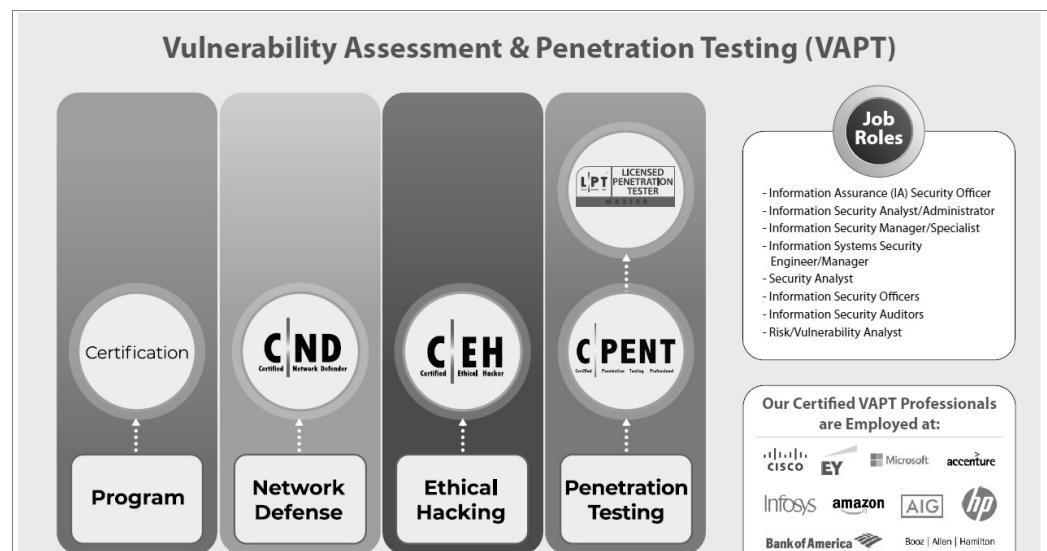


Abb. 1.1: Der Ethical-Hacking-Zertifizierungspfad beim EC-Council

Das Curriculum des CEHv11 umfasst insgesamt 20 Module, deren Inhalte in diesem Buch abgedeckt sind. Es wird ein breites Themen-Spektrum mit diversen Konzepten und unzähligen Tools abgearbeitet, wobei es hauptsächlich um Konzepte und Technologien geht und weniger darum, alle der vorgestellten Tools bis ins Detail zu beherrschen. Den Prüfling erwartet ein intensives Studium, das ein hohes Engagement und intensive Einarbeitung voraussetzt, um alle behandelten Themen in ausreichender Tiefe zu beherrschen.

### 1.5.2 Die CEHv11-Prüfung im Detail

Zur CEHv11-Prüfung werden Sie unter einer der folgenden Bedingungen zugelassen:

1. Sie absolvieren einen der offiziellen (und nicht gerade günstigen!) CEH-Kurse. Damit sind Sie automatisch qualifiziert für die Prüfung.
2. Sie reichen ein »Eligibility Form« (ein Formular für die Zulassung zur Prüfung) ein und weisen nach, dass Sie mindestens zwei Jahre Erfahrung auf dem Gebiet der IT-Sicherheit haben. Diese Zulassungsprüfung kostet Sie derzeit 100 Dollar – unabhängig vom Ausgang der Prüfung.

Im Gegensatz zum Themenspektrum und dem Inhalt des CEH-Curriculums ist die Prüfung derzeit eher geradlinig gehalten:

- Anzahl der Fragen: 125
- Maximale Testdauer: vier Stunden
- Test-Format: Multiple Choice mit nur einer richtigen Antwort
- Test wird angeboten über: VUE-Testcenter oder ECC-Online-Examen
- Test-Nummer: 312-50

Es gibt eine Aufschlüsselung in Themenkomplexe und deren Schwerpunkte, aber diese wird in regelmäßigen Abständen geändert. Die Prüfung wirkte in der Vergangenheit mitunter unausgeglichen. Ein bisher überdimensionierter Schwerpunkt lag auf Nmap-Befehlen und auf kryptografischen Konzepten. Dies ist jedoch keine Garantie für Ihren Prüfungszeitpunkt. Von daher empfehlen wir Ihnen, sich im Internet in einschlägigen Foren Informationen zur Prüfung einzuholen, wenn Ihr Prüfungszeitpunkt konkret wird.

Unter dem Strich ist die Zertifizierung zum CEH eine gute Ergänzung zur Schärfung Ihres Profils und kann Ihre Karrierechancen deutlich verbessern. Sie ist allerdings mit derzeit 950 bzw. 1200 Dollar sehr teuer. Der Preis ist abhängig davon, ob Sie die Prüfung im ECC Exam Center oder in einem VUE-Prüfungscenter absolvieren möchten.

Sie sollten insbesondere in folgenden Szenarien über eine CEH-Zertifizierung nachdenken:

- Sie möchten zukünftig als Penetrationstester arbeiten und benötigen einen Nachweis Ihrer Qualifikation.
- Ihre Tätigkeit liegt im IT-Security-Bereich und Sie möchten Ihr Einsatzgebiet erweitern.
- Sie arbeiten als Security Analyst und möchten Ihr Wissen zertifizieren.

Wir halten die Zertifizierung für ein sehr gutes Fundament für den Einstieg in eine Karriere als Ethical Hacker und Penetrationstester. Um aus diesem Buch das Maximum herauszuholen, ist jedoch die Prüfung zum CEH keine Voraussetzung. Trotzdem werden wir immer wieder auf die CEH-Prüfung zurückkommen und Tipps und Prüfungshinweise geben.

## 1.6 Die Schutzziele: Was wird angegriffen?

Distanzieren wir uns für einen Moment von unserer Hacker-Rolle und setzen die Brille derjenigen auf, die Computersysteme und deren Daten schützen müssen. Denn Hacking und Penetration Testing dient aus Sicht der Offensive Security zur Absicherung der Systeme. Betrachten wir also den Blickwinkel des Security-Verantwortlichen einer Organisation.

Die IT-Sicherheit definiert drei grundlegende Schutzziele, die durch Angriffe auf IT-Systeme bedroht werden. Sie werden mit **C I A** abgekürzt. Dies steht in diesem Fall nicht für Central Intelligence Agency, sondern ist eine Abkürzung für:

- **Confidentiality** = Vertraulichkeit
- **Integrity** = Integrität
- **Availability** = Verfügbarkeit

Manchmal wird ein vierter Schutzziel, die **Authenticity** (= Authentizität) definiert. Diese dient auch der **Non-Repudiation**, was etwas hölzern als *Nicht-Abstreitbarkeit* übersetzt wird. Dieses Thema wird aber oft im Schutzziel **Integrität** enthalten gesehen.

### Tipp: Kompromittierte Systeme sind per se nicht mehr sicher

Unter dem Strich möchten die Sicherheitsverantwortlichen hauptsächlich sicherstellen, dass die Daten und Systeme nicht *kompromittiert* werden. Bei einem kompromittierten System kann der Eigentümer sich nicht mehr sicher sein, dass die darauf enthaltenen Daten unverändert bzw. nach wie vor vertraulich sind und die korrekte Funktion der Dienste noch gegeben ist. Ein kompromittiertes System sollte meistens von Grund auf neu aufgesetzt werden.

Umgekehrt ist es also das Ziel von Hackern, Computersysteme zu kompromittieren und damit ganz oder teilweise unter ihre Kontrolle zu bringen. Eine Ausnahme stellen die destruktiven *Denial-of-Service-Angriffe* dar, bei denen es nur darum geht, dass das gesamte System oder Teile des Systems nicht mehr funktionieren.

Kaum zu glauben, dass sich der Schutzbedarf von Computersystemen auf die oben genannten drei bzw. vier Schutzziele herunterbrechen lässt. Sehen wir uns daher die einzelnen Schutzziele aus Sicht der IT-Sicherheit einmal im Detail an:

### 1.6.1 Vertraulichkeit

Es gibt Daten, bei denen ist es dem Eigentümer egal, ob sie öffentlich zugänglich sind oder nicht. Oftmals ist es aus Sicht des Eigentümers sogar wünschenswert, wenn diese Daten Beachtung finden. Hierzu zählen zum Beispiel:

- **Unternehmensadresse(n)**: Zumindest die meisten Unternehmen leben davon, gefunden zu werden.
- **Marketing-Materialien**: Stellen Sie sich vor, ein Unternehmen erstellt Werbespots, veröffentlicht diese aber nicht ... das ginge dann ziemlich am Sinn vorbei.
- **Produkt-Beschreibungen**: Soll das Produkt verkauft werden, müssen potenzielle Käufer einen Einblick in die Eigenschaften des Produkts erhalten können, z.B. in Form eines Downloads von PDF-Dateien von der Website.

- **White-Paper:** Diese Übersichtsdokumente enthalten Erläuterungen zu Technologien, Fallstudien und Ansätze für Problemlösungen. Sie dienen der Öffentlichkeitsarbeit.
- **Give-Aways:** Kleine Geschenke erhalten die Freundschaft. Kostenlose Downloads oder klassische Geschenke, wie Kugelschreiber oder Tassen, erhöhen die Kundenbindung.

Die obige Aufzählung ist nur exemplarisch. Es gibt noch jede Menge weiterer Informationen, die öffentlich zugänglich sind und es aus der Sicht des Eigentümers auch sein sollen.

Andererseits sind die meisten Daten und Informationen von Personen, Unternehmen und Organisationen schützenswert und sollten oder dürfen der Öffentlichkeit nicht zugänglich gemacht werden. Eine Veröffentlichung bedeutet im besten Falle Image-Schaden und im schlimmsten Fall den Untergang des Unternehmens.

Stellen Sie sich vor, ein Unternehmen entwickelt ein neues, hoch-innovatives Produkt, mit dem es eine Alleinstellung auf dem Markt anstrebt. Alle finanziellen Ressourcen werden in diese Entwicklung gesteckt. Leider gelingt es einem Hacker, die Pläne und alle Detailinformationen des Produkts zu stehlen und einem anderen Unternehmen zukommen zu lassen, das das Produkt schneller fertigstellt und auf den Markt bringen kann. Da kann unser Unternehmen dann vermutlich dicht machen. Übrigens fällt dieser Vorfall unter die Rubrik *Wirtschaftsspionage* und ist eine der am weitesten verbreiteten und lukrativsten Tätigkeiten von Black Hats und staatlich unterstützten Hackern.

Die Vertraulichkeit von Daten kann auch aus Datenschutzgründen notwendig sein. So müssen personenbezogene Daten von Kunden eines Unternehmens unbedingt vor unbefugtem Zugriff geschützt werden. Eine Veröffentlichung von Kundendaten geht in der Regel mit einem enormen Image-Schaden einher und kann auch für jeden einzelnen Kunden sehr teuer werden, wenn diese Daten dazu geeignet sind, der jeweiligen Person oder Organisation zu schaden. Dies ist z.B. bei Kreditkartendaten der Fall. (So geschehen 2011 bei Sonys Playstation Network.) Auch die Veröffentlichung von Patientendaten ist hochkritisch.

Die Vertraulichkeit ist also für viele Daten essenziell. Da nicht alle Daten den gleichen Schutzbedarf haben, werden oftmals Schutzklassen bzw. Sicherheitsstufen (z.B. *öffentlich*, *sensibel*, *geheim*, *Top Secret*) definiert, denen die jeweiligen Daten zugeordnet werden. In Deutschland existiert hierzu mit DIN 66399 sogar eine Norm.

Je nach Schutzklasse und Sicherheitsstufe wird in diesem Zusammenhang der jeweilige Sicherheitsbedarf festgelegt. Je höher, desto mehr und umfangreichere Sicherheitsmechanismen werden zum Schutz der Daten bereitgestellt und desto strenger sind die Kontrollen. Dies erklärt andererseits auch, warum (bösertige) Hacker insbesondere von den besonders geschützten Daten angezogen werden wie die Motten vom Licht.

Auf der anderen Seite gibt es für alle relevanten Daten immer auch Personen, die auf die jeweiligen Daten zugreifen müssen. Es ist also zum einen notwendig, die autorisierten Zugriffe festzulegen, und zum anderen, dafür zu sorgen, dass nicht-autorisierte Zugriffe unterbunden werden. Dabei erhält ein Benutzer oder eine Benutzergruppe in der Regel eine eindeutige Kennung (ID) und eine Möglichkeit, sich zu authentisieren. Ist seine *Authentizität* festgestellt, erhält er Zugriff auf diejenigen Daten, für die er *autorisiert* ist. In Abschnitt 1.6.4 gehen wir weiter in die Details der Authentisierung.

## Schutzmaßnahmen

Die Maßnahmen zur Sicherstellung der Vertraulichkeit können ganz unterschiedlich aussehen und auf unterschiedlichen Ebenen ansetzen. Typische Sicherheitssysteme in Computernetzwerken sind:

- **Firewalls:** Klassisches Instrument zur Steuerung von Netzwerk-Traffic und Verhinderung von unerwünschter Kommunikation.
- **Virenschutzsysteme:** Auch Antivirus-Systeme (kurz: AV) genannt. Dienen zum Verhindern von *Malware* (bösertiger Software).
- **Intrusion-Detection/Prevention-Systeme:** Kurz: IDS/IPS, dienen der Erkennung von Angriffsmustern und – im Falle von IPS – der automatischen Abwehr des Angriffs.
- **Application Gateways:** Analysieren die Kommunikation auf Protokollbene bis in die Details und können fehlerhafte und unerwünschte Kommunikation erkennen und blockieren.
- **Zugangskontrollsysteme:** Sowohl physische als auch logische Systeme dienen dazu, den Zugriff auf zu schützende Daten auf die autorisierten Personen zu beschränken.

Die wohl wichtigste Maßnahme zur Sicherstellung der Vertraulichkeit im Rahmen der Netzwerk-Kommunikation ist die *Verschlüsselung*. Sie stellt sicher, dass ein Angreifer den Inhalt einer Kommunikation nicht erkennen kann.

### **Vorsicht: Verschlüsselung verhindert nicht Veränderung**

Bei einem *Man-in-the-Middle-Angriff* positioniert sich der Angreifer zwischen den Kommunikationspartnern und übernimmt unbemerkt jeweils stellvertretend für den anderen die Kommunikation. Beide Kommunikationspartner glauben, dass sie mit dem jeweils anderen kommunizieren, während der Angreifer jedes Datenpaket abfangen, analysieren, ggf. verändern und dann an den echten Empfänger weiterleiten kann. Die Verschlüsselung verhindert, dass der Angreifer die Daten entziffern kann, jedoch nicht, dass sie verändert weitergeleitet werden.

Um sicherzustellen, dass die gesendeten Daten unverändert beim Empfänger ankommen oder auf einem Datenträger abgelegte Daten zwischenzeitlich nicht verändert wurden, müssen wir die *Integrität* der Daten wahren.

## **1.6.2 Integrität**

Es war einmal ein Mitarbeiter, dem von seinem Unternehmen gekündigt wurde. Dieser war ob der Kündigung erzürnt und wollte sich an seinem Unternehmen rächen. Zu diesem Zwecke erlernte er das Hacking und führte eine *Man-in-the-Middle-Attacke* aus, indem er ausgehende Angebots-mails des Unternehmens abfing und verändert an den Adressaten weiterleitete. Immer, wenn das Unternehmen ein Dienstleistungsangebot mit einem guten Preis an einen Interessenten aussendete, veränderte er den Preis derart, dass die Dienstleistung viel zu teuer wäre – statt 1500 Euro las der Interessent nun 15.000 Euro als Gesamtpreis, lachte kurz und wandte sich von diesem Unternehmen ab, um die Dienstleistung bei einem anderen Unternehmen einzukaufen ...

Dem Unternehmen ging viel Geld dadurch verloren und der ehemalige Mitarbeiter erhielt seine Rache. Ende der Geschichte.

Tatsächlich ist die Frage, ob gesendete Daten beim Empfänger unverändert ankommen, oftmals essenziell – dabei geht es nicht immer um Geld. Es gibt populäre Fälle, in denen eine renommierte Software auf dem Server so manipuliert wurde, dass sie auf dem Opfer-System eine sogenannte »Backdoor« installierte, um Angreifern einen unbemerkten Remote-Zugang zum System zu ermöglichen.

Angriffe der oben beschriebenen Art können verhindert werden, wenn es gelingt, die Integrität der Daten sicherzustellen. Wir betrachten also die »Echtheit« der Daten. Das Ziel ist es, Daten vor Manipulationen zu schützen.

Wie bereits dargelegt, können das Dateien sein, die auf einem Server liegen und unbemerkt gegen eine manipulierte Version ausgetauscht, oder Informationen, die bei der Übermittlung manipuliert werden, wie in unserem Eingangsbeispiel.

Es muss sichergestellt werden, dass die Daten, die den Sender verlassen, auch genauso beim Empfänger ankommen und unterwegs nicht verändert oder ausgetauscht werden. Neben veränderten Inhalten kann aber auch der Absender eines Datenpakets manipuliert werden. Hierbei geht es dann um Authentizität, die ebenfalls mit Mitteln der Integrität sichergestellt werden kann.

## Schutzmaßnahmen

Um die Integrität von Daten zu gewährleisten, kommt oft ein sogenannter *Hashwert* zum Einsatz. Das ist eine mathematische Funktion, die auf eine Nachricht oder eine Datei angewendet werden kann. Dabei wird die Original-Nachricht als Eingangswert von der Hash-Funktion verarbeitet. Daraus entsteht eine immer gleich lange Kombination aus Zeichen, das ist der Hashwert. Von diesem lässt sich nicht auf den Inhalt der Nachricht zurücksließen, aber er identifiziert diese ganz genau.

Wie der Fingerabdruck eines Menschen eine Person identifiziert, aber keinerlei Informationen zu Größe, Gewicht oder Haarfarbe preisgibt, so verschickt der Sender seine Nachricht inklusive Hashwert an den Empfänger. Dabei muss er den Hashwert so schützen, dass der Angreifer diesen nicht unerkannt ändern kann. Dies geschieht z.B. mittels digitaler Signatur.

Der Empfänger wendet dieselbe Hash-Funktion auf die Nachricht an und vergleicht den ermittelten Hashwert mit dem des Senders. Wurde an der Nachricht nur ein einziges Zeichen verändert, stimmt der Hashwert nicht überein. Damit kann der Empfänger die Echtheit der empfangenen Daten überprüfen.

### Vorsicht: Die Integritätsprüfung verhindert nicht die Manipulation der Daten!

»Moment mal!«, werden Sie vielleicht sagen: »Mit der Integritätsprüfung will ich doch die Echtheit der Daten sicherstellen?« Jupp! Das können Sie auch – was Sie aber *nicht* können, ist, zu *verhindern*, dass die Daten manipuliert werden. Sie können es lediglich erkennen und entsprechend reagieren. Mehr kann die Integritätsprüfung nicht leisten. Ein kleiner, aber feiner und wichtiger Unterschied.

Was also tun, wenn wir bemerken, dass die Integrität von Daten nicht gewahrt werden konnte? In diesem Fall muss die Nachricht oder Datei verworfen werden, sie ist nicht mehr vertrauenswürdig. Im Fall einer Netzwerk-Kommunikation muss der Absender seine Informationen erneut senden. Dumm nur, wenn die dazu notwendigen Systeme aufgrund eines Angriffs den Dienst versagen. Dieser Punkt betrifft das dritte Sicherheitsziel, die Verfügbarkeit von Daten in der gewünschten Art und zum gewünschten Zeitpunkt.

Auf das Thema Kryptografie gehen wir aufgrund seiner Bedeutung noch einmal gesondert ein. In Kapitel 5 erfahren Sie viele Details über Verschlüsselungsvarianten, -algorithmen und -verfahren.

### 1.6.3 Verfügbarkeit

Vielleicht erinnern Sie sich noch an Weihnachten 2014, als die Netzwerke der Spielekonsolen von Sony und Microsoft lahmgelegt wurden? Die neuen Spiele, die zum Fest verschenkt wurden, konnten erst einmal nur begrenzt zum Einsatz kommen, was den Herstellern viel Ärger einbrachte.

Ursache dafür war ein sogenannter *DoS-Angriff* (Denial-of-Service). Dabei versuchen Angreifer, ein System in die Knie zu zwingen, bis es seinen Dienst quittiert. Dies geschieht zum Beispiel durch eine Flut von Anfragen an das Zielsystem oder durch Ausnutzen einer bekannten Schwachstelle, die das System zum Absturz bringt. In diesem Fall reicht manchmal schon ein einziges, entsprechend manipuliertes Datenpaket.

Angreifer versuchen mittels der oben beschriebenen Denial-of-Service-Angriffe (DoS), die Verfügbarkeit von Systemen im Netzwerk und im Internet zu untergraben. Oftmals geschieht dies mit der Brechstange, indem die Opfer-Systeme mit so vielen Anfragen überhäuft werden, dass sie diese nicht mehr verarbeiten können.

Um die Wirksamkeit dieser Angriffe zu erhöhen, werden *Distributed-Denial-of-Service-Angriffe* (DDoS, sprich: Di-Dos) gefahren, bei denen der Angriff von Hunderten oder Tausenden Systemen aus dem Internet stattfindet. Hierzu dienen sogenannte »Botnetze«, bei denen eigentlich harmlose Computer zu einem früheren Zeitpunkt mit einer Software infiziert wurden, die ferngesteuert einen Angriff zu einem gewünschten Zeitpunkt initiiert.

### Schutzmaßnahmen

Sich gegen einen DoS- oder DDoS-Angriff zu schützen, ist eine der schwierigsten Angelegenheiten der IT-Sicherheit. Im März 2013 fand aus Rache am Blacklist-Anbieter *Spamhaus* ein DDoS-Angriff statt, der eine Woche dauerte. Initiiert wurde er vom niederländischen Provider Cyberbunker, der sich dagegen wehren wollte, dass Spamhaus diverse seiner Kunden auf die schwarze Liste (Blacklist) gesetzt hatte, weil diese Spam und anderen unerwünschten Traffic erzeugt hatten. Der DDoS-Angriff war derart heftig, dass ein nicht unerheblicher Teil des Internets davon betroffen war und es auch andernorts zu Leistungseinbußen kam.

Für viele Unternehmen und Organisationen ist die Verfügbarkeit des Computernetzwerks und seiner Systeme essenziell. Daher werden diverse Maßnahmen ergriffen, um dies sicherzustellen. Hierbei können verschiedene Technologien zum Einsatz kommen, zum Beispiel:

- **High Availability (HA):** Auch hierbei werden redundante Systeme bereitgestellt, die entweder parallel aktiv oder im Aktiv/Passiv-Modus arbeiten, also die Funktion sofort übernehmen können, wenn das Hauptsystem ausfällt. Bei HA ist es nicht unbedingt erforderlich, dass die Systeme als Cluster arbeiten.
- **Clustering:** Dabei werden mehrere gleichartige Systeme zu einem Verbund zusammengeschlossen. Fällt eines oder sogar mehrere dieser Verbundsysteme aus, können die anderen die Funktion trotzdem aufrechterhalten. Clustering unterscheidet sich von High Availability insofern, als es die Bereitstellung eines gemeinsamen Speichers erfordert, *Quorum* genannt.
- **Loadbalancing:** Dahinter versteckt sich das Konzept, die Anfragen von Client-Systemen automatisch nach bestimmten Kriterien auf verschiedene, gleichartige Systeme zu verteilen, um die Last aufzuteilen.

Es existieren diverse weitere Technologien speziell zur Vermeidung von DDoS-Angriffen, wie z.B. Scrubbing-Center und Content-Delivery-Netzwerke. Im Internet existieren Dienstanbieter, die sich

auf die Erhaltung der Verfügbarkeit der Systeme spezialisiert haben. Wir kommen in Kapitel 22 *DoS- und DDoS-Angriffe* darauf zurück.

### 1.6.4 Authentizität und Nicht-Abstreitbarkeit

Was passiert hinter den Kulissen, wenn Sie sich an einem Computer anmelden? Sie geben Ihren Benutzernamen an, tippen Ihr Kennwort ein und bestätigen diese Eingabe. Im Hintergrund prüft der Computer nun, ob er Sie kennt. Das ermittelt er anhand der Benutzer-ID, in diesem Fall Ihrem Benutzernamen. Dazu existiert in Windows-Systemen ein sogenanntes Benutzerkonto. Anschließend vergleicht er das für Ihr Benutzerkonto hinterlegte Passwort mit dem eingegebenen (in der Regel vergleicht er die Hashwerte, da das Passwort aus Sicherheitsgründen nicht direkt hinterlegt ist).

Passt alles zusammen, sind Sie *authentifiziert*. Das bedeutet nichts anderes, als dass der Computer Ihnen Ihre Identität glaubt und Sie für diejenige Person hält, für die Sie sich ausgeben. An dieser Stelle kommt immer auch die *Autorisierung* ins Spiel: Durch die Vergabe von Zugriffs- und Systemrechten erhalten Sie nun die Möglichkeit, in einer festgelegten Art auf bestimmte Daten zuzugreifen, z.B. nur lesend (*read-only*) oder lesend oder schreibend. Auch die Verwendung von Programmen und der Zugriff auf die Systemkonfiguration sind von Ihren Rechten abhängig. Ein Administrator darf hier deutlich mehr (im Zweifel alles) als ein nicht-privilegierter Benutzer.

Neben der Autorisierung dient die Authentizität bzw. Authentisierung in bestimmten Situationen auch der *Nicht-Abstreitbarkeit* (engl. *Non-Repudiation*). Geben Sie z.B. über das Internet eine Bestellung auf und behaupten später, dass Sie das gar nicht getan hätten, so streiten Sie die Bestellung ab und der Auftragnehmer hat das Beweisproblem. Gerade bei Geschäftsbeziehungen, die über das Internet laufen, spielt dies eine große Rolle.

Ziel der Nicht-Abstreitbarkeit ist der Nachweis, dass eine Nachricht mit einem bestimmten Inhalt tatsächlich von der Person gekommen ist, die als Absender angegeben ist. Dies wird durch ähnliche Methoden erreicht, wie sie bei der Sicherstellung der Integrität eingesetzt werden.

## Schutzmaßnahmen

Eine große Rolle spielen hier Hashwerte als Prüfsummen und ein Konzept namens *digitale Signatur* oder *elektronische Unterschrift*. Durch die digitale Signatur kann eindeutig nachgewiesen werden, dass eine Nachricht von einem bestimmten Absender stammt. Im Zusammenspiel mit der Integritätsprüfung kann auch der Inhalt verifiziert werden, sodass eine Nicht-Abstreitbarkeit erreicht wird. Dadurch werden Geschäftsbeziehungen im Internet glaubwürdig. Gelingt es einem Angreifer, diese digitale Signatur oder die Hashwerte zur Integritätsprüfung zu falschen, wiegt sich der Empfänger einer Nachricht in falscher Sicherheit. Im Rahmen von Kapitel 5 *Kryptografie und ihre Schwachstellen* nennen wir Ihnen effektive Methoden, Ihre Integrität und Authentizität zu schützen.

### 1.6.5 Die Quadratur des Kreises

Sind Sie verantwortlich für die IT-Sicherheit, sollten Sie immer die oben genannten Schutzziele im Auge behalten und sich entsprechend schützen.

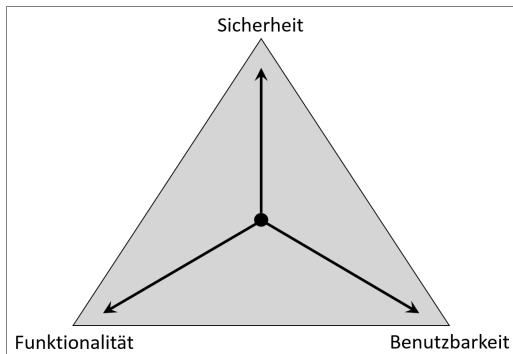
Bei allem Sicherheitsbewusstsein, das wir bei Ihnen im Laufe dieses Buches verstärken möchten, dürfen Sie allerdings nie das Verhältnis zwischen Sicherheit, Funktionalität und Bedienbarkeit außer Acht lassen.

Je nachdem, wo Sie Schwerpunkte setzen, verlagert sich die Balance Ihrer Computersysteme. Natürlich können Sie die Sicherheit zu 100 % sicherstellen – indem Sie die Systeme abschalten und nie-

mandem zugänglich machen. In diesem Fall würden Funktionalität und Benutzbarkeit auf 0 % reduziert. Und dies ist sicherlich nicht zielführend.

Die anderen Extreme bringen jedoch auch Probleme mit sich: Die Benutzbarkeit zu maximieren, führt in jedem Fall zu vermehrten Sicherheitslücken. So könnten Sie z.B. auf Zugangskontrolle verzichten und jedem Vollzugriff auf alle Systeme und Daten geben. Dass das ebenfalls nicht zum gewünschten Gesamtergebnis führt, müssen wir nicht weiter ausführen.

Das bedeutet letztlich, dass Sie als Sicherheitsbeauftragte(r) manchmal Kompromisse eingehen müssen, die gegen das Sicherheitsziel sprechen. Wenn die Funktionen zu sehr eingeschränkt sind oder sich Ihr System nicht mehr effizient bedienen lässt, haben Sie auch nichts gewonnen. Versuchen Sie, einen gesunden Mittelpunkt im Inneren des Dreiecks zu finden.



**Abb. 1.2:** Immer auf das Verhältnis achten

Welche Balance das Optimum in der jeweiligen Umgebung darstellt, lässt sich pauschal nicht beantworten. So wird eine Bank z.B. naturgemäß sehr viel mehr Wert auf Sicherheit legen – zur Not eben auch auf Kosten der Benutzbarkeit (Usability) und Funktionalität. Mittlerweile ist ja das Einloggen in den Online-Bankaccount oft schon ein dreistufiger Authentifizierungsprozess und teilweise recht nervig für den Kunden.

Auf der anderen Seite gibt es Unternehmen, die von der Kreativität und Individualität ihrer Mitarbeiter leben. Hier könnte es notwendig sein, vielen Mitarbeitern weitgehende Rechte bis hin zu Administratorprivilegien einzuräumen, damit diese ihre Jobs optimal ausfüllen können. Dies ist zwar ein Horrorszenario für jeden Security-Beauftragten, aber wenn die Alternative lautet, dass das Unternehmen pleitegeht, weil die Mitarbeiter nicht vernünftig arbeiten können, müssen entsprechende, aus Security-Sicht manchmal schmerzhafte, Kompromisse gefunden werden.

### Tipp: Das Prinzip der Least Privileges und das Vier-Augen-Prinzip

Grundsätzlich gilt: Jeder Benutzer erhält so viel Rechte wie nötig und so wenig wie möglich, um seine Tätigkeit ausüben zu können! Führt ein Recht zu einem Sicherheitsproblem, suchen Sie nach Alternativen: Ist es z.B. möglich, bestimmte, sicherheitskritische Prozesse durch nur einen oder wenige Mitarbeiter ausführen zu lassen, anstatt durch jeden einzelnen Benutzer? Sorgen Sie im Zweifel auch immer für ein Vier-Augen-Prinzip: Ein Mitarbeiter beantragt einen Prozess, ein zweiter genehmigt diesen und der dritte führt ihn schließlich aus. Das reduziert den Missbrauch von privilegierten Funktionen, wie z.B. das Ändern von Firewall-Regeln.

## 1.7 Systematischer Ablauf eines Hacking-Angriffs

Einer der Haupt-Unterschiede zwischen Scriptkiddies und echten Hackern oder auch Pentestern ist das systematische Vorgehen, das bei den Scriptkiddies fehlt. Ein professioneller Hacking-Angriff umfasst eine Reihe von Phasen, die aufeinander aufbauen. Es gibt verschiedene Ansätze, die leicht voneinander abweichen, aber inhaltlich weitgehend denselben Weg verfolgen. Abbildung 1.3 zeigt eine Übersicht über die einzelnen Etappen, wie sie vom CEH-Curriculum unterschieden werden.

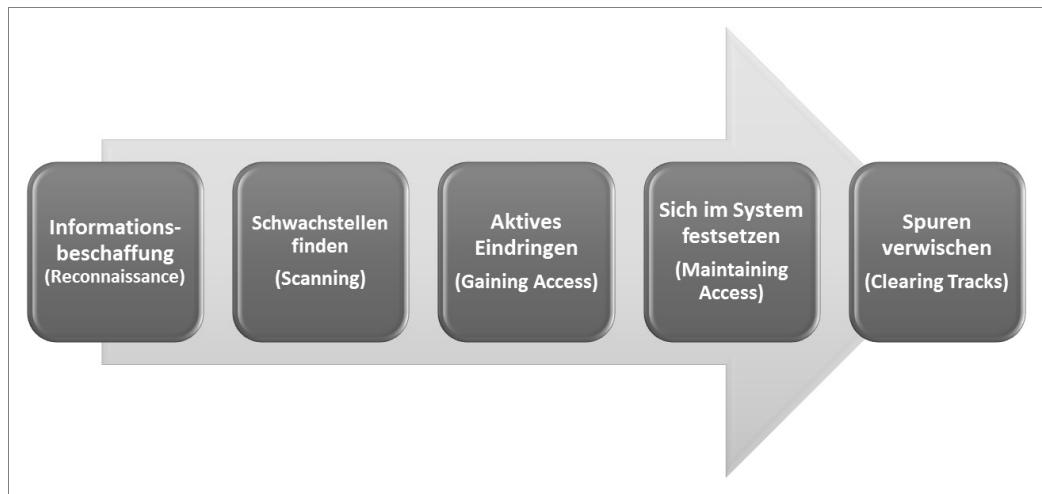


Abb. 1.3: Prozess-Schritte eines Hacking-Angriffs

Hierbei ergibt sich jedoch eine Begriffsüberschneidung, da die zweite Phase, das *Scanning*, in den meisten Quellen zur aktiven *Reconnaissance-Phase* hinzugerechnet wird. An dieser Stelle gibt es diverse Begrifflichkeiten zu unterscheiden. Wir werden das gleich noch etwas genauer erläutern.

Auch wenn die Vorgehensweise von Black Hat Hackern und White Hat Hackern grundsätzlich gleich ist, so sind die Phasen bei einem realen Angriff noch etwas umfangreicher und aggressiver. Schauen wir uns das einmal an.

### 1.7.1 Phasen eines echten Angriffs

Im Rahmen eines professionellen Hacking-Angriffs versucht der Angreifer, sein Ziel systematisch und nachhaltig zu erreichen. So hat er z.B. nichts gewonnen, wenn er zwar die gesuchten Daten findet und stehlen kann, dabei aber erwischt wird. Daher ist es notwendig, mit Bedacht vorzugehen und möglichst wenig Spuren zu hinterlassen. Zudem kann der Angreifer die Chance nutzen, im Rahmen eines erfolgreichen Angriffs eine Hintertür einzubauen, die ihm auch zukünftig Zugang zu dem betreffenden System sichert.

Für einen erfolgreichen Angriff wird der Hacker in der Regel eine bestimmte Reihenfolge seiner Handlungen verfolgen, um sich seinem Ziel schrittweise zu nähern und nach erfolgreichem Angriff auch wieder unbemerkt abtauchen zu können. Betrachten wir die einzelnen Schritte einmal genauer:

## Informationsbeschaffung (Reconnaissance)

Dies ist der erste Schritt für die Vorbereitung auf einen Angriff. Sammeln Sie möglichst viele Informationen über Ihr Ziel. Je mehr Informationen Sie haben, umso gezielter können die nächsten Schritte gewählt werden. Das spart nicht nur Zeit, sondern erhöht auch die Chance, Schwachstellen zu finden. Wir unterscheiden zwischen zwei Phasen:

- **Passive Discovery:** In dieser Phase versuchen Sie, Informationen über Ihr Ziel (also die Person oder das Unternehmen) zu erlangen, ohne direkt mit ihm in Kontakt zu treten. Dies umfasst z.B. Google-Suchen, Social-Media-Analysen und andere Recherchen über das Ziel, kann aber auch bedeuten, dass Sie das Gebäude des betreffenden Unternehmens beobachten, um die Verhaltensweisen und Gewohnheiten der Mitarbeiter und des Wachpersonals zu erkunden. Passive Discovery umfasst damit auch einen Teil des *Social Engineerings* (grob ausgedrückt ist das alles, was primär mit Menschen statt Computern zu tun hat, genauer wird dieses Thema in Kapitel 20 *Social Engineering* behandelt) sowie das sogenannte *Dumpster Diving*, bei dem der Angreifer versucht, aus dem Müll des Opfers relevante Informationen zu erlangen. Dies kann z.B. erfolgreich sein, wenn wichtige Dokumente nicht sachgerecht entsorgt werden.
- **Active Discovery:** Jetzt werden Sie als Angreifer konkreter und prüfen die Systeme durch aktives »Anklopfen«. Das heißt, Sie treten bereits mit den Systemen des Opfers in Kontakt. In dieser Phase setzen Sie sich erstmalig der Gefahr aus, entdeckt zu werden. Andererseits können Sie aber auch wichtige Informationen zu den Zielsystemen erlangen, die weitere Angriffsvorbereitungen ermöglichen.

### Wichtig: Verschiedene Perspektiven unterscheiden!

Der CEH sieht in der Active-Discovery-Phase noch keine Scanning-Aktivitäten, sondern die Verbindungsaufnahme mit dem Ziel auf anderen Ebenen, z.B. einem Telefonanruf beim Help Desk oder in der IT-Abteilung. Wir betrachten daher die Scanning-Phase formal auch von der Reconnaissance-Phase getrennt, sehen aber inhaltlich das Scanning als Bestandteil der Active-Discovery-Phase.

## Schwachstellen finden (Scanning)

Somit geht die Active-Discovery-Phase sozusagen fließend in die Scanning-Phase über. In dieser Phase werden die Zielsysteme genau unter die Lupe genommen. Dabei nutzen Sie als Angreifer die Informationen, die Sie im Rahmen des ersten Schrittes der (passiven) Informationsbeschaffung (Reconnaissance) erlangt haben. Hier kommen Netzwerk-Scanner und -Mapper sowie Vulnerability-Scanner zum Einsatz. Tatsächlich erhöht sich der Grad der Aggressivität des Scans gegenüber dem Active Discovery.

In dieser Phase ermittelt der Angreifer die Architektur des Netzwerks, offene Ports und Dienste, die Art der Dienste, Betriebssysteme, Patchstände, scannt auf bekannte Schwachstellen und Sicherheitslücken etc. In dieser Phase steigt die Entdeckungsgefahr weiter, da der Angreifer sehr aktiv und teilweise aggressiv mit den Zielsystemen kommuniziert.

## Aktives Eindringen (Gaining Access)

Hier geht es richtig los, denn jetzt versuchen Sie, die gefundenen Lücken auszunutzen und sich mittels entsprechender Exploits unerlaubten Zugriff zu verschaffen. Angriffe gibt es in allen mögli-

chen Varianten, wie Webserver-Attacken, SQL-Injection, Session Hijacking, Buffer Overflow etc. Diese werden wir ausführlich vorstellen und natürlich auch praktisch demonstrieren.

## Sich im System festsetzen (Maintaining Access)

Hat der Angreifer sich erst einmal Zugang verschafft, versucht er, den Zugriff auszubauen. Er bemüht sich mittels *Privilege Escalation* um noch mehr Rechte und versucht, das System weitestgehend einzunehmen. Mittlerweile hat er nicht nur Zugang zum System, sondern bestenfalls sogar Administrator-Privilegien. Damit gibt sich ein professioneller Angreifer jedoch nicht zufrieden. Denn an dieser Stelle nutzen Black Hats die Gunst der Stunde, weitere Sicherheitslücken zu schaffen und über entsprechende »Backdoors« dafür zu sorgen, dass sie das Opfer-System jederzeit wieder »besuchen« können.

Das kann auch hilfreich sein, sollte die Lücke, durch die der Angreifer hineingekommen ist, geschlossen werden. Jetzt wird Ihnen vermutlich auch klar, warum Sie einem einmal kompromittierten System nicht mehr trauen können: Als Administrator eines einmal kompromittierten Systems werden Sie keine ruhige Nacht mehr haben, mit dem Hintergedanken, dass der Angreifer evtl. weitere Einfallstore und Zugänge installiert hat.

## Spuren verwischen (Clearing Tracks)

In den meisten Fällen entstehen bei einem Hacking-Angriff Spuren, die durch Methoden der Computer-Forensik ausgewertet werden können. Ist der Angriff auf den Hacker zurückzuführen, so ist dessen Karriere schnell vorbei.

In dieser Phase geht es also darum, die Spuren seines (unerlaubten) Tuns möglichst nachhaltig und umfangreich zu verwischen. Hierzu werden Logging-Einträge manipuliert oder gelöscht, Rootkits installiert, die sehr tief im Kernel operieren und das System und dessen Wahrnehmung der Ereignisse manipulieren können, sowie Kommunikationsprotokolle und -wege eingesetzt, die eine Nachverfolgung erschweren.

Nicht immer müssen die Angriffe strikt in dieser Reihenfolge ablaufen. So kann es durchaus sein, dass Sie einen Scan auf ein System laufen lassen, während Sie in der Zwischenzeit in ein anderes einbrechen. Auch macht es Sinn, zwischen den einzelnen Schritten seine Spuren immer wieder zu verwischen, obwohl diese Phase generell erst am Ende der Kette steht. Um allerdings den grundlegenden Ablauf zu verstehen und zu verinnerlichen, ist es wichtig, die Phasen und ihre Reihenfolge zu kennen und ständig im Blick zu haben.

### 1.7.2 Unterschied zum Penetration Testing

Sie haben vielleicht bemerkt, dass die im vorigen Abschnitt vorgestellten Phasen – gerade die letzten beiden – doch recht »dunkel« anmuten. Und auch wenn das beschriebene Vorgehen weitgehend sowohl für White Hats als auch für Black Hats gilt, so ist der Vorgang beim Penetration Testing im Allgemeinen doch noch ein wenig modifiziert. Dies betrifft insbesondere folgende Punkte:

## Vorbereitung

Vor einem Penetrationstest wird sehr genau festgelegt, was die Ziele des Audits sind und in welchem Rahmen der Pentester sich bewegt. Es wird die Aggressivität des Tests festgelegt und die Kommunikation zwischen dem Pentester und dem Auftraggeber geklärt.

Der Auftraggeber wird während des Tests in der Regel in Intervallen über den aktuellen Stand aufgeklärt und über einzelne, geplante Schritte hinsichtlich Zeitraum und Umfang informiert. Dies wird ebenfalls in der Vorbereitungsphase geklärt. Das umfasst auch ggf. gesetzliche Regelungen. Wird das Audit im Rahmen einer *Compliance-Prüfung* durchgeführt, so müssen weitere Rahmenbedingungen und formale Anforderungen erfüllt werden, die vorab zu klären sind. »Compliance« bedeutet Regelkonformität und umfasst die Einhaltung von Gesetzen und Richtlinien. Diverse Unternehmen und Organisationen sind bestimmten Gesetzen unterworfen, die eine entsprechende regelmäßige Prüfung erfordern.

## Abschluss und Dokumentation

Während ein echter Angreifer zufrieden ist, wenn er das System kompromittiert und seine Ziele (Datendiebstahl, Sabotage etc.) erreicht hat, muss der Pentester den Auftraggeber bestmöglich unterstützen, um die gefundenen Schwachstellen zu erkennen und zu beseitigen. Daher wird ein umfangreicher Bericht über die Sicherheitslücken, Gefährdungen und Risiken erstellt und ein Maßnahmen-Katalog erarbeitet, der dem Auftraggeber die mögliche Beseitigung der Schwachstellen aufzeigt.

Dabei wird auch die Vorgehensweise des Pentesters detailliert beschrieben, um dem Auftraggeber darzulegen, wie die Informationsbeschaffung und Ausnutzung der Sicherheitslücken erfolgt ist. Zur Dokumentation eines Penetrationstests existieren diverse Tools und Hilfsmittel, die eine Datenbank-gestützte Auswertung ermöglichen. Auf die Details hierzu gehen wir in Kapitel 32 *Durchführen von Penetrationstests* am Ende des Buches ein.

## Was ein Pentester nicht macht

Im Rahmen eines Audits wird ein Pentester in der Regel nicht versuchen, sich im System festzusetzen, um zu einem späteren Zeitpunkt erneut in das System einzubrechen. Andererseits ist es natürlich durchaus sinnvoll, zu testen, wie weit der Angreifer kommen würde, um *Backdoors* und andere Schwachstellen zu platzieren. Diese werden jedoch im Rahmen eines Audits in der Regel nicht installiert, um sie später tatsächlich zu nutzen – es bleibt meistens beim »Proof-of-Concept«, also beim Ausloten der Möglichkeiten.

Darüber hinaus wird ein Pentester in der Regel auch keine aggressiven Techniken einsetzen, um seine Spuren zu verwischen. Dies erfordert eine Manipulation diverser wichtiger Subsysteme von Produktivsystemen, einschließlich des Einsatzes von Rootkits, die es ermöglichen, auf Kernel-Ebene elementare Prozesse und Dateien zu manipulieren und zu verstecken.

Dahinter steckt nicht zuletzt die Philosophie, dass die Systeme des Auftraggebers getestet und anschließend *gehärtet* (also sicherer gemacht) werden sollen, nicht jedoch als Spielwiese eines Hackers dienen sollen, um zu schauen, was alles geht. Das gezielte Schwächen eines Produktiv-Systems führt unter Umständen zur Notwendigkeit einer Neuinstallation und ist ein »No-Go« für einen Pentester.

### Tipp: Bleiben Sie neugierig und testen Sie Ihre Grenzen aus!

Damit wir uns nicht falsch verstehen: Wir fordern Sie geradezu auf, an die Grenze Ihrer Fähigkeiten zu gehen! Innerhalb Ihres Labornetzes sollten Sie alles, was irgendwie möglich erscheint, umsetzen und ausprobieren – hier sind Ihnen keine Grenzen gesetzt – virtuelle Maschinen und Snapshots machen es möglich.

Stellen Sie jedoch sicher, dass die von Ihnen angegriffenen Systeme vollständig unter Ihrer eigenen Kontrolle sind und keinerlei Produktivzwecken dienen! In Ihrem abgeschotteten Labor können Sie so viel herumexperimentieren, wie Sie wollen. Aber halten Sie strikt die Regeln ein, wenn Sie ein anderes Netzwerk oder Computersystem im Rahmen eines beauftragten Penetrationstests hacken.

Grundsätzlich gibt es auch spezielle Szenarien, in denen ein Pentester aggressiver vorgeht und bestimmte Black-Hat-Methoden anwendet, wie beispielsweise die Installation einer Backdoor. Dies hängt immer von der Zielstellung bzw. Auftragsformulierung ab. Unter dem Strich muss dies jedoch abgesprochen sein und dem Gesamtziel der Verbesserung der IT-Sicherheit dienen.

## 1.8 Praktische Hacking-Beispiele

In diesem letzten Abschnitt des Kapitels möchten wir Ihnen noch drei erfolgreiche Hacking-Angriffe vorstellen, um gleich einmal etwas »Praxis« einzubringen und Ihnen eine Vorstellung von »Real-World-Hacks« zu geben.

### 1.8.1 Angriff auf den Deutschen Bundestag

Am 13. April 2015 wurde ein Angriff auf das Netzwerk des Bundestages bekannt, bei dem diverse, teilweise als *Top Secret* eingestufte, Dokumente gestohlen wurden. Offensichtlich haben sich die Hacker Zugang zu einem Großteil der Systeme des Bundestages verschaffen können, sodass zum einen nicht im Detail nachvollziehbar ist, welche Informationen entwendet und welche Systeme kompromittiert wurden. Zum anderen wurde es dadurch notwendig, einen erheblichen Teil der IT-Infrastruktur neu aufzusetzen, um wieder Vertrauen in die Systeme haben zu können.

Nach den Analysen ist zunächst ein einzelner Computer eines Abgeordneten durch eine E-Mail mit entsprechendem Malware-Anhang oder einem *Drive-by-Download* (ein Schadcode wird automatisch beim Besuchen einer bestimmten Website unbemerkt im Hintergrund heruntergeladen) infiziert worden. So hatten die Angreifer vermutlich eine *Backdoor* (also eine Hintertür im System) installiert, über die sie Zugang zum Opfer-System erlangten.

Von dort aus gelang es den Angreifern mittels gängiger Open-Source-Software (namentlich *mimikatz*, siehe Kapitel 10 *Password Hacking*), Zugriff auf Administrator-Accounts zu erlangen, die ihnen wiederum Zugang zu diversen Systemen des Netzwerks ermöglichten und dazu führten, dass sich die Angreifer frei im Netzwerk des Bundestages bewegen konnten.

Interessant hierbei ist, dass bis zu diesem Zeitpunkt niemand wirklich reagierte: Obwohl sich einige Systeme merkwürdig verhielten, nahm man die Situation noch nicht so richtig ernst. Erst als ausländische Geheimdienste mitteilten, dass ein derartiger Angriffsplan entdeckt wurde, sind die entsprechenden Stellen, unter anderem das *Bundesamt für Sicherheit in der Informationstechnik* (BSI) involviert worden, um die Sachverhalte aufzuklären.

Das Verblüffende hierbei ist, dass die Angreifer bereits bekannte Schwachstellen und Hacking-Tools eingesetzt haben. Es muss sich also keineswegs um versierte Hacker gehandelt haben – stattdessen wäre es erschreckenderweise auch denkbar, dass hier Scriptkiddies (zugegebenermaßen mit deutlich erweiterten Kenntnissen) am Werk waren!

Unter dem Strich bleibt die Erkenntnis, dass das Netzwerk des Bundestages zum einen unzureichend geschützt war und zum anderen das Sicherheitsbewusstsein der Administratoren ganz offen-

sichtlich nicht ausreichte, um die (durchaus vorhandenen) Symptome des Angriffs rechtzeitig zu erkennen und entsprechend zu handeln. Aufgrund dieser Umstände war es sogar mit relativ einfachen Mitteln und Open-Source-Standard-Tools möglich, derart tief in das Netzwerk des Bundestages einzudringen und sich dort festzusetzen.

### 1.8.2 Stuxnet – der genialste Wurm aller Zeiten

Im krassen Gegensatz zum Angriff auf den Bundestag wurde 2010 ein Computerwurm entdeckt, der als *Stuxnet* bekannt wurde. Es handelt sich um den höchstentwickelten Wurm, der jemals gefunden wurde. Er nutzt eine Vielzahl von Schwachstellen und kann sogar, wie ein normales Programm, automatisch über das Internet aktualisiert werden.

*Stuxnet* wurde speziell für den Angriff auf *Siematic S7* entwickelt. Dabei handelt es sich um ein Steuerungssystem der Firma Siemens, das vielfach in verschiedenen Industrieanlagen, wie z.B. Wasserkörpern, Pipelines oder aber auch Urananreicherungsanlagen eingesetzt wird.

Letztere schienen auch das Ziel von *Stuxnet* zu sein, da zunächst der Iran den größten Anteil an infizierten Computern besaß und die Anlagen des iranischen Atomprogramms von Störungen betroffen waren. Durch die Störung der Leittechnik dieser Anlagen sollte wohl die Entwicklung des Atomprogramms gestört und verzögert werden.

Die Entwickler und Auftraggeber von *Stuxnet* sind bis heute nicht bekannt – selbstverständlich gibt es diverse Gerüchte und Indizien, die an dieser Stelle aber nicht von Belang sind. Entscheidend ist, dass hier kein einzelner Hobbyprogrammierer oder Scriptkiddie am Werk war, sondern eine hochversierte Gruppe professioneller Entwickler. Die Komplexität von *Stuxnet* legt die Vermutung nahe, dass hier hochspezialisierte Experten an der Arbeit waren und die Entwicklung des Wurms mehrere Monate professioneller Projektarbeit erforderte.

#### Hinweis: Zusatzmaterial zum Buch online

Mehr Informationen über *Stuxnet* haben wir in einem Dokument zusammengefasst und zum Download unter [www.hacking-akademie.de/buch/member](http://www.hacking-akademie.de/buch/member) bereitgestellt. Bitte nutzen Sie das im Vorwort genannte Passwort für den exklusiven Zugang zum Mitglieder-Bereich unserer Leser.

### 1.8.3 Angriff auf heise.de mittels Emotet

Auch Malware entwickelt sich weiter und ein neuer Meilenstein in der Evolution war *Emotet*. Dabei handelt es sich um einen sogenannten Banking-Trojaner. Derartige Schadsoftware ist darauf spezialisiert, Zugangsdaten von Online-Banking-Diensten auszuspähen. *Emotet* ist jedoch erheblich vielseitiger und leistungsfähiger als die meisten derartigen Schadprogramme und wird zudem aktiv weiterentwickelt.

Seit 2018 ist *Emotet* in der Lage, auch lokale E-Mails auszulesen und somit selbst Mails zu generieren, die scheinbar von bekannten Absendern kommen, mit denen das Opfer kürzlich bereits in Kontakt stand. Durch glaubwürdige Inhalte wird der Benutzer dazu verführt, schädliche Dateianhänge zu öffnen oder auf Links zu klicken, die zu infizierten Servern führen, wodurch sogenannte *Drive-by-Downloads* initiiert werden. Diese automatischen Downloads nutzen Browserlücken aus und platzieren Schadcode auf dem Computer des Opfers.

Im Mai 2019 wurde das bekannte Online-Magazin *heise.de* Opfer von *Emotet*. Es handelte sich um einen ausgeklügelten, mehrstufigen Angriff, der von *heise* vorbildlich und transparent aufgearbeitet wurde. Die detaillierten Untersuchungsergebnisse wurden veröffentlicht. Sie können unter

[www.heise.de/ct/artikel/Trojaner-Befall-Emotet-bei-Heise-4437807.html](http://www.heise.de/ct/artikel/Trojaner-Befall-Emotet-bei-Heise-4437807.html) den gesamten Vorfall in allen Details nachlesen.

## 1.9 Zusammenfassung und Prüfungstipps

Werfen wir einen kurzen Blick zurück: Was haben Sie gelernt, wo stehen Sie und wie geht es weiter?

### 1.9.1 Zusammenfassung und Weiterführendes

Sie haben in diesem Kapitel gelernt, was es mit dem Begriff »hacking« bzw. »Hacker« auf sich hat, und haben festgestellt, dass wir hier durchaus genau unterscheiden müssen, z.B. zwischen *Script-kiddie*, *White Hat*, *Grey Hat* und *Black Hat* bzw. dem *Cracker*. Weiterhin haben wir Motive und Ziele von Hacking-Angriffen beleuchtet.

Ein ganz elementares Konzept, das Sie sich unbedingt zu Eigen machen sollten, ist das »Ethical Hacking«. Hierbei geht es darum, als White Hat Hacker die Kunst des Hackings einzusetzen, um die Sicherheit von Computersystemen und -netzwerken zu verbessern. Wenn Sie die Zukunft Ihrer Karriere im Ethical Hacking sehen, dann sollten Sie sich überlegen, die Prüfung zum *Certified Ethical Hacker* zu absolvieren.

Es ist wichtig, beide Seiten zu berücksichtigen. Daher haben wir vorübergehend einen Perspektiv-Wechsel vorgenommen und betrachtet, welche Schutzziele es gibt und wie sie von den IT-Sicherheitsbeauftragten verfolgt werden. Der Abkürzung *CIA* stehen die englischen Begriffe *Confidentiality* (Vertraulichkeit), *Integrity* (Integrität) und *Availability* (Verfügbarkeit) gegenüber. Dazu kommt in manchen Betrachtungen noch die *Authenticity* (Authentizität) bzw. die *Non Repudiation* (Nichtabstreitbarkeit). Beides wird aber häufig auch unter der Integrität zusammengefasst. Die Herausforderung für einen IT-Sicherheitsbeauftragten ist die Sicherstellung der Schutzziele einerseits, ohne andererseits die Benutzerfreundlichkeit und die Funktionalität zu stark einzuschränken – sonst heißt es am Ende: »Operation gelungen, Patient tot!«

Wird das *White Hat Hacking* im Rahmen eines abgesprochenen Audits durchgeführt, so nennt sich dieser Prozess *Penetrationstest*, oder in der englischen Form: *Penetration Test* bzw. kurz: *Pentest*. Dabei werden die Computersysteme und/oder das Netzwerk des Auftraggebers nach detaillierter Absprache systematisch auf Schwachstellen untersucht. Hierzu bedient sich der Pentester professioneller Hacking-Methoden.

In diesem Zusammenhang haben Sie die Phasen eines Hacking-Angriffs kennengelernt, die aus dem *Ausspähen* (Reconnaissance), dem *Finden von Schwachstellen* (Scanning), dem *aktiven Eindringen* (Gaining Access), dem *Festsetzen im Opfer-System* (Maintaining Access) sowie der *Verwischung der Einbruchsspuren* (Clearing Tracks) besteht. Im Rahmen eines Pentests werden einige der Phasen angepasst, da es hier insbesondere um das Aufzeigen und Dokumentieren von Schwachstellen geht.

### 1.9.2 CEH-Prüfungstipps

In diesem ersten Kapitel sind schon einige wichtige Begriffe und Konzepte enthalten, die in der Prüfung abgefragt werden können. Hierzu zählen die unterschiedlichen Hackertypen, die Schutzziele und die Phasen eines Hacking-Angriffs. Stellen Sie sicher, dass Sie Hacking-Aktivitäten den einzelnen Phasen zuordnen können und dass Sie verstanden haben, welche Schutzziele durch bestimmte Maßnahmen sichergestellt bzw. bedroht werden. Letzteres werden Sie im Laufe dieses Buches immer wieder gegenüberstellen können.

### 1.9.3 Fragen zur CEH-Prüfungsvorbereitung

Mit den nachfolgenden Fragen können Sie Ihr Wissen überprüfen. Die Fragestellungen sind teilweise ähnlich zum CEH-Examen und können daher gut zur ergänzenden Vorbereitung auf das Examen genutzt werden. Die Lösungen zu den Fragen finden Sie in Anhang A.

1. Welcher Hacker-Typ hat beschränkte oder kaum Kenntnisse im Security-Bereich und weiß lediglich, wie einige einschlägige Hacking-Tools verwendet werden?
  - a) Black Hat Hacker
  - b) White Hat Hacker
  - c) Scriptkiddie
  - d) Grey Hat Hacker
  - e) Cracker
2. Welche der im Folgenden genannten Phasen ist die wichtigste Phase im Ethical Hacking, die häufig die längste Zeitspanne in Anspruch nimmt?
  - a) Gaining Access
  - b) Network Mapping
  - c) Privilege Escalation
  - d) Footprinting
  - e) Clearing Tracks
3. Ein CEH-zertifizierter Ethical Hacker wird von einer Freundin angesprochen. Sie erklärt ihm, dass sie befürchtet, ihr Ehemann würde sie betrügen. Sie bietet dem Ethical Hacker eine Bezahlung an, damit er in den E-Mail-Account des Freundes einbricht, um Beweise zu finden. Was wird er ihr antworten?
  - a) Er lehnt ab, da der Account nicht der Freundin gehört.
  - b) Er sagt zu, da der Ehemann unethisch handelt und die Freundin Hilfe benötigt.
  - c) Er sagt zu, lehnt aber die Bezahlung ab, da es sich um einen Freundschaftsdienst handelt.
  - d) Er lehnt ab und erklärt der Freundin, welcher Gefahr sie ihn damit aussetzt.
4. Die Sicherheitsrichtlinie (Security Policy) definiert die Grundsätze der IT-Security in der Organisation. Für einige Bereiche gibt es ggf. Sub-Policies, wie z.B. Computer-Sicherheitsrichtlinie, Netzwerk-Sicherheitsrichtlinie, Remote-Access-Richtlinie etc. Welche drei der im Folgenden genannten Ziele sollen damit sichergestellt werden?
  - a) Availability, Non-repudiation, Confidentiality
  - b) Authenticity, Integrity, Non-repudiation
  - c) Confidentiality, Integrity, Availability
  - d) Authenticity, Confidentiality, Integrity
5. Welcher Phase eines Hacking-Angriffs kann die Installation eines Rootkits zugerechnet werden?
  - a) Reconnaissance
  - b) Scanning
  - c) Gaining Access
  - d) Maintaining Access
  - e) Clearing Tracks

# Stichwortverzeichnis

- 6LoWPAN 1129
- A**
- Access Control List (ACL) 719
  - Active Directory (AD) 310, 969
  - Active Discovery 221, 256
  - Acunetix 864
  - Address Resolution Protocol (ARP) 259, 643
  - Address Space Layout Randomization (ASLR) 1023
  - Ad-hoc-Netzwerk (WLAN) 1036
  - ADS 490
  - Advanced Message Queuing Protocol (AMQP) 1130
  - AdwCleaner 522
  - airbase-ng 1073, 1075
  - aircrack-ng 1045, 1057
  - AirDroid 1091
  - aireplay-ng 1054, 1056
  - Airgeddon 1075
  - airodump-ng 1049, 1056, 1058, 1062
  - Ajax 850
  - Alternate Data Stream 490
  - Amplifying Attack 808
  - Android 1084
  - Android Debug Bridge (ADB) 1097
  - Android x86 1093
  - Angler 472
  - Angriffsphasen 58
  - Anonymizer 137
  - Anonymous 46
  - Antivirus-System (AV) 473
  - Any Run 507
  - apache2 469
  - Apache-Webserver 851
  - App 1085
  - ARP-Cache-Poisoning 643
  - ARP-Inspection 669
  - ARP-Spoofing 643
    - arp spoof 650
  - ASP.net 850
  - Asymmetrische Verschlüsselung 175
    - Authentizitätsprüfung 178
    - Diffie-Hellman-Schlüsselaustausch 179
    - Digital Signature Algorithm (DSA) 181
    - Elgamal 180
    - Private Key 176
    - Public Key 176
    - Public-Key-Authentifizierung 178
- Rivest Shamir Adleman (RSA) 180
- Schlüsselaustausch 176
- auditpol 568, 572
- Audit Policies (Windows) 567
- Ausführen-Recht 107
- Autoruns 525
- Autostart-Eintrag 524
- AV-Signatur 473
- AWS 1164
- Azure (Microsoft) 1164
- B**
- Backdoor 415, 455
  - BackTrack 72
  - Bad Character 1010
  - Baseband-Hack 1088
  - Bash 106, 414
  - Bash Bunny 786
  - Beacon Frame (WLAN) 1040
  - Best(er) Keylogger 493
  - Bettercap 667
  - Bildschirmauflösung 104
  - Bind-Shell 421
  - Black-Box-Test 1194
  - Black Hat 43
  - Blackhole Exploit Kit 472
  - Blind Hijacking 685
  - BlueBorne 1141
  - Bluebugging 1090
  - Bluejacking 1090
  - Bluesnarfing 1090
  - BlueStacks 1092
  - Blue Teaming 1195
  - Bluetooth Low Energy (BLE) 1128
  - Boot-Sektor-Virus 460
  - Botnet 456
  - Botnet 815
  - Bricking 814
  - Bring Your Own Device (BYOD) 1113
  - Browser in the Box (BitBox) 538
  - Brute-Force-Angriff 389
  - BSS (Basic Service Set) 1036
  - BSSID (Basic Service Set Identifier) 1040
  - btmp 585
  - Buffer Overflow (Pufferüberlauf) 993
  - Bug-Bounty-Programm 883
  - BulkFileChanger 579

bully (WPS-Cracking) 1063

Burp Suite 692

Proxy 693

Sequencer 697

## C

c99 (Webshell) 987

C/C++ (Buffer Overflow) 994

Cain & Abel 402

Capسا 517

Captive Portal (WLAN) 1069

Capture 600

Cavity Virus 461

CCleaner 158, 457, 522, 582

CEHv11-Prüfung 50

CeWL 396

CGI 850

ChameleonMini 795

chmod 108

chntpw 372

CIFS 295

Clear\_Event\_Viewer\_Logs.bat 575

Cloud 1159

CloudGoat 1186

Clustering 55

cmd.exe 415

Colasoft Packet Builder 285

Command-Injection 685, 965

Community Cloud 1163

Community-String 304

Companion-Virus 461

Compliance 1193

Computervirus 454, 455

Computerwurm 455, 461

Config-Register (Cisco) 375

Constrained Application Protocol (CoAP) 1130

Contentfilter 721

Contiki 1126

Cookies 845

Covert Channel 486

Crazyradio PA 793

Credential Scan 349

Credential Stuffing 893, 924

Cross-Site-Scripting (XSS) 709, 910

Crunch 394, 1052

Crypter 507

Cryptojacking 1183

Crypto-Mining 1183

CrypTool 166

CSRF (Cross-Site-Request-Forgery) 917

CSS 850

Cuckoo 538

CurrPorts 516

Custom-Recovery 1098

Custom-ROM (Android) 1095

CVE 332

Cyber-Terrorist 44

## D

Dander Spritz 573

Darknet 147

Data Execution Prevention (DEP) 1024

Datei

anzeigen 114

finden 115

Dateimanager 101

Dateisignaturverifizierung 530

Datei-Virus 460

Deauthentication Attack (WLAN) 794, 1053

Debugger 996

Decompiler 504

Deep Web 147

Defacing 44

Default-Passwörter 366

Denial-of-Service-Angriff (DoS-Angriff) 804

DHCP-Snooping 669

DHCP-Spoofing 647

Dictionary-Angriffe 390

Dienst

prüfen 526

verwalten 119

Diffie-Hellman-Schlüsselaustausch *siehe* Asymmetrische Verschlüsselung

Digispark Development Board 788

Digitale Signatur 56

DirBuster 861

Directory-Traversal-Angriff 853, 983

Disassembler 504

diskpart 370

DistCC (Schwachstelle) 426

Distributed-Denial-of-Service-Angriff (DDoS-Attacke) 515, 804

Distributed-Reflected-DoS-Angriff (DRDoS) 814

DMZ 723

DNS-Amplification-Angriff 814

DNS-Cache-Poisoning 644

DNS-Footprinting 233

DNS-Hijacking 645

DNS-Injection 645

DNS over TLS 669

DNSQuerySniffer 519

dnsrecon 321

DNSSEC 669

dnsspoof 653

DNS-Spoofing 644

Domain Name System (DNS) 319, 644

DOM-Interface 708

Drive-by-Download 457, 768

DriverView 528

Dropbox 1160

Dropper 454

dsniff (Tool) 648, 656

Dumpster Diving 363

DVWA 886

**E**

Eavesdropping 634  
 EAX, EBX, ECX und EDX (Stack Register) 996  
 EBP (Stack Pointer) 996  
 EICAR 510  
 EIP (Stack Pointer) 996  
 Elektronische Unterschrift 56  
 E-Mail-Footprinting 237  
 Empire-Framework 442  
     Agents 447  
     Listener 444, 462  
     Module 447  
     Stager 445, 462  
 Encoder 476  
 Encryption Code 461  
 Entropie 699  
 Entry Point 966  
 enum4linux 299  
 Enumeration 218, 293  
     NetBIOS 294  
     SMB 294  
 Ereignisanzeige 566  
 ESS (Extended Service Set) 1037  
 ESSID (Extended Service Set Identifier) 1040  
 Etcher 796  
 Ethereal 601  
 Ethical Hacking 1192  
 Ettercap 657, 1071, 1074  
 Evasion (IDS/IPS) 736  
 eventlogedit 573  
 eventvwr.exe 566  
 evilginx2 776  
 Evil Twin (WLAN) 1074  
 Exploit 332, 350, 433  
 Exploit-Database 229  
 Exploit Kit 472  
 Exposure *siehe* Vulnerability  
 Extensible Markup Language (XML) 847

**F**

False Positives 349  
 Fastboot 1098  
 FCIV (File Checksum Integrity Verifier) 531  
 Federation Services 1184  
 FGDump 383  
 Fingerabdruck-Scan 362  
 Firewalking 726  
 Firewall 717  
     Application Layer Gateway 721  
     Contentfilter 721  
     Deep Packet Inspection 722  
     Failover/Cluster 725  
     iptables 720  
     Netzwerk-Firewall 718  
     Paketfilter-Firewall 719  
     Perimeterschutz 719  
     Personal-Firewall 718  
     Proxy-System 721

**S** Stateful Inspection 720  
 UTM-Lösung 723

FISMA 1198  
 Footprinting 218  
 FoxyProxy 136  
 FQDN 234  
 Fragmentation-Angriff 811  
 Fragmentierung 739  
 Framegrabber 790  
 Freenet-Netzwerk 153  
 fsutil 579  
 FTP-Zugangsdaten ermitteln 616  
 Fuzzing 1002

**G**

Gerätetreiber prüfen 528  
 Gesichtsscan 362  
 GHDB *siehe* Google Hacking Database  
 Golden Ticket 1184  
 Google Cloud Platform 1164  
 Googledork 228  
 Google-Hacking 227  
 Google Hacking Database 229  
 gpedit.msc 567  
 Gqrx 1138  
 Greenshot 1207  
 Grey-Box-Test 1194  
 Grey Hat 44  
 Gruppenrichtlinienverwaltungs-Editor 567  
 G-Zapper 159

**H**

Hacker-Paragraf 48  
 HackRF One 1137  
 Hacktivist 44  
 Handler 707  
 Hard Brick 1095  
 Hash-Algorithmen 181  
     Bcrypt und Scrypt 187  
     Integritätsprüfung 182  
     Kryptologische Hashfunktionen 185  
     Message Digest 5 (MD5) 186  
     Passwort-Hashfunktionen 185  
     PBKDF2 186  
     Prüfsummen 185  
     Secure Hash Algorithm (SHA) 186  
 Hash Injection Attack 392  
 Hash Suite 400  
 Hashwert 54  
 Haveibeenpwned (Website) 390  
 Heap-Buffer-Overflow-Angriff 1021  
 Heap Spraying (Heap Overflow) 1022  
 Heartbleed-Angriff 204  
 Hidden Field (HTML-Formular) 973  
 High Availability 55  
 HijackThis 523  
 HIPAA 1197  
 Honeypot 741

- hosts (Datei) 518, 523, 646  
Hotspot 1033  
hping3 283, 823  
HTML 850  
HTTP 842  
    CONNECT 845  
    DELETE 845  
    GET 844  
    HEAD 845  
    Host-Header-Wert 843, 852  
    PATCH 845  
    POST 844  
    PUT 845  
    User-Agent 843  
HTTprint 858  
HTTTrack 240, 863  
Hub 600, 638  
Hub-Modus (Switch) 639  
Human Hacker 756  
Hunt (Session Hijacking) 685  
Hybrid Cloud 1164  
Hydra 405  
Hyperion 479  
Hypertext Transfer Protocol (HTTP) 842  
Hyper-V 69
- I
- IBSS (Independent Basic Service Set) 1036  
ICMP 260, 643  
ICMP-Flood-Angriff 806  
ICMP-Tunneling 487  
Identity and Access Management (IAM) 1182  
IDOR (Insecure Direct Object References) 902  
IDS (Intrusion-Detection-System)  
    Hostbasiertes IDS (HIDS) 729  
    Netzwerkbasiertes IDS (NIDS) 729  
IEEE 802.11 1035  
IEEE 802.15.4 1129  
IIS 853  
Immunity Debugger 1000  
IMSI-Catcher 1090  
Informationsbeschaffung 59  
Infrared Data Association (IrDA) 1128  
Infrastructure as a Service (IaaS) 1161  
Injection-Angriff 929  
Internes Netzwerk 91  
Internet Information Services (IIS) 853  
Internet of Everything 1125  
Internet of Things (IoT) 1123  
Internet Protocol (IPv4) 259  
Intrusion-Detection-System (IDS) 533  
iOS (Apple) 1084  
IPS (Intrusion-Prevention-System) 730  
IPsec 198  
    Authentication Header (AH) 198  
    Encapsulation Security Payload (ESP) 198  
    Internet Key Exchange (IKE) 199
- J
- Iris-Scan 362  
ISO/IEC 27001 und 27002 1198
- K
- Kali Linux 72  
    Einstellungen 103  
    Netzwerk-Konfiguration 121  
    Systemsprache ändern (Xfce) 80  
    Tastatur-Layout (Xfce) 79  
    Update 82  
KARMA-Attacke 794  
Kazam 1206  
KDE 96  
Kerberos 310, 379  
Key Distribution Center 379  
Keylogger 456, 492  
Keystroke-Injection 784  
KFSensor 745  
KillerBee 1142  
Klick Fraud 816  
Kontextmenü 97  
Krypto-Algorithmen 164  
Kryptoanalyse 163, 201, 202  
    Brute Force 202  
    Chosen Ciphertext 203  
    Chosen Plaintext 203  
    Dictionary Attack 201  
    Frequency Analysis 203  
    Known Ciphertext 203  
    Known Plaintext 203  
    Man-in-the-Middle-Angriff (MITM) 203  
    Probable Plaintext 203  
    Rubberhose Attack 203  
    Seitenkanal-Angriff (Side-Channel Attack) 202  
    Timing Attack 202  
    Trickery And Deceit 203  
    Wörterbuchangriff 201  
Kryptografie  
    Algorithmus 165  
    Blockchiffre 168

- Cäsar-Chiffre 168  
 Chiffre 168  
 digitale Signaturen 188  
 Geheimtext 165  
 Klartext 165  
 Poodle-Angriff 205  
 Public Key Cryptography Standards (PKCS) 187  
 Schlüssel 165  
 Stromchiffre 168  
 symmetrische Verschlüsselung 167  
 VeraCrypt 172  
 Vertraulichkeit 167  
 Kryptosystem 164  
 Kryptotrojaner 206
- L**
- Laborumgebung 71  
 LAMP 853  
 Lan Manager (LM) 378  
 LAN Turtle 792  
 Lawful interception 634  
 LDAP 310, 969
  - Common Name 310
  - Distinguished Name 310
  - Organisationseinheit 310
 LDAP Admin 314  
 LDAP-Injection 973  
 libpcap 601  
 Light Fidelity (Li-Fi) 1129  
 Lightweight-Access Point (LAP) 1038  
 LimeSDR 793  
 Linset 1075  
 Linux-Befehle 104  
 Linux-Rechtesystem 106  
 Listener 419, 423, 431  
 Loadbalancing 55  
 Local File Inclusion (LFI) 902, 986  
 Locky 208  
 Logging 565  
 Lokale Sicherheitsrichtlinie 567  
 Long Range Wide Area Network (LoRaWAN) 1129  
 Low Orbit Ion Cannon (LOIC) 828  
 LSASS 383
- M**
- MAC-Adressstabelle 639  
 macchanger 1065  
 MAC-Flooding 639  
 macof 653  
 Magisk 1100  
 Makrovirus 460  
 Maltego 245  
 Malware 454  
 Malware-Analyse 503  
 Management-Report 1207  
 Man-in-the-Browser-Angriff (MIB/MITB) 707  
 Man-in-the-Cloud (MITC) 1178
- Man-in-the-Middle (MITM) 635  
 Man-in-the-Mobile 1088  
 Man-Pages 118  
 Mausezahn 285  
 Maximum Transmission Unit (MTU) 739  
 mdk3 1052, 1054  
 Medusa 403  
 Mesh-Netzwerk (WLAN) 1038  
 Metagoofil 240  
 Metasploit 277
  - Exploit für vsftpd 338
  - Module 279
  - Nmap in Metasploit nutzen 281
  - Webscanning 863
  - WMAP 863
  - Workspaces 279
 Metasploitable 87, 278  
 Meterpreter 432, 581  
 Microdot 549  
 Microsoft 365 1164  
 Microsoft Baseline Security Analyzer 351  
 Mimikatz 441  
 Mirai 819, 1138  
 Mobile Device Management (MDM) 1115  
 Mobile Proxy-Tools 156
  - CyberGhost 157
  - Onion Browser 157
  - OpenDoor 156
  - Orbot 157
  - ProxyDroid 156
  - Psiphon 157
 Mobizen 1091  
 Mona (Immunity Debugger) 1013  
 Most Recently Used (MRU) 575  
 MouseJack-Angriff 794  
 Mouse Jiggler 790  
 MP3Stego 560  
 MQ Telemetry Transport (MQTT) 1130  
 msconfig (Autostart) 524  
 msfconsole 469, 1109  
 msfvenom 437, 468, 1010, 1017, 1108  
 Multihandler 469  
 Multipartite-Virus 460  
 Mutillidae II 884
- N**
- nasm\_shell.rb 1014  
 Nbtscan 296  
 nbtstat 297  
 Ncat 286, 416  
 Ncrack 406  
 Near-Field Communication (NFC) 1129  
 Nessus 339, 864  
 net-Befehle 298  
 NetBEUI 295  
 NetBIOS 294  
 NetBIOS Enumerator 300  
 Netcat 286, 416

- Netcraft 222  
Netsparker 864  
Netstat 516  
net user 370  
Network Address Translation 131  
Netzwerkbrücke 91  
Netzwerkschnittstelle konfigurieren 122  
Netzwerk-Sniffer 599  
Neutrino 472  
Nexpose 345  
Nikto2 350, 865  
NIST 170, 334  
Nmap 263, 296  
    Firewall/IDS Evasion 273  
    Half-Open-Scan 267  
    Host Discovery 264  
    IPv6-Netzwerke scannen 287  
    NSE 275  
    OS Detection 272  
    Ping-Scan 265  
    Ports festlegen 269  
    Reports 274  
    Service Identification 272  
    SYN-Stealth-Scan 267  
    TCP-ACK-Scan 270  
    TCP-Connect-Scan 268  
    TCP-IDLE-Scan 271  
    TCP NULL-, FIN- und Xmas-Scan 271  
    TCP-SYN-Scan 267  
    UDP-Scan 268  
    Vulnerability-Scanning 336  
Webscanning 863  
Zenmap 277  
Noise Jamming 1053  
NOP-Byte 1019  
Notepad++ 1207  
Npcap 601  
nslookup 320, 967  
NTLM 378  
NTP 317  
ntpdc 319  
ntpq 318  
ntptrace 318  
Null-Session 301
- O**  
Obfuscator 507  
Obfuscating 478  
onesixtyone 308  
OpenLDAP 969  
OpenPuff 560  
OpenSSL 201  
OpenStego 554  
OpenVAS 345  
OSINT 218  
OSI-Referenzmodell 257  
OSSTMM 1201  
OUI (MAC-Adresse) 605
- OWASP 879, 1202  
OWASP Broken Web Application 887  
OWASP Top 10 882, 888
- P**  
Packet Sqirrel 791  
Pacu 1187  
Paketlisten aktualisieren 124  
PAM 385  
Passive Discovery 217  
Pass the Hash (PTH) 392  
passwd (Datei) 385  
Password Guessing 364  
Passwort-Richtlinie 365  
PATH-Variable 414  
pattern\_create.rb 1006  
pattern\_offset.rb 1008  
Payload 432, 454  
    staged 433  
    unstaged 433  
PCI DSS 1197  
Peer-to-Peer-Netzwerk 147  
Penetrationstest 1192  
Penetrationtester 43, 44  
Penetration Testing Execution Standard (PTES) 1202  
Pepper (Passwort-Hashes) 387  
Perimeter-Schutz 535  
Permanenter DoS-Angriff (PDoS) 813  
Personen-Suchmaschine 226  
pestudio 506  
Petya 207  
Pfadangabe 113  
Pharming 766  
Phishing 760, 766  
Phlashing 813  
PHP 850  
Ping 643  
Ping of Death 807  
Pivoting 1182  
Platform as a Service (PaaS) 1161  
Pluggable Authentication Modules 385  
Polymorphic Code 460  
Post-Exploitation 413, 428  
Potential Unwanted Application (PUA) 522  
Potential Unwanted Program (PUP) 522, 530  
Powershell 414, 442  
Printer Exploitation Toolkit (PRET) 1144  
Private Cloud 1163  
Privilegien-Eskalation 413  
Process Explorer 512, 525  
Process Monitor 514  
Programmausführung abbrechen 115  
Promiscuous Mode 92, 600, 604  
Prompt 105  
Proxifier 146  
Proxmark 3 795  
Proxychains 134, 146

Proxys 131  
Arten 132  
Public Cloud 1162  
Public-Key-Infrastruktur (PKI) 190  
Certificate Authority 190  
Digitale Zertifikate 191  
OCSP 196  
Zertifikatsspeicher 192  
Zertifikatssperrenlisten und OCSP 195  
Puffer (Buffer Overflow) 995  
PuTTY 140, 468, 521  
PWDump 383

**R**

Radio-Frequency Identification (RFID) 1129  
Rainbow-Tables 387, 391  
Ransomware 206, 456  
Raspberry Pi 795  
reaver (WPS-Cracking) 1063  
Reconnaissance 59, 218  
Recon-ng 241  
Red Teaming 1194  
REG.exe 578  
RegAssassin 523  
RegCleaner 522  
regedit.exe 520  
Register (Stack) 995  
Registrierungsdatenbank (Windows) 520  
Registrierungs-Editor 520  
Registry 520  
RegScanner 521  
Regshot 521  
Remote File Inclusion (RFI) 987  
Remote Scan 349  
Report  
Management- 1207  
technischer 1207  
Rescue-Disk 508  
REST-API 849  
Retina-Scan 362  
Retire.js 922  
Reverse Engineering 503  
Reverse Proxy 860  
Reverse-Shell 422  
RFCrack 1138  
Rijndael *siehe* Symmetrische Algorithmen  
RIoT 1126  
Risk-Assessment 348  
robots.txt 861  
Rogue Access Point 794, 799  
Rogue DHCP-Server 647  
Rolling Code 1136  
ROMMON-Modus (Cisco) 375  
root 103  
Rooten (Android) 1095  
Rootkit 416, 483  
LKM-Rootkit 484  
Userland-Rootkit 484

XCP 485  
ZeroAccess 485  
Root-Shell 339, 424  
RouterSploit 1144  
rpclient 301  
RST Hijacking 685  
Rsyslog 570  
Rubber Ducky 784

**S**

SafeSEH 1024  
SafetyNet-Service (Android) 1096  
Salt-Wert (Passwort-Hashes) 386  
Samba 295  
SAM-Datenbank 377  
Sample (Malware) 507, 539  
Sandbox 509, 536  
Sandboxie 536  
Sandcat Browser 858  
Sanitizer 919  
Sarbanes-Oxley Act (SOX) 1198  
Scanning 218, 256  
Scareware 456  
Schutzzlassen 52  
Schutzziele 51  
SCP 619  
Scriptkiddie 43  
Scrubbing Center 822  
Searchbot 816  
Seattle Lab Mail (SLmail) 997  
Secure Shell (SSH) 619  
Security Audit 1193  
Security Autorun 525  
Security Policy 542  
SEH Overwrite Protection (SEHOP) 1024  
Service-Manager 527  
Service Set Identifier (SSID) 1039  
Session Fixation-Angriff 710  
Session Hijacking 673  
Active Session Hijacking 675  
Application Level Hijacking 674  
Application Level Session Hijacking 686  
Network Level Hijacking 674  
Passive Session Hijacking 675  
Session-ID 686  
Session Replay-Angriff 710  
Session Token 686  
SFTP 619  
shadow (Datei) 385  
Shebang-Zeile 108  
Sheep-Dipping 534  
Shell 414  
Shellcode 994, 1018  
Shellshock 866  
Shellter 480  
Shodan 224, 1144  
shred 585  
Sicherheitsstufe 52

- Sidejacking 674, 706  
 SIEM-System 336, 348, 589, 729, 923  
 sigverif.exe 530  
 SIM-Lock 1103  
 Skipfish 865  
 Skriptvirus 460  
 slapd 970  
 SlowHTTPTest 813  
 Slowloris 812  
 Smart Home 1124, 1126  
 SMB 294, 295  
 SMiShing 1088  
 SMTP 314  
 Smurf Attack 807  
 Snagit 1207  
 Sniffing 599, 634  
 SNMP 301
  - Community-String 302
  - MIB 302
  - OID 302
  - Trap 304
 snmpwalk 309  
 Snort 732
  - Konfiguration 733
  - Regeln 733
 SNscan 308  
 SOAP 848  
 Social Bot 816  
 Social Engineering 230, 755
  - CEO Fraud 763
  - Computer Based Social Engineering 766
  - Dumpster Diving 765
  - Eavesdropping 764
  - Fake Websites 760
  - Human Based Social Engineering 759, 761
  - Mobile Based Social Engineering 760
  - Pharming 766
  - Phishing 760, 766
  - Piggybacking 765
  - Reverse Social Engineering 760
  - Shoulder Surfing 764
  - Spear Phishing 767, 775
  - Tailgating 765
  - Technical Support Scam 763
  - Vishing 762
  - Whaling 768
 Social-Engineer Toolkit (SET) 770  
 Social-Media-Footprinting 229  
 SOCKS 141
  - Clientkonfiguration 142
  - Dante 142
  - vicSOCK 145
 Software
  - entfernen 126
  - installieren 125
  - suchen 126
 Software as a Service (SaaS) 1162  
 Software Defined Radio (SDR) 793, 1137  
 Source Routing 685  
 Spam Mimic 550  
 Spear Phishing 767  
 Spoofing 675  
 SpyAgent 494  
 Spytech SpyAgent 494  
 Spyware 456, 492  
 SQL 890  
 SQL-Injection 889, 929
  - Blind SQL-Injection 943
  - Boolean SQL-Injection 949
  - Tautology based SQL-Injection 933
  - Time based SQL-Injection 950
 SQLMap 952  
 SSH (Secure Shell) 139, 143
  - PuTTY 140
  - SSH-Server 119
  - TCP-Verbindungen tunneln 139
 SSL 200  
 sslstrip 1074  
 SSL-VPN 200  
 Stack 995  
 Stack Buffer Overflow 993  
 Stack Canary (Stack Cookie) 1024  
 Stack Pointer (SP) 996  
 Stapel 995  
 Steganografie 547
  - Jargon Code 552
  - Least Significant Bits 554
  - Open Code 552
  - Semagramm 551
  - Steganalyse 560
  - Steganogramm 553
 StegoStick 558  
 Stegosuite 557  
 Strings (Sysinternals) 505  
 Stuxnet 63  
 sudo 388  
 Suicide Hacker 44  
 Switch 600, 639  
 Symmetrische Algorithmen 169
  - Data Encryption Standard (DES) 170
  - Rivest Cipher 171
  - Serpent 172
  - Triple-DES (3DES oder DESEde) 170
  - Twofish und Blowfish 171
    - und Rijndael) 170
  - SYN-Cookies 809
  - Syn-Flood-Angriff 808
  - Syslog 568
  - Syslog-ng 570

**T**

- Tails (Linux-Distribution) 154  
 Task-Manager 512, 524  
 TCP 262
  - desynchronized state 680
  - Initial Sequence Number (ISN) 678

- Receive Window 677  
 RST/Reopen 680  
 SACK 807  
 Session Splicing 740  
 Sliding Window 677  
 Window Size 677
- tcpdump 624  
 TCP-Handshake 608  
 TeamViewer (Mobile) 1091  
 TeamWinRecoveryProject (TWRP) 1100  
 Teardrop-Angriff 812  
 Technischer Report 1207  
 Technitium MAC Address Changer 1065  
 Telnet 286, 617  
 Temporal Key Integrity Protocol (TKIP) 1043  
 THC Hydra 405  
 Throwing Star LAN Tap Pro 792  
 Ticket Granting Server 380  
 Ticket Granting Ticket 379  
 Tier (Architektur) 840  
 Timestamp 578  
 TLS 200  
 Tomcat 699  
 Tor-Netzwerk 148
  - DuckDuckGo 149
  - Hidden Wiki 151
  - Onion-Adressen 149
  - Onion-Proxy 148
  - Onion Services 149
- touch 587  
 Tracking-Pixel 130  
 Transparenter Proxy 132  
 Transport Layer Security (TLS 200)  
 Treiber prüfen 528  
 Tripwire 533  
 Trojaner 454, 466
  - Baukasten 471
  - Botnet-Trojaner 466
  - CLI-Trojaner 466
  - Covert-Channel-Trojaner 467
  - destruktive Trojaner 467
  - E-Banking-Trojaner 467
  - FTP-Trojaner 466
  - HTTP/HTTPS-Trojaner 467
  - ICMP-Tunneling-Trojaner 467
  - Proxy-Server-Trojaner 466
  - Remote-Access-Trojaner 467
  - VNC-Trojaner 466
- TShark 627
- U**  
 Überwachungsrichtlinien (Windows) 567  
 U-Boot (Bootloader) 1140  
 Ubuntu Core 1126  
 UDDI 848  
 UDP 261  
 UDP-Flood-Angriff 806
- UDP Hijacking 686  
 UNC (Uniform Naming Convention) 295  
 Uniform Resource Identifier (URI) 687  
 Uniform Resource Locator (URL) 687, 841  
 Universal Asynchronous Receiver Transmitter (UART) 1139  
 Update (Kali Linux) 124  
 USB-Keylogger 783  
 USBNinja 789  
 USB-Sticks infizieren mit SET 775  
 Use-after-free (Heap Overflow) 1022  
 UserLAND (App) 1104  
 UTF-8 842
- V**  
 Veil-Framework 480  
 VeraCrypt 172  
 Verzeichnis 113  
 VideoGhost 790  
 Viren-Baukasten 471  
 Virencheck 508  
 VirtualBox 69, 70
  - Gasterweiterungen 81
  - Hostkey 81
  - Netzwerk-Konfiguration 90
  - Sicherungspunkt 82
  - Snapshot 82
- Virtualisierung (Cloud) 1165  
 Virtualisierungssoftware 69  
 Virtual Private Network (VPN) 137, 197
  - IPsec 137
  - IPsec-VPN 198
  - OpenVPN 137
  - Remote-Access-VPN 198
  - Site-to-Site-VPN 198
  - SSL-VPN 198
  - VPN-Anbieter 138
  - VPN-Gateway 137
- Virus 459  
 Virus Maker 471  
 VirusTotal 474  
 Vishing 762  
 VMware 69  
 Vulnerability 332  
 Vulnerability Assessment 256, 346, 1193  
 Vulnerability-Scanner 335
- W**  
 Wachstafel (Steganografie) 549  
 WAFW00F 860  
 WannaCry 206  
 Wardriving 1040  
 wash (WiFi-Scanning) 1062  
 Watering-Hole-Angriff 769  
 WayBack Machine 223  
 WDS (Wireless Distribution Set) 1038  
 Wearables 1124

- Web Application Firewall (WAF) 860  
Web Bug 130  
Webcrawler 816  
WebDAV 849  
Web-Hacking 839  
WebInspect 864  
Web Security Dojo 887  
Webserver 840, 851  
Webshell 986  
Website-Footprinting 239  
Web Spider (Web Crawler) 861  
Web Vulnerability Scanner (WVS) 864  
WebWolf 884  
weevely 986  
WEP (Wired Equivalent Privacy Protocol) 1042  
wetutil.exe 575  
Whaling 768  
White-Box-Test 1194  
White Hat 43  
White Hat Hacking 1192  
Whois 231  
Wi-Fi Alliance 1036  
WiFiKill 1105  
wifiphisher 1075  
WiFi Pineapple 794, 1074  
Win32DiskImager 796  
Windows 10 83  
Windows 7 83  
Wine 479  
WinPcap 601  
Wireless Access Point (AP) 1036  
WirelessKeyView 1067  
Wireless LAN (WLAN) 1033  
    Frequenzen 1034  
    Honeypot 794  
    Phishing 1074  
    Sniffing 640  
Wireshark 518, 599  
    Anzeigefiltern 612  
    Capture Filter 607  
    Display Filter 607, 612  
Ncap 601  
Pcap 601  
Wiretapping 634  
WordPress 869  
Wörterbuch-Angriffe 390  
Wortlisten (Passwort-Hacking) 390  
WPA 1043  
WPA (Wi-Fi Protected Access) 1043  
WPA/WPA2-Angriff 1058  
WPS (Angriff) 1061  
WPS (Wi-Fi Protected Setup) 1044  
WPScan 874  
Wrapper 468  
WS-\* 848  
WSDL 848  
wtmp 585  
Wurm 455, 461
- X**
- XAMPP 853  
XEN 69  
Xfce 96  
XML 847  
XML-Entity 899  
XSS 910  
XXE (XML External Entities) 899
- Z**
- zAnti 1104  
Zed Attack Proxy (ZAP) 880  
Zeitstempel 578  
Zeitzone einstellen 98  
Zenmap 277  
Zephyr 1126  
Zero-Day-Exploit 350  
ZigBee 1129, 1142  
Zombie (Botnetze) 817  
Zwei-Faktor-Authentifizierung (2FA) 362  
Zwiebel-Routing (Tor) 148