



mitp

Jürgen
Ebner

4. Auflage

Einstieg in

Kali Linux

Penetration Testing und
Ethical Hacking mit Linux

Inhaltsverzeichnis

Einleitung	13
Warum Kali Linux?	13
Über dieses Buch	15
Teil I Grundlagen von Kali Linux	17
<hr/>	
1 Einführung	19
1.1 Unterschied zwischen Kali und Debian	19
1.2 Ein Stück Geschichte	19
1.3 Kali Linux – für jeden etwas	22
1.3.1 Varianten von Kali Linux	23
1.4 Die Hauptfeatures	25
1.4.1 Live-System	27
1.4.2 Ein maßgeschneiderter Linux-Kernel	29
1.4.3 Komplett anpassbar	29
1.4.4 Ein vertrauenswürdiges Betriebssystem	31
1.4.5 Auf einer großen Anzahl von ARM-Geräten verwendbar	31
1.5 Richtlinien von Kali Linux	32
1.5.1 Benutzer ohne root-Rechte	32
1.5.2 Netzwerkdienste sind standardmäßig deaktiviert	32
1.5.3 Eine organisierte Sammlung von Tools	33
1.6 Zusammenfassung	33
2 Linux-Grundlagen	35
2.1 Was ist Linux und wie funktioniert es?	35
2.1.1 Hardwaresteuerung	37
2.1.2 Vereinheitlichtes Dateisystem	38
2.1.3 Prozesse verwalten	39
2.1.4 Rechtemanagement	40
2.2 Die Kommandozeile (Command Line)	41
2.2.1 Wie komme ich zur Kommandozeile?	41
2.2.2 Verzeichnisbaum durchsuchen und Dateien verwalten	42

2.3	Das Dateisystem.	44
2.3.1	Dateisystem-Hierarchie-Standard	44
2.3.2	Das Home-Verzeichnis des Anwenders	45
2.4	Hilfreiche Befehle	46
2.4.1	Anzeigen und Ändern von Text-Dateien.	46
2.4.2	Suche nach Dateien und innerhalb von Dateien	46
2.4.3	Prozesse verwalten	47
2.4.4	Rechte verwalten.	47
2.4.5	Systeminformationen und Logs aufrufen.	51
2.4.6	Hardware erkennen	52
2.5	Zusammenfassung	53
3	Installation von Kali.	57
3.1	Systemanforderungen	57
3.2	Erstellen eines bootfähigen Mediums	58
3.2.1	Herunterladen des ISO-Images.	58
3.2.2	Kopieren des Images auf ein bootfähiges Medium	59
3.2.3	Aktivieren der Persistenz auf dem USB-Stick	62
3.3	Stand-Alone-Installation	64
3.3.1	Partitionierung der Festplatte	70
3.3.2	Konfigurieren des Package Managers (apt)	77
3.3.3	GRUB-Bootloader installieren	79
3.3.4	Installation abschließen und neu starten	81
3.4	Dual-Boot – Kali Linux und Windows	81
3.5	Installation auf einem vollständig verschlüsselten Dateisystem	85
3.5.1	Einführung in LVM	85
3.5.2	Einführung in LUKS	85
3.5.3	Konfigurieren verschlüsselter Partitionen	86
3.6	Kali Linux auf Windows Subsystem for Linux.	91
3.6.1	Win-KeX	94
3.7	Kali Linux auf einem Raspberry Pi.	95
3.8	Systemeinstellungen und Updates.	98
3.8.1	Repositories.	98
3.8.2	NVIDIA-Treiber für Kali Linux installieren	99
3.8.3	Terminal als Short-Cut (Tastenkombination).	100
3.9	Fehlerbehebung bei der Installation.	101
3.9.1	Einsatz der Installer-Shell zur Fehlerbehebung.	102
3.10	Zusammenfassung	103

4	Erste Schritte mit Kali	105
4.1	Konfiguration von Kali Linux	105
4.1.1	Netzwerkeinstellungen	106
4.1.2	Verwalten von Benutzern und Gruppen	109
4.1.3	Services konfigurieren	111
4.2	Managing Services.	119
4.3	Hacking-Labor einrichten	121
4.3.1	Kali Linux – Test Lab Environment	123
4.4	Sichern und Überwachen mit Kali Linux	127
4.4.1	Sicherheitsrichtlinien definieren.	127
4.4.2	Mögliche Sicherheitsmaßnahmen	129
4.4.3	Netzwerkservices absichern.	131
4.4.4	Firewall- oder Paketfilterung	131
4.5	Weitere Tools installieren	140
4.5.1	Meta-Packages mit kali-tweaks installieren	140
4.5.2	Terminator statt Terminal	141
4.5.3	OpenVAS zur Schwachstellenanalyse.	142
4.5.4	SSLstrip2.	146
4.5.5	Dns2proxy.	147
4.6	Kali Linux ausschalten.	148
4.7	Zusammenfassung	148
Teil II Einführung in Penetration Testing		151
5	Einführung in Security Assessments	153
5.1	Kali Linux in einem Assessment	155
5.2	Arten von Assessments	156
5.2.1	Schwachstellenanalyse	158
5.2.2	Compliance-Test.	163
5.2.3	Traditioneller Penetrationstest	164
5.2.4	Applikations-Assessment.	166
5.3	Normierung der Assessments	168
5.4	Arten von Attacken	169
5.4.1	Denial of Services (DoS)	170
5.4.2	Speicherbeschädigungen	171
5.4.3	Schwachstellen von Webseiten	171
5.4.4	Passwort-Attacken	172
5.4.5	Clientseitige Angriffe	173
5.5	Zusammenfassung	173

6	Kali Linux für Security Assessments vorbereiten	175
6.1	Kali-Pakete anpassen	175
6.1.1	Quellen finden	177
6.1.2	Build-Abhängigkeiten installieren	180
6.1.3	Änderungen durchführen	181
6.1.4	Build erstellen	185
6.2	Linux-Kernel kompilieren	185
6.2.1	Einführung und Voraussetzungen	186
6.2.2	Quellen finden	187
6.2.3	Kernel konfigurieren	188
6.2.4	Pakete kompilieren und erstellen	191
6.3	Erstellen eines individuellen Kali-Live-ISO-Images	192
6.3.1	Voraussetzungen	193
6.3.2	Erstellen von Live-Images mit verschiedenen Desktop- Umgebungen	194
6.3.3	Ändern der Liste installierter Pakete	195
6.3.4	Verwenden von Hooks zum Optimieren des Live-Images	196
6.3.5	Hinzufügen von Dateien zum ISO-Image oder Live- Filesystem	196
6.4	Hinzufügen von Persistenz auf einem USB-Stick	197
6.4.1	Erstellen einer unverschlüsselten Persistenz auf einem USB-Stick	198
6.4.2	Erstellen einer verschlüsselten Persistenz auf einem USB-Stick	199
6.4.3	Verwenden von mehreren Persistenzspeichern	201
6.5	»Automatisierte« Installation	202
6.5.1	Antworten auf Installationsabfragen vorbereiten	202
6.5.2	Erstellen der Voreinstellungsdatei	204
6.6	Zusammenfassung	205
6.6.1	Kali-Pakete ändern	205
6.6.2	Linux-Kernel neu kompilieren	206
6.6.3	Benutzerdefinierte ISO-Images erstellen	207
7	Ablauf eines Penetrationstests	209
7.1	Informationen sammeln	213
7.1.1	Was nun?	213
7.1.2	Kali-Tools zur Informationsbeschaffung	215
7.1.3	Informationen nach angreifbaren Zielen durchsuchen	215

7.2	Scannen	216
7.2.1	Pings	219
7.2.2	Portscan.	221
7.2.3	Nmap Script Engine – Transformationen eines Tools	229
7.2.4	Schwachstellen-Scan	232
7.3	Eindringen über das lokale Netzwerk	233
7.3.1	Zugriff auf Remotedienste.	234
7.3.2	Übernahme von Systemen	235
7.3.3	Passwörter hacken	238
7.3.4	Abrissbirnen-Technik – Passwörter zurücksetzen	243
7.3.5	Netzwerkverkehr ausspähen	244
7.4	Webgestütztes Eindringen	246
7.4.1	Schwachstellen in Webapplikationen finden.	249
7.4.2	Webseite analysieren	249
7.4.3	Informationen abfangen	249
7.4.4	Auf Schwachstellen scannen.	250
7.5	Nachbearbeitung und Erhaltung des Zugriffs.	250
7.6	Abschluss eines Penetrationstests	252
7.7	Zusammenfassung	253

Teil III Tools in Kali Linux 255

8	Tools zur Informationsbeschaffung und Schwachstellenanalyse ...	257
8.1	Tools zur Informationssammlung.	257
8.1.1	Nmap – Das Schweizer Taschenmesser für Portscanning.	257
8.1.2	TheHarvester – E-Mail-Adressen aufspüren und ausnutzen	262
8.1.3	Dig – DNS-Informationen abrufen.	264
8.1.4	Fierce – falls der Zonentransfer nicht möglich ist.	264
8.1.5	MetaGooFil – Metadaten extrahieren	265
8.1.6	HTTrack – Webseite als Offline-Kopie	267
8.1.7	Maltego – gesammelte Daten in Beziehung setzen.	269
8.1.8	Legion – Automation in der Informationsbeschaffung.	271
8.2	Schwachstellenanalyse-Tools	273
8.2.1	OpenVAS – Sicherheitslücken aufdecken	273
8.2.2	Nikto – Aufspüren von Schwachstellen auf Webservern ...	277
8.2.3	Siege – Performance Test von Webseiten	278

8.3	Sniffing und Spoofing	280
8.3.1	Dsniff – Sammlung von Werkzeugen zum Ausspionieren von Netzwerkdatenverkehr	280
8.3.2	Ettercap – Netzwerkverkehr ausspionieren	281
8.3.3	Wireshark – der Hai im Datenmeer	284
9	Tools für Attacken	287
9.1	Wireless-Attacken	287
9.1.1	aircrack-ng	287
9.1.2	wifiphisher	291
9.1.3	Kismet	293
9.2	Webseiten-Penetration-Testing	295
9.2.1	WebScarab	295
9.2.2	Skipfish	300
9.2.3	Zed Attack Proxy	301
9.3	Exploitation-Tools	304
9.3.1	Metasploit	304
9.3.2	Armitage	312
9.3.3	Social Engineer Toolkit (SET)	313
9.3.4	Searchsploit	316
9.4	Passwort-Angriffe	318
9.4.1	Medusa	319
9.4.2	Hydra	321
9.4.3	John the Ripper	322
9.4.4	Samdump2	326
9.4.5	chntpw	327
10	Forensik-Tools	331
10.1	Dcfldd – Abbild für forensische Untersuchung erstellen	331
10.2	Autopsy	333
10.3	Binwalk	336
10.4	chkrootkit	338
10.5	Bulk_extractor	338
10.6	Foremost	339
10.7	Galleta	340
10.8	Hashdeep	340
10.9	Volafox	342
10.10	Volatility	343

11	Tools für Reports	345
11.1	Cutycapt	345
11.2	Faraday-IDE	347
11.3	Pipal	350
11.4	RecordMyDesktop	351
A	Terminologie und Glossar	353
B	Übersicht Kali-Meta-Pakete	357
B.1	System-Pakete	358
B.2	Tools	360
B.3	Menü	368
C	Checkliste: Penetrationstest	381
C.1	Scope	381
C.2	Expertise	383
C.3	Lösung	383
D	Installation von Xfce und Undercover-Modus	385
	Stichwortverzeichnis	389

Einleitung

Es ist noch nicht lange her, dass Hacking eher ein Tabu war, und es gab auch keine Schulungen dazu. Aber inzwischen hat sich die Erkenntnis breitgemacht, dass auch ein offensiver Ansatz einen Mehrwert für die IT-Sicherheit liefert. Diese neue Herangehensweise wird von vielen Organisationen aller Größen und Branchen begrüßt: Staatliche Stellen machen inzwischen Ernst mit offensiver Sicherheit, Regierungen geben auch offiziell zu, dass sie daran arbeiten.

Für das Sicherheitskonzept einer Organisation spielen vor allem Penetrationstests eine wichtige Rolle. Richtlinien, Risikobewertungen, Notfallpläne und die Wiederherstellung nach Katastrophen sind zu unverzichtbaren Maßnahmen zum Erhalt der IT-Sicherheit geworden und genauso müssen auch Penetrationstests in die Gesamtplanung für die Sicherheit aufgenommen werden. Mit solchen Tests können Sie erkennen, wie Sie vom Feind wahrgenommen werden. Das kann zu vielen überraschenden Entdeckungen führen und Ihnen kostbare Zeit geben, um Ihre Systeme zu verbessern, bevor es einen echten Angriff gibt.

Warum Kali Linux?

Für das Hacking stehen heutzutage viele gute Werkzeuge zur Verfügung. Viele davon sind nicht einfach nur »da«, sondern laufen aufgrund der langjährigen Entwicklungszeit auch sehr stabil. Noch schwerer wiegt für viele die Tatsache, dass die meisten dieser Tools kostenlos erhältlich sind.



Abb. 1: Kali Linux Homepage

Es ist zwar schön, dass diese Werkzeuge kostenlos verfügbar sind, aber Sie müssen sie erst einmal finden, kompilieren und installieren, bevor auch nur der einfachste Penetrationstest durchgeführt werden kann. Auf den modernen Linux-Betriebssystemen geht das zwar relativ einfach, aber für Neulinge kann es immer noch eine abschreckende Aufgabe sein. Auch für Fortgeschrittene ist es mühsam, alle Tools erst mal zusammenzusuchen und zu installieren.

Die Security-Community ist glücklicherweise eine sehr aktive und freigiebige Gruppe. Mehrere Organisationen haben unermüdlich daran gearbeitet, verschiedene Linux-Distributionen für Hacking und Penetrationstests zu erstellen. Eine Distribution (kurz Distro) ist eine Variante von Linux. Für Hacking und Penetrationstests gibt es Linux-Distros, wie:

- Parrot Security OS
- BlackBox
- BlackArch
- Fedora Security Spin
- Samurai Web Testing Framework
- Pentoo Linux
- DEFT Linux
- Caine
- Network Security Toolkit (NST)
- Kali Linux

Die bekannteste Distro für Penetrationstests ist Kali Linux.

Mit Kali Linux erhalten angehende Sicherheitsexperten, Pentester und IT-Verantwortliche eine umfangreiche Plattform, um digitale Attacken zu planen und durchzuführen.

Warum sollte man das tun wollen?

Einerseits, um sich mit potenziellen Angriffen auf die eigenen Systeme auseinanderzusetzen, und zum Zweiten, um interne und externe Schwachstellen besser zu verstehen.

Sollte es so etwas wie ein »Hacker-Betriebssystem« geben, dann trifft diese Bezeichnung wohl am ehesten auf Kali Linux zu. Diese Linux-Distribution ist standardmäßig schon voller Tools, die Sicherheitsexperten und IT-Verantwortlichen entweder den Schlaf rauben oder ihre Augen glitzern lassen.

Kali Linux enthält eigentlich nichts Exklusives – man kann sich jedes Tool, jede Software und jedes Skript auf jedem beliebigen Linux installieren –, dennoch greifen viele Sicherheitsforscher zu Kali.

Die meisten Programme samt den passenden Einstellungen werden bereits mit der Installation von Kali mitgeliefert. Viele der neuen Tools tauchen auch zuerst in den Kali-Repositories auf – auch wenn diese noch nicht ganz stabil sind. Ein weiterer Grund ist, dass Kali sich sehr gut als isolierte Umgebung betreiben lässt. Sollte doch mal etwas schiefgehen, kann das System rasch neu installiert werden und man kann von vorne anfangen – das ist natürlich um vieles besser, als sich eine Produktivumgebung komplett zu zerschießen.

Hinweis

Bevor Sie den Einsatz von Kali Linux erwägen, sollten Sie sich über eines klar sein: Kali ist nicht für jeden das Richtige! Beachten Sie, dass Kali eine Linux-Distribution ist, die speziell für professionelles Penetration Testing und Security Auditing ausgelegt ist. Daher empfiehlt es sich, diese nur zu verwenden, wenn Sie sie für diesen Zweck nutzen möchten. Es ist von Vorteil, wenn Sie bereits mit Linux vertraut sind, da es Ihnen die Arbeit erleichtert und Sie die in diesem Buch beschriebenen Tools so effizienter einsetzen können.

Vorsicht

Die falsche Anwendung von Security-Tools in Ihrem Netzwerk – vor allem ohne Erlaubnis – kann irreparablen Schaden mit erheblichen Folgen anrichten.

Über dieses Buch

In diesem Buch werden keine Vorkenntnisse vorausgesetzt, aber Sie werden sich einen Gefallen tun, wenn Sie sich selbst mit Linux besser vertraut machen, das wird Ihnen die Arbeit mit diesen Tools erleichtern. Besuchen Sie einen Kurs, lesen Sie ein Buch¹ oder erkunden Sie Linux auf eigene Faust. Für diesen Rat werden Sie mir noch dankbar sein. Wenn Sie sich für Penetrationstests und Hacking interessieren, sind Linux-Kenntnisse auf lange Sicht gesehen unabdingbar.

Ich habe das Buch so aufgebaut, dass Sie es auch verwenden können, wenn Sie noch keine Erfahrungen mit Security-Assessments haben bzw. noch nicht mit Linux gearbeitet haben. Wenn Sie das Buch gelesen haben, sollten Sie als Penetrationstester – auch wenn Sie ein Anfänger sind – Security-Assessments mit Kali Linux erfolgreich durchführen können.

Um den Einstieg in die Welt von Kali Linux und Penetrationstests mit Kali Linux zu erleichtern, habe ich das Buch in drei Teile gegliedert.

¹ Linux – Praxiswissen für Ein- und Umsteiger von Christoph Troche (mitp) wäre ein kompaktes Einsteigerbuch

Im ersten Teil wird die Geschichte von Kali Linux beleuchtet und wie Sie Kali installieren und konfigurieren können, um es Ihren Anforderungen anzupassen. Außerdem finden Sie hier auch eine kurze Einführung in Linux, damit Sie, falls Sie Linux-Anfänger sind, trotzdem keine Probleme mit dem Einstieg in Kali Linux haben.

Anschließend zeige ich Ihnen im zweiten Teil, wie Sie am besten einen Penetrationstest aufbauen und wie Sie dabei die Tools von Kali Linux einsetzen. Bedenken Sie aber, dass der Teil nur eines der Modelle behandelt, die beschreiben, wie man einen Penetrationstest aufbauen kann.

Da Kali Linux sehr viele Tools für Security-Assessments mitliefert, werde ich Ihnen im dritten Teil ein paar Tools, die ich für nützlich halte, kurz vorstellen. Sie erfahren, wie Sie diese Tools einsetzen können, aber ich kann Ihnen nur empfehlen, sich mit allen Tools, die Sie für Ihre Security-Assessments benötigen, noch ausführlicher zu beschäftigen. Gerade in dieser Tätigkeit bestätigt sich der Spruch »Übung macht den Meister«. Je mehr Sie sich mit diesen Tools vertraut machen, desto besser und effektiver können Sie diese auch einsetzen.

Im Anhang finden Sie ein praktisches Glossar, eine Übersicht über die Meta-Pakete von Kali Linux sowie eine Checkliste für Penetrationstests, die Ihnen noch eine zusätzliche Hilfestellung gibt, um das Security-Assessment erfolgreich durchzuführen.

Linux-Grundlagen

Um einen fundierten Einstieg ohne Vorkenntnisse zu ermöglichen, starten wir in diesem Buch ganz am Anfang. Sollten Sie bereits Erfahrungen mit Linux haben, können Sie dieses Kapitel getrost überspringen. Es ist jedoch denjenigen, die über Linux-Erfahrung verfügen, zu empfehlen, zumindest die Installation und Konfiguration von Kali Linux in Kapitel 3 zu überfliegen, da sich Kali hier von so mancher Distribution etwas unterscheidet.

2.1 Was ist Linux und wie funktioniert es?

Neben den bekannteren Betriebssystemen wie Windows oder Mac OS gibt es auch noch Linux. Wie jedes Betriebssystem enthält auch eine Linux-Installation eine ganze Reihe von Tools, wie z.B. Internet Browser, Taschenrechner, Texteditor u.v.m. Bei Windows und Mac OS ist die Zusammenstellung dieser Tools standardisiert – sie kann sich zwar je nach Version ändern, aber in jedem Windows 7 Professional sind immer die gleichen Tools enthalten. Das liegt daran, dass Windows nur von Microsoft herausgegeben wird. Gleiches gilt für Mac OS von Apple.

Bei Linux handelt es sich jedoch um eine freie Software, das heißt, jeder kann sich den Kern von Linux herunterladen und seine eigene Distribution erstellen. Eine Distribution ist eine Software-Zusammenstellung. Aktuell gibt es mehrere Hundert Linux-Distributionen, die von genauso vielen Anbietern zur Verfügung gestellt werden. Dazu gehören firmeneigene Distributionen, die für den Eigenbedarf erstellt wurden, aber auch Hobby-Projekte von Enthusiasten sowie professionelle Distributionen mit teilweise kostenpflichtigem Support.

Man kann Distributionen nach dem jeweiligen Einsatzgebiet einteilen. Es gibt hier Distributionen, die darauf ausgelegt sind, als Firewall zu laufen, andere sollen ein möglichst stabiles Arbeitsumfeld mit langfristigem Support liefern, wieder andere stellen die neuesten Programme zur Verfügung und sind für Entwickler zum Testen ihrer Software interessant, diese laufen nicht so stabil. Kali Linux – die Distribution, um die es in dem Buch eigentlich geht – ist eine Distribution, die mit einer enormen Sammlung an Tools für Sicherheitstest, Datenforensik usw. ausgeliefert wird.

Kali Linux ist also ein System, das mit allem geliefert wird, was man benötigt, um in Computersysteme einzudringen. Das ist ideal zum Testen der eigenen Sicherheit, da man damit ein perfektes System zum Hacken hat.

Linux ist eine Open-Source-Software, das heißt, jeder kann den Quelltext einsehen, aus dem Linux besteht. Der Quelltext ist eine Ansammlung von Befehlen, die dann in ein ausführbares Programm übersetzt werden. Das ermöglicht es jedem, den es interessiert, zu sehen, wie Linux programmiert wird. So können Sicherheitslücken schnell gefunden, bekannt gemacht und wieder geschlossen werden. Linux folgt dem Grundsatz: *Alles ist eine Datei*. So werden Programmkonfigurationen gut leserlich in einer Textdatei verwaltet und in der Regel getrennt vom Programm gespeichert. Damit ist es möglich, Programmeinstellungen sehr einfach zu sichern und auf einen anderen Computer zu übertragen.

Da es sich bei Linux um Open-Source handelt, kann man es völlig legal und kostenlos aus dem Internet herunterladen, verwenden und auch weitergeben. Man hat bei Linux sogar die Wahl, welche grafische Oberfläche man verwenden möchte. Bei Kali Linux hat man die Auswahl zwischen mehreren Oberflächen, z.B.

- KDE
- GNOME3
- Enlightenment
- LXDE
- XFCE

Die beiden ersten sind deutlich ressourcenhungriger. Enlightenment, LXDE und XFCE können auch auf bescheidener Hardware eingesetzt werden. Die Vorteile und was die einzelnen grafischen Oberflächen ausmacht, würde den Umfang dieses Buchs sprengen. Laden Sie einfach das ISO-Image herunter und testen Sie selbst. Bei Kali Linux handelt es sich um eine sogenannte Live-CD, die man auch ohne Installation sofort von der DVD oder dem USB-Stick starten und testen kann.

Windows-Rechner sind weitverbreitet und deshalb schon einmal ein beliebtes Ziel für Angriffe. Man kann auch davon ausgehen, dass viele Systeme unsicher konfiguriert sind, weil häufig mit der voreingestellten Konfiguration und zusätzlich auch mit den Administrationsrechten gearbeitet wird.

Linux ist deshalb standardmäßig schon mal sicherer, da es den Benutzer zwingt, eine sichere Konfiguration zu verwenden, und man auch in der Regel standardmäßig nicht mit Administrationsrechten arbeitet. Dadurch, dass Linux, obwohl es kostenlos erhältlich ist, nicht so verbreitet ist wie Windows, ist außerdem die Zahl der Viren, Würmer, Spyware und Trojaner geringer.

Da es bei Linux auch von der Distribution und der grafischen Oberfläche abhängt, welche Tools installiert sind, wird es schwieriger, gezielte Angriffe auf Exploits

zu starten. Bei Windows dagegen kann man davon ausgehen, dass, wenn eine Schwachstelle in Windows-Explorer entdeckt wird, diese auf allen Windows-Systemen ausgenutzt werden kann.

Es ist zwar aufgrund der Einschränkungen und der geringeren Verbreitung weniger effektiv, Schadsoftware für Linux zu entwickeln, aber es ist grob fahrlässig zu behaupten, dass es für Linux keine Viren, Spyware & Co. gibt. Es gibt nur deutlich weniger und in der Regel richten sie deutlich weniger Schaden an, da es ihnen in den meisten Fällen an den notwendigen Rechten fehlt. Aber man darf nicht vergessen, dass man dennoch nicht vollkommen sicher ist.

Als Windows-Anwender kennen Sie sicher Systemabstürze und Bluescreens. Bei Linux – abhängig von der verwendeten Distribution – kommen sie deutlich weniger oft vor, aber ausschließen kann man diese nie gänzlich. Setzt man die neuesten Programmversionen ein, wie z.B. Fedora-Linux, hat man häufig noch mit solchen Kinderkrankheiten zu kämpfen. Verwendet man jedoch Distributionen wie CentOS oder Debian, die vor allem auf Stabilität Wert legen, muss man sich mit einer geringeren Auswahl an Software in den Repositories begnügen, aber man kann sich dafür darauf verlassen, dass diese ausführlich getestet wurden und sehr stabil laufen.

Die Auflistung von Vor- und Nachteilen ist in der Regel sehr subjektiv und es sollte jeder für sich selbst entscheiden, was ihm besser gefällt.

Der Begriff »Linux« wird häufig verwendet, um sich auf das gesamte Betriebssystem zu beziehen, aber Linux ist der Begriff des Betriebssystem-Kernels, der vom Bootloader gestartet wird, und der wiederum wird vom BIOS/UEFI gestartet. Den Kern kann man mit einem Dirigenten in einem Orchester vergleichen – er sorgt für die Koordination zwischen Hard- und Software. Diese Rolle umfasst die Verwaltung von Hardware, Prozessen, Benutzern, Berechtigungen und das Dateisystem. Der Kernel bietet eine gemeinsame Basis für alle anderen Programme und läuft im sogenannten Kernel Space¹.

2.1.1 Hardwaresteuerung

Der Kernel steuert in erster Linie die Hardwarekomponenten des Computers. Er erkennt und konfiguriert diese, wenn der Computer eingeschaltet wird oder ein Gerät (z.B. USB-Stick) hinzugefügt oder entfernt wird. Er bietet auch für übergeordnete Software eine vereinfachte API an, sodass Anwendungen Geräte nutzen können, ohne zu wissen, auf welchem Steckplatz das Gerät angeschlossen ist. Die

1 Bei modernen Betriebssystemen wird der virtuelle Speicher in Kernel-Space und User-Space geteilt. Die Trennung dient zum Speicher- und Hardwareschutz vor böswilliger oder fehlerhafter Software. Kernel-Space ist ausschließlich für die Ausführung vom privilegierten Betriebssystemkern, von Kernel-Erweiterungen und der meisten Gerätetreiber reserviert. Der User-Space wird für Anwendungssoftware und einige Treiber verwendet.

Schnittstelle stellt auch eine Abstraktionsschicht bereit. Das ermöglicht zum Beispiel einer Videokonferenzsoftware das Verwenden einer Webcam unabhängig von Hersteller und Modell. Die Software kann die Video-für-Linux(V4L)-Schnittstelle verwenden und der Kernel übersetzt Funktionsaufrufe der Schnittstelle in tatsächliche Hardware-Befehle, die von der jeweiligen Webcam benötigt werden.

Der Kernel exportiert Daten über erkannte Hardware über die virtuellen Dateisysteme `/proc/` und `/sys/`. Anwendungen greifen häufig auf Geräte über Dateien zu, die in `/dev/` erstellt wurden.

Bestimmte Dateien sind Laufwerke (beispielsweise `/dev/sda`), Partitionen (`dev/sda1`), Mäuse (`/dev/input/mouse0`), Tastaturen (`/dev/input/event0`), Soundkarten (`/dev/snd/*`), serielle Anschlüsse (`/dev/ttyS*`) und andere Komponenten.

Es gibt zwei Arten von Gerädateien: Block und Zeichen. Erstere haben Merkmale eines Blocks von Daten: Sie haben eine begrenzte Größe und Sie können an jeder Stelle eines Blocks auf Bytes zugreifen. Letztere benehmen sich wie ein Fluss von Zeichen. Sie können Zeichen lesen und schreiben, aber man kann nicht nach einer bestimmten Position suchen und beliebige Bytes ändern. Um den Typ einer bestimmten Gerädatei herauszufinden, überprüft man den ersten Buchstaben in der Ausgabe von `ls -l`. Entweder `b` für Blockgeräte oder `c` für Zeichengeräte.

```
root@ictekali:/dev# ls -l /dev/sda /dev/input/mouse0
crw-rw---- 1 root input 13, 32 Mai  5 14:01 /dev/input/mouse0
brw-rw---- 1 root disk  8,  0 Mai  5 14:01 /dev/sda
root@ictekali:/dev#
```

Abb. 2.1: Übersicht der Geräte (Maus und Festplatte), Block oder Zeichengerät

Wie erwartet, verwenden Plattenlaufwerke und Partitionen Blockgeräte, während Maus, Tastatur und serielle Ports Zeichengeräte verwenden. In beiden Fällen enthält die API spezifische Gerätebefehle, die über den `Ioctl`-Systemaufruf aufgerufen werden können.

2.1.2 Vereinheitlichtes Dateisystem

Dateisysteme sind ein wichtiger Aspekt des Kernels. Unix-ähnliche Systeme fassen alle Datenspeicher in einem zusammen. Es gibt also eine einzige Hierarchie, die Benutzer und Anwendungen den Zugriff auf Daten ermöglicht, wenn sie ihren Pfad in dieser Hierarchie kennen.

Der Startpunkt dieses hierarchischen Baums wird als Wurzel (`root`) bezeichnet und durch das Zeichen `»/«` dargestellt. Dieses Verzeichnis kann benannte Unterverzeichnisse enthalten. Zum Beispiel wird das Home-Verzeichnis von `/` aufgerufen: `/home/`. Dieses Unterverzeichnis kann wiederum andere Unterverzeichnisse enthalten usw.

Jedes Verzeichnis kann auch Dateien enthalten, in denen die Daten gespeichert werden. So bezieht sich `/home/user/Desktop/hello.txt` auf eine Datei namens `hello.txt`, die im Unterverzeichnis `Desktop` des User-Unterverzeichnisses des Home-Verzeichnisses gespeichert ist, das im Root-Verzeichnis vorhanden ist. Der Kernel übersetzt zwischen diesem Benennungssystem und dem Speicherort auf einer Festplatte.

Im Gegensatz zu anderen Betriebssystemen verfügt Linux nur über eine solche Hierarchie und kann Daten von mehreren Festplatten dort integrieren. Eine dieser Festplatten wird zum Root-Verzeichnis, und die anderen werden in Verzeichnisse in die Hierarchie gemountet (der Linux-Befehl heißt `mount`). Diese anderen Festplatten sind dann unter den Mountpunkten verfügbar. Dies ermöglicht das Speichern des Home-Verzeichnisses der Benutzer (gewöhnlich in `/home/`), das das User-Verzeichnis enthält (zusammen mit den Basisverzeichnissen von anderen Benutzern). Wenn man eine Festplatte in `/home/` anhängt, sind diese Verzeichnisse an ihrem üblichen Speicherort verfügbar und Pfade wie `/home/user/Desktop/hello.txt` funktionieren weiterhin.

Es gibt viele Dateisystemformate, die vielen Arten der physischen Speicherung von Daten auf Disks entsprechen. Die bekanntesten sind `ext3`, `ext3` und `ext4`, andere gibt es auch noch. Zum Beispiel ist VFAT das Dateisystem, das früher von DOS- und Windows-Betriebssystemen verwendet wurde. Die Unterstützung von Linux für VFAT ermöglicht den Zugriff auf Festplatten sowohl unter Kali als auch unter Windows. In jedem Fall ist die Einrichtung eines Dateisystems auf einer Festplatte notwendig, bevor man diese einhängen kann. Der Vorgang wird als »Formatierung« bezeichnet.

Befehle wie `mkfs.ext3` – wobei `mkfs` für MaKe FileSystem steht – behandeln die Formatierung. Diese Befehle erfordern als Parameter eine Gerätedatei, die die zu formatierende Partition darstellt – beispielsweise `/dev/sda1` für die erste Partition auf dem ersten Laufwerk. Der Vorgang ist destruktiv und sollte nur einmal ausgeführt werden, es sei denn, Sie möchten ein Dateisystem löschen und neu starten.

Es gibt auch Netzwerkdateisysteme wie NFS, die keine Daten auf einer lokalen Festplatte speichern. Stattdessen werden Daten über das Netzwerk an einen Server übertragen, der diese speichert und bei Bedarf abrufen. Dank der Abstraktion des Dateisystems muss man sich keine Gedanken machen, wie diese Festplatte angeschlossen ist, da die Dateien auf ihre gewohnte hierarchische Weise zugänglich bleiben.

2.1.3 Prozesse verwalten

Ein Prozess ist eine laufende Instanz eines Programms, für das Speicherplatz zum Speichern des Programms selbst und seiner Betriebsdaten zur Verfügung gestellt wird. Der Kernel ist für das Erstellen und Verfolgen von Prozessen verantwortlich. Wenn ein Programm ausgeführt wird, stellt der Kernel zunächst etwas

Speicherplatz zur Verfügung, lädt den ausführbaren Code aus dem Dateisystem und startet den Code. Der Kernel speichert Informationen über diesen Prozess, von denen die auffälligste eine Identifikationsnummer ist, die als Prozesskennung (PID) bezeichnet wird.

Wie die meisten modernen Betriebssysteme sind auch Betriebssysteme mit Unix-ähnlichen Kernen, einschließlich Linux, Multitasking-fähig. Anders ausgedrückt: Sie erlauben dem System, viele Prozesse gleichzeitig auszuführen. Es gibt eigentlich immer nur einen laufenden Prozess, aber der Kernel teilt die CPU-Zeit in kleine Scheiben auf und führt jeden Prozess der Reihe nach durch. Da diese Zeitscheiben sehr kurz sind (im Millisekundenbereich), erzeugen sie das Erscheinungsbild von parallel laufenden Prozessen, obwohl sie nur während ihres Zeitintervalls aktiv und die restliche Zeit im Leerlauf sind. Die Aufgabe des Kernels ist es, seine Zeitplanungsmechanismen so anzupassen, dass dieses Erscheinungsbild erhalten bleibt, während die globale Systemleistung maximiert wird. Wenn die Scheiben zu lang sind, erscheint die Anwendung möglicherweise nicht wie gewünscht. Sind sie zu kurz, verliert das System Zeit, da die Aufgaben zu häufig gewechselt werden. Diese Entscheidungen können mit den Prozessprioritäten verfeinert werden, wobei Prozesse mit hoher Priorität über längere Zeiträume und häufiger ausgeführt werden als Prozesse mit niedriger Priorität.

Hinweis

Die oben beschriebene Einschränkung, dass jeweils nur ein Prozess ausgeführt wird, gilt nicht immer: Die wirkliche Einschränkung besteht darin, dass nur ein Prozess pro Prozessorkern ausgeführt werden kann. Multiprozessor-, Multi-Core- oder Hyperthreading-Systeme erlauben, dass mehrere Prozesse parallel laufen. Das gleiche Time-Slicing-System wird jedoch verwendet, um Fälle zu behandeln, in denen mehr aktive Prozesse vorhanden sind als verfügbare Prozessorkerne. Das ist nicht ungewöhnlich: Ein Basissystem, selbst ein größtenteils untätiges, hat fast immer Dutzende laufende Prozesse.

Der Kernel ermöglicht die Ausführung mehrerer unabhängiger Instanzen desselben Programms. Jeder dieser Instanzen ist es jedoch nur erlaubt, auf seine eigenen Zeitscheiben und Speicher zuzugreifen. Ihre Daten bleiben somit unabhängig.

2.1.4 Rechtemanagement

Unix-ähnliche Systeme unterstützen mehrere Benutzer und Gruppen und ermöglichen die Steuerung von Berechtigungen. In der Regel wird ein Prozess über den Benutzer identifiziert, der ihn gestartet hat. Dieser Prozess darf nur Aktionen ausführen, die seinem Besitzer erlaubt sind. Wenn Sie beispielsweise eine Datei öffnen, muss der Kernel die Prozessidentität anhand der Zugriffsberechtigungen prüfen – weitere Informationen hierzu finden Sie in Abschnitt 2.4.4.

2.2 Die Kommandozeile (Command Line)

Mit »Befehlszeile« (Kommandozeile) wird eine textbasierte Schnittstelle bezeichnet, über die Befehle eingegeben, ausgeführt und Ergebnisse angezeigt werden. Sie können ein Terminal (einen Textbildschirm innerhalb der grafischen Oberfläche oder außerhalb einer grafischen Benutzeroberfläche die Textkonsole selbst) und einen Befehlsinterpreter (die Shell) darin ausführen.

2.2.1 Wie komme ich zur Kommandozeile?

Wenn das System ordnungsgemäß funktioniert, können Sie auf die Befehlszeile am einfachsten zugreifen, indem Sie ein Terminal in der grafischen Desktop-Sitzung ausführen.

Auf einem Standard-Kali-Linux-System können Sie das Terminal aus der Favoritenleiste starten. Sie können das Terminal auch über ANWENDUNGEN (in der linken oberen Ecke) starten.

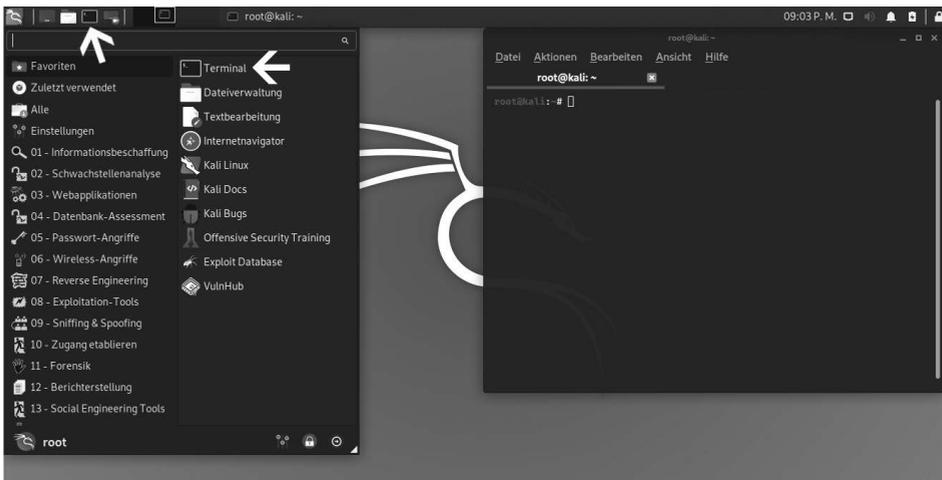


Abb. 2.2: Terminal aufrufen

Für den Fall, dass die grafische Benutzeroberfläche beschädigt ist, können Sie immer noch eine Befehlszeile auf virtuellen Konsolen erhalten (bis zu sechs davon sind über die sechs Tastenkombinationen `Strg` + `Alt` + `F1` bis `Strg` + `Alt` + `F6` aufrufbar, die `Strg`-Taste kann weggelassen werden, wenn Sie sich bereits im Textmodus außerhalb der grafischen Benutzeroberfläche von Xorg² oder Wayland³ befinden). Sie erhalten daraufhin einen sehr einfachen Anmeldebildschirm, in

2 Xorg ist ein Protokoll für die Kommunikation zwischen Ausgabegeräten.

3 Wayland ist wie Xorg ein Protokoll für die Kommunikation zwischen Ausgabegeräten.

dem Sie Ihr Login und Kennwort eingeben, bevor Sie Zugriff auf die Befehlszeile mit der Shell erhalten.

Das Programm, das die Eingabe verarbeitet und die Befehle ausführt, wird als *Shell* (oder Befehlszeileninterpreter) bezeichnet. Die in Kali Linux bereitgestellte Standard-Shell ist Bash (das steht für **B**ourne **A**gain **S**hell). Das abschließende Zeichen \$ oder # zeigt an, dass die Shell auf die Eingabe wartet. Es gibt auch an, ob man die Bash als normaler Benutzer (\$) oder als Superuser (#) nutzt.

2.2.2 Verzeichnisbaum durchsuchen und Dateien verwalten

In diesem Abschnitt erhalten Sie nur einen kurzen Überblick über die behandelten Befehle, von denen alle viele Optionen haben, die hier nicht einzeln beschrieben werden. Weitere Informationen finden Sie in der umfangreichen Dokumentation, die in den jeweiligen Handbuchseiten verfügbar sind. Bei Penetrationstest erhalten Sie nach einem erfolgreichen Exploit meistens Shell-Zugriff auf ein System statt einer grafischen Benutzeroberfläche. Die Kenntnis der Befehlszeile ist für den Erfolg als Sicherheitsprofi also unerlässlich.

Sobald eine Sitzung geöffnet ist, zeigt der Befehl `pwd` (print working directory) den aktuellen Speicherort im Dateisystem an. Das aktuelle Verzeichnis wird mit dem Befehl `cd` (change directory) geändert werden. Wenn das Zielverzeichnis nicht angegeben wird, gelangen Sie zum Home-Verzeichnis. Wenn Sie `cd-` verwenden, kehren Sie zum vorherigen Arbeitsverzeichnis zurück (also die Verwendung vor dem letzten `cd`-Aufruf). Das übergeordnete Verzeichnis heißt immer `..` (zwei Punkte), während das aktuelle Verzeichnis auch als `.` (ein Punkt) bezeichnet wird. Mit dem Befehl `ls` können Sie den Inhalt eines Verzeichnisses auflisten. Wenn Sie keine Parameter angeben, wirkt sich `ls` auf das aktuelle Verzeichnis aus.

```
root@ictekali:~# pwd
/root
root@ictekali:~# cd Desktop
root@ictekali:~/Desktop# pwd
/root/Desktop
root@ictekali:~/Desktop# cd .
root@ictekali:~/Desktop# pwd
/root/Desktop
root@ictekali:~/Desktop# cd ..
root@ictekali:~# pwd
/root
root@ictekali:~# ls
Desktop  Downloads  Pictures    Public      Templates
Documents Music       Programme  sslstrip.log Videos
root@ictekali:~#
```

Abb. 2.3: Befehle `pwd`, `cd` und `ls`

Sie können ein neues Verzeichnis mit dem Befehl `mkdir` *Verzeichnis* erstellen und ein vorhandenes (leeres) Verzeichnis mit dem Befehl `rmdir` *Verzeichnis* entfernen. Mit dem Befehl `mv` können Sie Dateien und Verzeichnisse verschieben und umbenennen. Das Entfernen einer Datei wird mit `rm` *Datei* erreicht, und das Kopieren einer Datei erfolgt mit `cp` *Quelle* *Ziel*.

```

root@ictekali:~# mkdir test
root@ictekali:~# ls
Desktop    Downloads  Pictures   Public     Templates  Videos
Documents Music      Programme  sslstrip.log test
root@ictekali:~# mv test neu
root@ictekali:~# ls
Desktop    Downloads  neu        Programme  sslstrip.log  Videos
Documents Music      Pictures   Public     Templates
root@ictekali:~# rmdir neu
root@ictekali:~# ls
Desktop    Downloads  Pictures   Public     Templates
Documents Music      Programme  sslstrip.log  Videos
root@ictekali:~# █

```

Abb. 2.4: Befehle `mkdir`, `mv`, `rmdir`

Die Shell führt jeden Befehl aus, indem sie das erste Programm des angegebenen Namens in einem Verzeichnis ausführt, das in der Umgebungsvariablen `PATH` aufgeführt ist. Meistens befinden sich diese Programme in `/bin`, `/sbin`, `/user/bin` oder `/usr/sbin`. Der Befehl `ls` befindet sich beispielsweise in `/bin/ls`. Der Befehl `which` gibt die Position einer bestimmten ausführbaren Datei an. Manchmal wird der Befehl direkt von der Shell aus gehandhabt. In diesem Fall wird er als eingebauter Shellbefehl bezeichnet (dazu gehören `cd` und `pwd`). Mit dem Befehl `type` kann man den Typ jedes Befehls abfragen.

```

root@ictekali:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
root@ictekali:~# which ls
/usr/bin/ls
root@ictekali:~# type rm
rm ist /usr/bin/rm
root@ictekali:~# type cd
cd ist eine von der Shell mitgelieferte Funktion.
root@ictekali:~# █

```

Abb. 2.5: Befehle `PATH`, `which`, `type`

Hinweis

Die Verwendung des `echo`-Befehls zeigt einfache Zeichenfolgen auf dem Terminal an. In diesem Fall (siehe Abbildung 2.5) wird der Inhalt einer Umgebungsvariablen angezeigt, da die Shell vor dem Ausführen der Befehlszeile automatisch Variablen mit ihren Werten ersetzt.

Umgebungsvariablen

In Linux ermöglichen die Umgebungsvariablen das Speichern von globalen Einstellungen für die Shell und verschiedene Anwendungen. Diese sind immer kontextbezogen, können aber vererbbar sein. So hat beispielsweise jeder Prozess seine eigene Menge von Umgebungsvariablen. Shells, wie beispielsweise Login-Shells, können Variablen deklarieren, die an andere Programme weitergegeben werden. Diese Variablen können systemweit in `/etc/profile` oder benutzerspezifisch in `~/.profile` definiert werden. Variablen, die nicht für den Befehlszeileninterpreter spezifisch sind, sollten jedoch besser unter `/etc/environment` abgelegt werden, da diese Variablen in alle Benutzer eingefügt werden. Sitzungen können dank des Pluggable Authentication Module (PAM) auch ausgeführt werden, wenn die Shell nicht aktiv ist.

2.3 Das Dateisystem

2.3.1 Dateisystem-Hierarchie-Standard

Wie auch andere Linux-Distributionen ist Kali so organisiert, dass es mit dem Filesystem Hierarchy Standard (FHS) übereinstimmt. So finden sich Benutzer anderer Linux-Distributionen auch leicht mit Kali zurecht. FHS definiert den Zweck eines jeden Verzeichnisses. Die Verzeichnisse der obersten Ebene werden wie folgt beschrieben:

- `/bin/`: Standardprogramme
- `/boot/`: Kali-Linux-Kernel und andere Dateien, die für die frühe Bootphase benötigt werden
- `/dev/`: Geräte-Dateien
- `/home/`: persönliche Dateien des Benutzers
- `/lib/`: Bibliothek
- `/media/*`: Einhängepunkt für entfernbare Geräte – CD-ROM, USB-Stick usw.
- `/mnt/`: vorübergehender Einhängepunkt
- `/opt/`: zusätzliche Anwendungen, die von Dritt-Herstellern bereitgestellt werden
- `/root/`: Root-Verzeichnis des Administrators (*root*)
- `/run/`: Laufzeitdaten, die flüchtig sind und nach einem Neustart nicht bestehen bleiben
- `/sbin/`: Systemprogramme
- `/srv/`: Daten, die von Servern auf diesem System verwendet werden
- `/tmp/`: temporäre Dateien

- */usr/*: Applikationen – das Verzeichnis wird in weitere Verzeichnisse geteilt, *bin*, *sbin*, *lib*, und folgt der gleichen Logik wie das Root-Verzeichnis. Des Weiteren enthält das Verzeichnis */usr/share/* Architektur-unabhängige Daten. Das Verzeichnis */usr/local/* wird vom Administrator für die manuelle Installation von Programmen verwendet, ohne dass Dateien überschrieben werden, die vom Paketsystem (dpkg) verwendet werden.
- */var/*: variable Daten, die von Daemon⁴ verarbeitet werden. Das umfasst Protokolldateien, Warteschlangen, Spools und Caches.
- */proc/* und */sys/*: sind spezifische Linux-Kernel (und nicht Teil des FHS). Diese werden vom Kernel für den Export von Daten in den User-Space benötigt.

2.3.2 Das Home-Verzeichnis des Anwenders

Das Home-Verzeichnis eines Benutzers ist nicht standardisiert, aber es gibt einige außergewöhnliche Konventionen. Das Ausgangsverzeichnis eines Benutzers wird mit einer Tilde (>~<) gekennzeichnet. Diese Info ist vor allem deshalb hilfreich, da der Befehlsinterpreter eine Tilde automatisch durch das richtige Verzeichnis ersetzt (das in der Umgebungsvariablen *HOME* gespeichert ist und dessen üblicher Wert */home/user/* ist).

Üblicherweise sind Anwendungskonfigurationsdateien direkt in Ihrem Home-Verzeichnis gespeichert und die Dateinamen beginnen in der Regel mit einem Punkt. Dabei sollten Sie beachten, dass Dateinamen, die mit einem Punkt beginnen, standardmäßig ausgeblendet sind. Um diese versteckten Dateien auch auflisten zu können, müssen Sie die Option *-a* für den Befehl *ls* mitgeben – also *ls -a*.

Es gibt auch einige Programme, die mehrere Konfigurationsdateien in einem Verzeichnis verwenden (z.B. *~/.ssh/*). Andere Programme (z.B. der Browser Firefox) speichern in ihrem Verzeichnis auch einen Cache mit heruntergeladenen Daten. Das heißt, dass diese Verzeichnisse auch viel Speicherplatz verbrauchen können.

Die Konfigurationsdateien, die direkt im Home-Verzeichnis des Benutzers liegen, bezeichnet man häufig als »Dotfiles«. Diese Konvention ist schon so lange verbreitet, dass diese Verzeichnisse überfüllt sein können. Es gibt aber glücklicherweise auch gemeinsame Anstrengungen unter dem Dach der FreeDesktop.org, aus der die XDG Base Directory Specification hervorgegangen ist, eine Konvention festzusetzen, die darauf abzielt, diese Dateien und Verzeichnis zu bereinigen. In dieser Spezifikation wurde vereinbart, dass Konfigurationsdateien unter *~/.config*, Cache-Dateien unter *~/.cache* und Anwendungsdateien unter *~/.local* (oder deren Unterverzeichnissen) gespeichert werden sollen. Glücklicherweise wird diese Konvention immer häufiger bereits berücksichtigt.

4 Daemon oder auch Dämon bezeichnet in Linux ein Programm, das im Hintergrund abläuft und bestimmte Dienste zur Verfügung stellt.

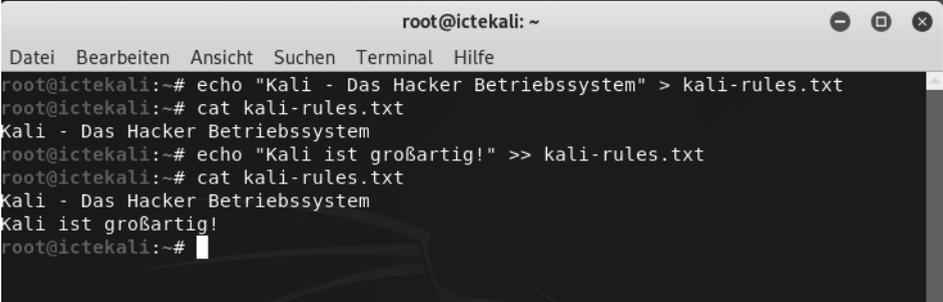
Grafische Desktops verfügen normalerweise über Verknüpfungen, mit denen Inhalte des Verzeichnisses `~/Desktop/` angezeigt werden können (oder auch entsprechende Übersetzungen für Systeme, die nicht auf Englisch konfiguriert sind).

2.4 Hilfreiche Befehle

2.4.1 Anzeigen und Ändern von Text-Dateien

Der Befehl `cat file` liest die Datei und zeigt den Inhalt am Terminal an. Sollte die Datei zu groß sein, um auf einen Bildschirm zu passen, kann man wie auf einem Pager Seite für Seite durchscrollen.

Der Editor-Befehl (abhängig vom Editor) startet einen Texteditor (wie Vi oder Nano) und ermöglicht das Erstellen, Ändern und Lesen von Textdateien. Einfache Dateien können manchmal dank Redirection⁵ mit Befehl `>Datei` erstellt werden. Es wird eine Datei mit dem Namen `file` erzeugt, die die Ausgabe des Befehls als Inhalt hat. Mit Befehl `>>Datei` funktioniert es ähnlich, nur die Ausgabe des Befehls wird an die Datei angehängt, statt diese zu überschreiben.



```
root@ictekali: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
root@ictekali:~# echo "Kali - Das Hacker Betriebssystem" > kali-rules.txt
root@ictekali:~# cat kali-rules.txt
Kali - Das Hacker Betriebssystem
root@ictekali:~# echo "Kali ist großartig!" >> kali-rules.txt
root@ictekali:~# cat kali-rules.txt
Kali - Das Hacker Betriebssystem
Kali ist großartig!
root@ictekali:~#
```

Abb. 2.6: Ausgabe von Befehlen in Datei umleiten

2.4.2 Suche nach Dateien und innerhalb von Dateien

Mit dem Befehl `find Verzeichnis Kriterien` sucht man nach Dateien der Hierarchie des *Verzeichnisses* nach den angegebenen *Kriterien*. Das häufigste verwendete Kriterium ist `-name Dateiname`, mit dem Sie nach einem Dateinamen suchen können. Sie können auch die gebräuchlichen Wildcards, wie `»*«` im Dateinamen für die Suche verwenden.

5 Bei Redirection wird die Ausgabe, die ein Befehl üblicherweise am Bildschirm ausgibt, stattdessen in eine Datei geschrieben.

```
root@ictekali:~# find /etc -name hosts
/etc/avahi/hosts
/etc/hosts
root@ictekali:~# find /etc -name "hosts*"
/etc/hosts.allow
/etc/avahi/hosts
/etc/hosts.deny
/etc/hosts
root@ictekali:~#
```

Abb. 2.7: Der Befehl `find` mit dem Suchkriterium `-name` in unterschiedlichen Varianten

Mit `grep` *Ausdruck Datei* durchsuchen Sie den Inhalt einer Datei und extrahieren Zeilen, die mit dem regulären Ausdruck übereinstimmen. Wollen Sie eine rekursive Suche nach Dateien in allen Verzeichnissen durchführen, verwenden Sie die Option `-r`. Auf diese Weise können Sie nach einer Datei suchen, wenn Sie nur einen Teil des Inhalts kennen.

2.4.3 Prozesse verwalten

Um alle gerade ausgeführten Prozesse aufzulisten, verwenden Sie den Befehl `ps aux`. Durch das Anzeigen der PID (Prozess-ID) können Sie diese Prozesse identifizieren. Kennen Sie die PID eines Prozesses, so können Sie mit dem Befehl `kill -signal PID` ein Signal an den Prozess senden, um diesen sofort zu beenden – vorausgesetzt Sie sind der Eigentümer des Prozesses. Es gibt mehrere Signale. Am häufigsten werden `TERM` – eine Aufforderung, den Prozess ordnungsgemäß zu beenden – und `KILL` – um den Prozess sofort zu beenden (killen) – verwendet.

Der Befehlsinterpreter kann Programme auch im Hintergrund ausführen, wenn dem Befehl ein »&« folgt. Mit dem kaufmännischen »Und« können Sie die Kontrolle über die Shell sofort wieder übernehmen, auch wenn der Befehl noch ausgeführt wird – als Hintergrundprozess wird dieser ausgeblendet.

Mit dem Befehl `jobs` listen Sie alle im Hintergrund laufenden Prozesse auf. Wenn Sie `fg %job-number` eingeben, bringt der Befehl den Job in den Vordergrund. Wird ein Befehl im Vordergrund ausgeführt (entweder weil er normal gestartet wurde oder mit `fg` wieder in den Vordergrund gebracht wurde), halten Sie mit der Tastenkombination `[Strg]+[Z]` den Vorgang an und übernehmen wieder die Steuerung des Terminals. Der Prozess kann dann im Hintergrund neu gestartet werden mit `bg% job-number`.

2.4.4 Rechte verwalten

Bei Linux handelt es sich um ein Multi-User-System, deshalb ist es auch erforderlich, ein Berechtigungssystem zur Steuerung einer Reihe von autorisierten Vorgängen für Dateien und Verzeichnisse bereitzustellen. Das Berechtigungssystem muss dabei alle Systemressourcen und Geräte umfassen – auf einem Unix-System

wird jedes Gerät durch eine Datei oder ein Verzeichnis dargestellt. Dieses Prinzip haben alle Unix-basierenden Systeme gemeinsam.

Eine jede Datei und ein jedes Verzeichnis verfügt dabei über bestimmte Berechtigung für drei Benutzerkategorien:

- **Besitzer (Owner):** wird durch ein `u` wie in User gekennzeichnet
- **Besitzerguppe (owner group):** wird durch ein `g` wie in group gekennzeichnet
- **Die Anderen (others):** wird durch ein `o` gekennzeichnet

Diese drei Typen von Rechten können kombiniert werden:

- **Lesen (reading):** durch ein `r` gekennzeichnet
- **Schreiben (writing):** durch ein `w` gekennzeichnet
- **Ausführen (executing):** durch ein `x`, wie in execute, gekennzeichnet

Bei einer Datei sind diese Rechte einfach zu verstehen: Der Lesezugriff ermöglicht Ihnen das Lesen des Inhalts – inklusive Kopieren –, mit dem Schreibzugriff können Sie die Datei verändern und mit dem Ausführen-Zugriff kann ein Programm auch ausgeführt werden – das funktioniert aber nur, wenn es sich um ein Programm handelt.

Für eine ausführbare Datei sind zwei bestimmte Rechte relevant: `setuid` und `setgid` (durch `s` gekennzeichnet). Zu beachten gilt, dass man häufig von Bit spricht, da jeder dieser booleschen Werte durch eine 0 oder eine 1 dargestellt werden kann. Diese beiden Rechte ermöglichen jedem Benutzer die Ausführung des Programms mit den Rechten des Eigentümers bzw. der Gruppe. Dieser Mechanismus gewährt Zugriff auf Funktionen, für die höhere Berechtigungen als normalerweise erforderlich sind. Da `setuid` Root-Programme systematisch unter der Superuser-Identität ausführt, ist es sehr wichtig, dass das Programm sicher und zuverlässig ist. Jeder Benutzer, der es schafft, ein `setuid`-Programm zu unterwandern, um einen Befehl seiner Wahl aufzurufen, könnte sich als Root-Benutzer ausgeben und alle Rechte auf dem System besitzen. Penetrationstester suchen regelmäßig nach diesen Datentypen, wenn sie Zugriff auf ein System erhalten, um die Rechte zu erweitern.

Ein Verzeichnis wird nicht wie eine Datei behandelt. Lesezugriff gibt das Recht, das Inhaltsverzeichnis (Dateien und Verzeichnisse) zu sehen; der Schreibzugriff ermöglicht das Erstellen oder Löschen von Dateien und Verzeichnissen; das Ausführen-Recht ermöglicht das Durchsuchen des Verzeichnisses und auf dessen Inhalt zuzugreifen (z.B. mit dem Befehl `cd`). Die Möglichkeit, in ein Verzeichnis zu wechseln, ohne Lesezugriff zu besitzen, erlaubt es dem Benutzer, namentlich auf bekannte Einträge darin zuzugreifen. Er kann diese aber nicht finden, ohne deren genauen Namen und Pfad zu kennen.

Sicherheitshinweis

Das setgid-Bit gilt auch für Verzeichnisse. Jedem neu erstellten Element in einem solchen Verzeichnis wird automatisch die Eigentümergruppe des übergeordneten Verzeichnisses zugewiesen, anstatt die Hauptgruppe des Erstellers zu erben. Deshalb müssen Sie die Hauptgruppe nicht (mit dem Befehl `newgrp`) ändern, wenn Sie in einem Verzeichnisbaum arbeiten, der von mehreren Benutzern mit der gleichen dedizierten Gruppe gemeinsam genutzt wird. Das Sticky-Bit – durch `t` symbolisiert – ist eine Berechtigung, die nur in Verzeichnissen nützlich ist. Es wird insbesondere für temporäre Verzeichnisse verwendet, in denen jeder Schreibzugriff hat – z.B. `/tmp/`: Es schränkt das Löschen von Dateien ein, sodass nur deren Eigentümer oder der Eigentümer des übergeordneten Verzeichnisses diese löschen kann. Ansonsten könnte jeder Dateien anderer Benutzer in `/tmp/` löschen.

Drei Befehle steuern die mit einer Datei bzw. einem Verzeichnis verknüpften Berechtigungen:

- `chown User Datei`: ändert den Besitzer einer Datei/eines Verzeichnisses
- `chgrp Gruppe Datei`: ändert die Eigentümer-Gruppe
- `chmod Rechte Datei`: ändert die Zugriffsrechte

Hinweis

Häufig möchten Sie die Gruppe einer Datei gleichzeitig mit dem Eigentümerwechsel ändern. Der Befehl dazu hat eine spezielle Syntax: `chown User:Gruppe Datei`.

Sie haben zwei Möglichkeiten, Rechte darzustellen. Am einfachsten zu verstehen und zu merken ist wahrscheinlich die symbolische Darstellung. Es handelt sich dabei um die bereits genannten Buchstabensymbole. Sie können die Rechte für jede Benutzerkategorie (`u/g/o`) definieren, indem Sie diese explizit festlegen (=) oder durch Hinzufügen (+) bzw. Wegnehmen (-). Das würde bei der Formel `u=rwx,g+rw,o-r` Folgendes ergeben:

- Eigentümer (owner) – `u` – erhält Lese-, Schreib- und Ausführrechte.
- Eigentümergruppe (owner group) – `g` – werden Lese- und Schreibrechte hinzugefügt.
- Rest (Andere/others) – `o` – alle anderen Benutzer, die nicht in die ersten beiden Gruppen fallen, verlieren ihre Leserechte.

Rechte, die durch Hinzufügen oder Entfernen nicht geändert werden, bleiben unverändert. Der Buchstabe `a` deckt dabei alle drei Benutzerkategorien ab, sodass

`a=rx` allen drei Kategorien die gleichen Rechte – Lesen und Ausführen, aber nicht Schreiben – einräumt.

Die (oktale) numerische Darstellung ordnet jedem Recht einen Wert zu: 4 zum Lesen, 2 zum Schreiben und 1 zum Ausführen. Verknüpft man jede Kombination von Rechten mit der Summe der drei Zahlen und jeder Benutzerkategorie, wird in der üblichen Reihenfolge (Eigentümer, Gruppe, Andere) ein Wert zugewiesen.

Wird beispielsweise der Befehl `chmod 754 Datei` ausgeführt, so werden folgende Rechte festgelegt:

- Lesen, Schreiben und Ausführen für den Eigentümer (da $7 = 4 + 2 + 1$)
- Lesen und Ausführen für die Gruppe (da $5 = 4 + 1$)
- Schreibgeschützt für andere ($4 =$ nur Leserechte)

Die 0 bedeutet keine Rechte, somit würde `chmod 600 Datei` nur Lese- und Schreibrechte für den Besitzer und keine Rechte für alle anderen bedeuten. Die häufigste Kombination ist 755 für ausführbare Dateien und Verzeichnisse und 644 für Datendateien.

Um Sonderrechte zu vergeben, können Sie dieser Zahl nach dem gleichen Prinzip eine vierte Ziffer voranstellen, wobei die Bits `setuid`, `setgid` und `sticky` jeweils 4, 2 und 1 sind. Der Befehl `chmod 4754` ordnet das `stuid`-Bit den zuvor beschriebenen Rechten hinzu.

Beachten Sie dabei, dass bei der Verwendung der Oktalnotation nur alle Rechte auf einmal für eine Datei festgelegt werden können. Sie können diese nicht dazu verwenden, ein neues Recht hinzuzufügen, z.B. einen Lesezugriff für den Gruppeneigentümer, da Sie die vorhandenen Rechte berücksichtigen und einen neuen entsprechenden numerischen Wert berechnen müssen. Die oktale Darstellung wird auch mit dem Befehl `umask` verwendet, mit dem die Berechtigungen für neu erstellte Dateien eingeschränkt werden. Wenn eine Anwendung eine Datei erstellt, weist sie indikative Berechtigungen zu, in dem Wissen, dass das System die mit `umask` definierten Rechte automatisch entfernt. Gibt man `umask` in der Shell ein, sieht man eine Maske wie `0022`. Das ist eine einfache oktale Darstellung der Rechte, die systematisch entfernt werden müssen (in diesem Fall die Schreibrechte für die Gruppe und andere Benutzer).

Wenn Sie einen neuen Oktalwert eingeben, ändert der Befehl `umask` die Maske. In einer Shell-Initialisierungsdatei (z.B. `~/.bash_profile`) wird die Standardmaske für die Arbeitssitzung geändert.

Tipp

Manchmal müssen die Rechte für einen gemeinsamen Verzeichnisbaum geändert werden. Alle oben angeführten Befehle besitzen die Option `-R`, um in Unter-

verzeichnissen rekursiv zu arbeiten. Die Unterscheidung zwischen Verzeichnissen und Dateien verursacht manchmal Probleme mit rekursiven Operationen. Deshalb wurde der Buchstabe »X« in die symbolische Darstellung von Rechten eingefügt. Er stellt ein Ausführungsrecht dar, das nur für Verzeichnisse gilt – und nicht für Dateien, denen dieses Recht fehlt. Daher fügt `chmod -R a+X Verzeichnis` nur Ausführungsrechte für alle Benutzerkategorien (a) für alle Unterverzeichnisse und Dateien hinzu, für die mindestens eine Benutzerkategorie bereits Ausführungsrechte besitzt (auch wenn es nur ihr alleiniger Eigentümer ist).

2.4.5 Systeminformationen und Logs aufrufen

Der Befehl `free` gibt Informationen zum Arbeitsspeicher (Memory) aus, `disk free` (`df`) berichtet den verfügbaren Speicherplatz von jeder dem System angehängten Festplatte. Die Option `-h` (für Menschen lesbar) konvertiert die Größe in eine besser lesbare Einheit – üblicherweise Mega- oder Gigabyte. In ähnlicher Weise unterstützt der Befehl `free` auch die Optionen `-m` und `-g` und zeigt seine Daten entweder in Mega- oder in Gigabyte an.

```
root@ictekali:~# free
              total        used         free       shared  buff/cache   available
Mem:           2043104      817808       588760        18704     636536     1054948
Swap:          2095100           0       2095100

root@ictekali:~# df
Dateisystem  1K-Blöcke  Benutzt  Verfügbar  Verw%  Eingehängt auf
udev          989872      0     989872    0%  /dev
tmpfs         204312     6436    197876    4%  /run
/dev/sdal    79980100 17821204 58053120   24%  /
tmpfs        1021552      0    1021552    0%  /dev/shm
tmpfs         5120         0       5120    0%  /run/lock
tmpfs        1021552      0    1021552    0%  /sys/fs/cgroup
tmpfs         204308     16     204292    1%  /run/user/135
tmpfs         204308     28     204280    1%  /run/user/0
root@ictekali:~#
```

Abb. 2.8: Die Befehle `free` und `disk free` (`df`)

Der Befehl `id` zeigt die Identität des Users an, der die Sitzung ausführt, sowie die Liste der Gruppen, zu denen er gehört. Da der Zugriff auf einige Dateien und Geräte möglicherweise auf Gruppenmitglieder beschränkt ist, kann eine Überprüfung der verfügbaren Gruppenmitgliedschaften hilfreich sein.

Der Befehl `uname -a` gibt eine einzelne Zeile zurück, in der der Name des Kernels (Linux), der Hostname, das Kernel-Release, die Kernel-Version, der Maschinentyp (ein Architekturstring, wie `x86_64`) und der Name des Betriebssystems (GNU/Linux) stehen. Die Ausgabe dieses Befehls sollte normalerweise in Fehlerberichten

enthalten sein, da sie den verwendeten Kernel und die verwendete Hardwareplattform, auf der sie ausgeführt werden, klar definiert.

Diese Befehle liefern zwar Laufzeitinformationen, aber um zu verstehen, was auf dem Computer passiert, sollten Sie die Protokolle zur Hilfe nehmen. Vor allem der Kernel sendet Nachrichten, die in einen Ringbuffer gespeichert werden, wenn etwas Interessantes passiert (z.B. Einstecken eines neuen USB-Geräts, eine fehlerhafte Festplattenoperation oder eine erste Hardwareerkennung beim Booten). Sie können die Kernel-Protokolle mit dem Befehl `dmesg` abrufen.

Das Journal von `Systemd`⁶ speichert auch mehrere Protokolle (stdout/stderr-Ausgabe von Daemons, Syslog-Nachrichten, Kernelprotokollen) und macht es einfach, sie mit `journalctl` abzufragen. Ohne Argumente werden alle verfügbaren Protokolle in chronologischer Reihenfolge gesichert. Mit der Option `-r` wird die Reihenfolge umgekehrt, sodass neuere Nachrichten zuerst angezeigt werden. Mit der Option `-f` werden fortlaufend neue Protokolleinträge gespeichert, indem sie an die Datenbank angehängt werden. Die Option `-u` kann die Nachrichten auf die von einer bestimmten Systemeinheit ausgegebenen Nachrichten beschränken (z.B. `journalctl -u ssh.service`).

2.4.6 Hardware erkennen

Der Kernel speichert viele Details über erkannte Hardware in den virtuellen Dateisystemen `/proc/` und `/sys/`. Mehrere Tools fassen diese Details zusammen. Dazu gehören

- `Ispci` (im Paket `pciutils`), das PCI-Geräte auflistet
- `Isubsb` (im Paket `usbutils`), das USB-Geräte auflistet
- `Ispcmcia` (im Paket `pcmciautils`), das PCMCIA-Karten auflistet

Diese Tools sind nützlich, um das genaue Modell eines Geräts zu identifizieren. Diese Identifizierung ermöglicht präzisere Suchvorgänge im Internet, die zu relevanteren Ergebnissen führt. Die Tools `pciutils` und `usbutils` werden bereits im Kali-Basissystem mitgeliefert, `pcmciautils` muss jedoch erst installiert werden (`apt install pcmciautils`).

Bei diesen Tools bietet die Option `-v` die Möglichkeit, noch viel detailliertere – aber in der Regel nicht benötigte – Informationen angezeigt zu bekommen. Der Befehl `lsdev` (im Paket `procinfo` – muss erst mit `apt-get install procinfo` installiert werden) listet die von Geräten verwendeten Kommunikationsressourcen auf.

⁶ `Systemd` ist ein Hintergrundprozess, der als Erstes gestartet wird und dient zum Starten, Überwachen und Beenden von weiteren Prozessen.

```

root@ictekali: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
root@ictekali:~# lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: InnoTek Systemberatung GmbH VirtualBox Graphics Adapter
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:05.0 Multimedia audio controller: Intel Corporation 82801AA AC'97 Audio Controller (rev 01)
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:0b.0 USB controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB2 EHCI Controller
00:0d.0 SATA controller: Intel Corporation 82801HM/HEM (ICH8M/ICH8M-E) SATA Controller [AHCI mode] (rev 02)
root@ictekali:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
root@ictekali:~#

```

Abb. 2.9: Beispiel der Informationen, die lspci und lsusb liefern

Das lshw-Tool (muss mit `apt-get install lshw` installiert werden) ist eine Kombination der oben genannten Tools und zeigt eine Beschreibung der gefundenen Hardware auf hierarchische Weise an. Eine vollständige Ausgabe von `lshw` sollte an jedem Bericht über Hardware-Support-Probleme angehängt werden.

2.5 Zusammenfassung

In diesem Kapitel haben Sie einen Kurzüberblick über die Linux-Landschaft bekommen. Das Konzept von Kernel- und Userspace und viele Linux-Shell-Befehle wurden erläutert wie auch die Prozesse und deren Verwaltung sowie das Benutzer- und Gruppensicherheitskonzept erklärt. Außerdem sind das FHS und einige der gebräuchlichsten Verzeichnisse und Dateien unter Kali Linux vorgestellt worden.

- Linux wird oft verwendet, um auf das gesamte Betriebssystem zu verweisen, jedoch handelt es sich bei Linux selbst um den Betriebssystemkern, der vom Bootloader gestartet wird, der selbst vom BIOS bzw. UEFI geladen wird.
- Der User-Space bezeichnet alles, was außerhalb des Kernels passiert. Unter den Programmen, die im User-Space ausgeführt werden, gibt es viele Kerndienstprogramme aus dem GNU-Projekt, die meistens über die Shell ausge-

führt werden (eine textbasierte Oberfläche, über die Befehle eingegeben, ausgeführt und die Ergebnisse angezeigt werden können).

- Zu den allgemeinen Befehlen gehören:
 - `pwd` – Arbeitsverzeichnis drucken
 - `cd` – Verzeichnis ändern
 - `ls` – Datei- und Verzeichnisinhalt auflisten
 - `mkdir` – Verzeichnis erstellen
 - `rmdir` – Verzeichnis entfernen
 - `mv`, `rm` und `cp` – Verschieben, Entfernen und Kopieren von Dateien bzw. Verzeichnissen
 - `cat` – Verketteten oder Anzeigen von Dateien
 - `editor` – startet einen Texteditor
 - `find` – findet eine Datei oder ein Verzeichnis
 - `free` – zeigt den freien Memory-Speicher an
 - `df` – zeigt den freien Speicherplatz der Festplatten an
 - `id` – zeigt die Identität eines Benutzers zusammen mit einer Liste der Gruppen, zu denen er gehört, an
 - `dmesg` – Überprüfung der Kernel-Protokolle
 - `journalctl` – zeigt alle verfügbaren Protokolle an
- Die Hardware auf einem Kali-System kann mit mehreren Befehlen überprüft werden:
 - `lspci` – listet die PCI-Geräte auf
 - `lsusb` – listet die USB-Geräte auf
 - `ls pcmcia` – listet die PCMCIA-Karten auf
- Ein Prozess ist eine laufende Instanz eines Programms, die Speicher benötigt, um sowohl das Programm selbst als auch seine Betriebsdaten zu speichern. Man kann die Prozesse mit folgenden Befehlen verwalten:
 - `ps` – Prozesse anzeigen
 - `kill` – Prozesse beenden
 - `bg` – Prozesse in den Hintergrund verschieben
 - `fg` – Hintergrundprozesse in den Vordergrund verschieben
 - `jobs` – zeigt Hintergrundprozesse an
- Unix-ähnliche Systeme sind Mehrbenutzersysteme. Das heißt, sie unterstützen mehrere Benutzer und Gruppen und ermöglichen die Steuerung von Aktionen basierend auf Berechtigungen. Sie können Datei- und Verzeichnisrechte mit verschiedenen Befehlen verwalten:

- `chmod` – Berechtigungen ändern
- `chown` – Besitzer ändern
- `chgrp` – Gruppe ändern
- Wie auch bei anderen professionellen Linux-Distributionen ist Kali Linux so organisiert, dass es mit dem Filesystem Hierarchy Standard (FHS) konsistent ist, sodass Benutzer, die Erfahrungen mit anderen Linux-Distributionen haben, sich auch in Kali Linux leicht zurechtfinden.

Üblicherweise werden Anwendungskonfigurationsdateien in Ihrem Ausgangsverzeichnis in versteckten Dateien oder Verzeichnissen gespeichert, die mit einem Punkt beginnen.

Nach diesem Kapitel sollten Sie die Grundlagen von Linux kennen und Sie können im nächsten Schritt Kali Linux installieren und starten.

Stichwortverzeichnis

A

- Abstraktionsschicht 38
- Abuse-Meldung 276
- Access Point 287
- ACK-Paket 222, 223
- address resolution protocol 258
- Address Space Layout Randomization 158
- Administrationsrecht 32
- Administrativer Zugang 212
- Administrativer Zugriff 233
- Administratorkonto 329
 - knacken 238
- Administratorpasswort
 - zurücksetzen 328
- Adresse
 - physische 259
- Adware 353
- aircrack-ng 27, 287
- Aktive Informationsbeschaffung 258
- amd64-Plattformen 57
- Analysieren
 - von Kennwörtern 280
- Android 360
- Android-Exploit 317
- Anforderungen
 - behördliche 163
 - branchenspezifische 163
- Angriff 353
 - clientseitiger 173
 - webgestützter 246
- Angriffserkennungssystem 258
- Anmeldeinformationen 282
- Ansatz
 - hybrider 168
- Anti-Exploit-Technologie 158
- Anwenderaktualisierung 173
- Anwendung
 - Standard-Anwendungen 358
- Anwendungs-Assessment 168
- Anwendungsdatei 45
- Anwendungskonfigurationsdatei 45
- Anwendungsverhalten 167
- Apache-Konfigurationsanweisungen 118
- Apache-Prozess 116
- Apache-Standardmodule 116
- Apache-Webserver
 - konfigurieren 115
- Applikations-Assessment 157, 166
- APT 181
- Arbeitsspeicher 86
- ARM-Computer 95
- Armel-Plattformen 57
- Armhf-Plattformen 57
- Armitage 236, 237, 312
- ARM-Plattformen 57
- ARP 258
- ARP-Cache 259
- ARPreplay-Attacke 290
- ARP-Request 290
- arpspoof 281
- ARP-Spoofing 283
- Assessment
 - Arten von 156
 - Installation 155
- Assessment-Plattform 160
- Aufklärung 209, 214, 263
- Aufklärungsphase 210
- Ausbildung 367
- Auslagerungsdatei 86
- Auslagerungspartition 86
- Auswirkungen 162
- Authentifizierter Scan 160
- Authentifizierung
 - Access Point 290
 - Basic 118
- Authentifizierungsebene 247
- Automatisierte Installation 155
- Automatisierter Scan 159
- Automatisierte Tools 161
- Automatisierung 221
- Autopsy 333
 - Analyse 335
- Availability 153

B

Backdoor 353
 Back-End-Seitengenerierungslogik 171
 BackTrack 19
 Banner 231
 Base64-Codierung 300
 Bash 42
 Basissystem-Bibliothek 358
 Bedrohung 155
 Bedrohungsstufe 277
 Befehle
 Übersicht 54
 Befehlsinterpreter 47
 Befehlszeileninterpreter 42
 Befehlszeilenwerkzeuge 92
 Befehlszeile *siehe* Kommandozeile
 Belastungstest 279
 Benchmarking 278
 Benutzerkennwort 70
 Berechtigungssystem 47
 Bereitstellungspunkt 63, 326
 Bericht 379
 erstellen 166, 212
 Berichterstattung 212
 Betriebssystemversion 158
 Bettercap 27
 Bibliothek
 Basissystem-Bibliotheken 358
 BID-Nummer 307
 Bildanalyse 339
 Binärer Hook 196
 Binär-Image 336
 Binärpaket 185
 Bind-Payload 311
 Binwalk 336
 BIOS 37
 Black-Box-Assessment 167
 Bluetooth 363
 Bootfähiges Speichermedium 64
 Bootkey 326
 Bootloader 37, 79, 186
 Bootloader-Konfiguration
 ändern 203
 Boot-Parameter 202, 204
 Bot 353
 Botnet 353
 Breitband
 mobiles 107
 Bridged Sniffing 283
 Broadcast 230, 244
 Brute Force 279
 Brute-Force-Anmeldetool 319
 Brute-Force-Attacke 234, 322, 353

Brute-Force-Methode 318, 320
 BSI 232
 BSSID 289
 Buffer-Overflow 157, 171, 353
 Bugtraq ID Database 307
 Build-Abhängigkeiten installieren 180
 Build-Environment 183
 Build-Essential-Paket 186
 Build-Option 183
 Build-Prozess 185
 Bulk_extractor 338
 Burp Suite 247

C

Caching-Proxy 78
 CANVAS 304
 Capture-Filter 285
 CentOS 37
 chntpw 327
 Chromebook 59
 Chroot Hooks 196
 chrootkit 338
 Chroot-Umgebung 196
 CIA-Triade 153
 Clientseitiger Angriff 173
 Clone Phishing 353
 Closed-Source-Datei 29
 Cloud-Dienstanbieter 169
 Cloud-Installation 24
 Cloud-Service 169
 Codeausführung 232
 Code-Execution-Exploit 170
 Common Vulnerabilities Exposures 307
 Compliance 163
 Compliance-Framework 164
 Compliance-Test 157, 163, 164
 Confidentiality 153
 Connect-Scan 259
 Cookies 340
 CORE Impact 304
 Cracker 353
 Cracks pro Sekunde
 messen 323
 Crawler 268
 Crawling 249
 Cross Site Scripting (XSS) 172, 248, 356
 Cryptcat 251
 Cutycapt 345
 CVE 232
 CVE-Nummer 160, 307
 CVSS-Score 161
 Cyber-Hygiene 137
 Cyberuntersuchung 339

D

Daemon-Daten 73
 Daemons 111
 Data Execution Prevention 158
 Dateisystem 38, 39
 virtuelles 52
 Dateisystemformat 39
 Datenbankserver
 PostgreSQL 113
 Datenbanksicherheit 370
 Datenintegrität 332
 Datenpaket
 suchen 285
 WLAN 288
 Datenstruktur
 wiederherstellen 339
 Datenverkehr 219, 245
 Dcfldd 331
 DDoS 354
 Debconf-Datenbank 102
 Debconf-Fragen 203
 Debconf-Voreinstellungen 202
 DEBEMAIL 182
 DEBFULLNAME 182
 Debian 19, 37
 Debian-Kernel-Handbuch 186
 Debian-Kernel-Paket 186
 Debian-Live-Systemhandbuch 194
 Debian-Packaging 181
 Debian-Paket 186, 191
 Debian-Quellverwaltungs-Datei 177
 Debian-Richtlinien 32
 Debian Unstable 29
 Debugging-Symbol 192
 Debug-Meldung 176
 Dedizierte Gruppe 49
 Dedizierte Schnittstelle 189
 Default Desktop 82
 Default Gateway 108
 Denial of Service 170, 354
 Denial-of-Service-Angriff 245
 Denial-of-Service-Bedingung 170
 Desktop-Anwendungen 166
 Desktop-Sitzung 41
 Desktop-Umgebung 30, 194
 Device-Mapper 85
 dget-Quellpaket 180
 DHCP 109, 226
 DHCP-Einstellungen 156
 Dienst
 aktiver 216
 Dig 215, 264

Digitale Forensik 378
 Digitaler Fingerabdruck 267
 Display-Filter 285
 Distribution 19, 35
 DMZ 258
 DNS 226
 Dns2proxy 147
 DNS-Abfrage 264
 DNS-Server 108, 214
 dnsspoof 281
 Domain Controller 351
 Domänenadministratorkonto 238
 DoS 154, 170, 354
 DoS-Angriff 170
 DoS-Ergebnis 318
 dpkg-Dateien 176
 Drei-Wege-Handshake 222, 225
 Drohne 294
 dsniff 245, 280

E

EDB-ID 160
 Eindringen 166, 212
 netzwerkgestütztes 246
 Eingangsbuffer 135
 Einstellungs-Reiter 298
 Eintrittswahrscheinlichkeit 161
 E-Mail-Adressen
 aufspüren 262
 E-Mail-Passwort 282
 Embedded Device 59
 Encoder 305
 Endgeräte
 mobiles 167
 Engineering
 soziales 375
 Enlightenment 36
 Ermittler
 forensischer 331
 Erstellungszeitpunkt 183
 Ethernet-Netzwerk 281
 Ethischer Hacker 212
 Ettercap 281
 Sniff-Modi 283
 Exploit 157, 233, 251, 306, 354
 Definition 157
 Exploitation 374
 Exploitation-Tools 304
 Exploit-Code 170
 Exploit-Datenbank 316
 Exploit-DB-Package 316
 Exploit-Framework 304
 Exploit Kit 354

Exploit-Writer 170
 ext3-Filesystem 62
 ext4-Dateisystem 199

F

Fail Open 245
 Fail-Open-Modell 245
 False Negative 159
 False Positive 159
 Faraday 347, 348
 Fedora-Linux 37
 Fehlerbericht 103
 Fehlkonfigurationen 273
 FHS 26, 44
 Fierce 215, 264
 File Inclusion 157
 filesnarf 280
 Filesystem Hierarchy Standard *siehe* FHS
 File Transfer Protocol 234
 Filternetz-Gateway 132
 Fingerabdruck
 digitaler 267
 Firewall 226, 229, 258, 354
 Firewall-Log 259
 Firmware 336
 Firmware-Datei 192
 Firmware-Image
 analysieren und extrahieren 336
 Foremost 339
 Forensik 29
 digitale 378
 Image erstellen 331
 Forensik-Modus 28
 Forensik-Tools 156
 Forensischer Ermittler 331
 Format String 171
 FPING 220
 FQDN 108
 FTP 234
 FTP-Datenverbindung 137
 FTP-Protokoll 137
 FTP-Server 246
 Funksignal 365
 Fuzzing 362

G

Galleta 340
 Garbage-String 338
 Genehmigungsprozess 168
 Gerichtsverfahren 334
 Gesamtrisiko 162
 GID-Variable 111
 Git 176

Git-Workflows 182
 GNOME3 36
 GNOME-Desktop-Umgebung 60
 GNOME-Shell 20
 GnuPG-Schlüssel 185
 GNU-Toolchain 358
 Google Direktiven 214
 GParted 82
 GPS 294
 GPU 172
 Grafikprozessor 172
 GRUB 79
 GRUB-Bootmenü 84
 GRUB-Konfiguration 79
 Gruppe
 dedizierte 49
 Gruppenvariable 111

H

Hacker
 ethischer 212
 Hacker-Befehlsshell 236
 Hacking 233
 Hacking-Labor 121
 Hail-Mary-Funktion 312
 Hardwareerkennung 66
 Hardwarekonfiguration 185
 Hash 332
 verschlüsselter 239
 Hash-Algorithmus 239
 Microsoft 323
 Hashdeep 340
 Hash-Wert 239, 340
 Header-Datei 191
 Headless-Umgebung 358
 Heap Corruption 171
 Heap-Speicher-Pointer 171
 Heimlicher Scan 223
 Herstellerhinweise 161
 Heuristik 300
 Hierarchie 46
 Hintertür 232
 Hintertürzugriff 222
 Home-Verzeichnis 45
 Hook 196
 binärer 196
 Hop 287
 Host 214
 virtueller 116
 Host-Betriebssystem
 Shell-Zugriff 154
 Hosterkennung 223, 224

- HTTP-Anforderungen 345
 - abfangen 298
 - anpassen 301
 - HTTP-Proxy 297
 - HTTP-Regression 278
 - HTTPS-Regression 278
 - https-Verbindung
 - protokollieren 295
 - http-Verbindung
 - protokollieren 295
 - HTTrack 215, 267
 - Hub 244
 - Hybrider Ansatz 168
 - Hydra 321
- I**
- i386-Plattformen 57
 - ICMP 135, 219, 260
 - Identitätsuntersuchung 338
 - Identität
 - verschleiern 280
 - IDS 258
 - Image
 - forensisches 331
 - Hash-Wert 334
 - Informationen
 - sammeln 213
 - Information Gathering 209
 - Informationsbeschaffung 165, 209, 210, 211, 213, 257
 - aktive 258
 - automatisierte Werkzeuge 214
 - Informationsquellen
 - mehrere 169
 - Informationssicherheit 164
 - Initialisierungsvektor 287
 - initrd-Generator 186
 - Installation
 - Fehlerbehebung 101, 103
 - Voraussetzungen 103
 - Installationsprotokoll 103
 - Integer Overflow 171
 - Integrated Penetration-Test Environment 347
 - Integrität 153, 332
 - Integrity 153
 - Internet Control Message Protocol 135
 - Internetsimulation 279
 - Intrusion-Detection-System 223, 258
 - Intrusionuntersuchung 338
 - IP-Adressbereich 236
 - IP-Adresse 107, 156, 215, 219, 307
 - IP-Adressraum 260
 - IPE 347
 - IRC-Client 236
 - IRC-Programm 236
 - ISO 30
 - ISO-Image 36, 59
 - Dateien hinzufügen 196
 - herunterladen 58
 - IV 287
- J**
- JavaScript 345
 - John the Ripper 27, 241, 322
 - JtR *siehe* John the Ripper
- K**
- Kali-Boot-USB-Stick 198
 - Kali Bug Tracker 33
 - Kali-Build
 - anpassen 192
 - Kali Evil Wireless Access Point 193
 - Kali-ISO 30
 - Kali-ISO-Image
 - erstellen 193
 - Kali Linux
 - Anpassungsmöglichkeiten 175
 - Kali-Linux-Image 30, 62
 - Kali Linux ISO of Doom 193
 - Kali Live 64
 - Kali-Live-ISO-Image 192
 - Kali-Live-System 201
 - Kali-Mirror 177
 - Kali Rolling 20
 - Kali Rolling ISO of Doom, Too 193
 - kali-rolling-Tool 181
 - kali-tweaks 140
 - Kali-USB-Stick 197
 - KDE 36
 - Kennwort
 - analysieren 350
 - für den Root-Benutzer 69
 - Kennwortangriff
 - offline 172
 - online 172
 - Kernel 37, 52
 - Konfigurationsdatei 188
 - konfigurieren 188
 - Neukompilierung 186
 - Quellen 187
 - Sicherheitsupdate 186
 - Standardkonfigurationen 188
 - Kernel-Code 186

Kernel-Image 191
 Kernel-Konfigurationsoberfläche 189
 Ketten 132
 Keylogger 232
 Keylogging 354
 Kimon 294
 Kismet 26, 293
 Kismon 294
 Klartext-Netzwerkprotokoll 281
 Klartextpasswort 239
 Klonvorgang 267
 Kommandozeile 41, 345
 Kommandozeilenbefehl 214
 Konfigurationsdatei 45, 273
 Konfigurationseinstellung 112
 Konfigurationsparameter 192
 Konfigurationsverzeichnis 194
 Konsole
 virtuelle 41, 101
 Kreuzkontamination 155
 Kritisches System 211
 Kryptografie 362

L

LAN Manager 240, 323
 Laufzeitinformation 52
 Laufzeitkonfiguration 279
 Leiser Scan 258
 libfreefare 176
 Linux 35
 Linux-Befehle 54
 Linux-Derivate 98
 Linux-Kernel 132, 358
 kompilieren 185
 Linux-Systemstruktur 72
 Linux Unified Key Set-up 85
 Live-Boot Hooks 196
 Live-Build 193
 live-build Skript 29
 Live-CD 36
 Live-Dateisystem
 Dateien hinzufügen 196
 Live-System 27
 LM-Passwort 323
 Logical Volume Management 85
 Logikbombe 354
 Login-Funktion 321
 Login-Shell 44
 LUKS 85, 86
 LUKS-Container 199
 LUKS-verschlüsselte Partition 199
 LVM 85
 LVM-Laufwerke 89

LVM-Tool 88
 LXDE 36

M

MAC-Adresse
 gefälschte 244
 macof 245, 281
 mailsnarf 281
 Maltego 26, 269
 Malware 354
 aufspüren 342
 Malwareuntersuchung 338
 Man-in-the-Middle-Angriff 147, 281
 Massenangriff 233
 Master Boot Record 80
 Master-Programm 354
 MBR 79
 MD4 340
 MD5-Hash 332
 Medusa 319
 Memory-Dump 344
 Metadaten 265, 336
 Metadateneintrag 336
 MetaGooFil 215, 265
 Meta-Paket 31, 195, 357
 Metasploit 27, 304
 Exploits 237
 Payloads 311
 Rang 307
 Metasploitable 231
 Metasploit-Dokumentation 309
 Meterpreter 235, 251
 Mobiles Breitband 107
 Mobiles Endgerät 167
 Mobilfunk 365
 mount 39
 Mounten 29
 msfconsole 305
 msgsnarf 281

N

Nacharbeiten 12
 Nachexploitation 377
 Namensauflösung 108
 Namensservers 108
 Navigation 365
 Nessus 232
 Netcat 251
 NetworkManager 106
 Netzwerk 233
 drahtloses 365
 ohne Internetzugang 316
 scannen 236
 Netzwerkanbindung 106

- Netzwerkdateisystem 39
 - Netzwerkdatenverkehr 244
 - ausspionieren 280
 - Netzwerkeinstellung
 - überprüfen 156
 - Netzwerkgestütztes Eindringen 246
 - Netzwerkinfrastruktur 271
 - Netzwerk-Intrusion 156
 - Netzwerkkonfiguration 67, 107
 - Netzwerkkontrolle 173
 - Netzwerkpaket 219
 - Netzwerkprotokoll-Analysator 284
 - Netzwerkrand
 - Geräte 219
 - Netzwerkschnittstelle 107
 - Netzwerk-Sniffer 281
 - Netzwerk-Sniffing 244
 - Netzwerk-Sniffing-Attacke 281
 - Netzwerkverkehr
 - analysieren 280
 - ausspähen 244
 - ausspionieren 281
 - erfassen 289
 - überwachen 281
 - NFC-Karte 176, 183
 - NFS 39
 - Nikto 249, 277
 - NIST-Sonderpublikation 161
 - Nmap 26, 216, 218, 221, 223, 225, 227, 236, 257, 307
 - Befunde 237
 - Portscan 222
 - Script Engine 229
 - Versionsscan 227
 - NOPS 305
 - Normierung
 - Assessments 168
 - NSE 216, 218, 229
 - NSE-Skript 230
 - NTLM 324
 - NTP-Server 70
 - NULL-Scan 227, 228, 229
 - NVIDIA-Grafik 99
- O**
- Offener Port 212
 - Offensive Security 22, 31
 - Office-Dokument 265
 - Online-Shop 299
 - Open Source 26, 36
 - Open-Source-Software 36
 - OpenVAS 26, 142, 216, 232, 273, 306, 350
 - Open Vulnerability Assessment System 232
 - OpenWRT-Router 294
 - OSVDB 232
 - OWASP 301
 - OWASP-ZAP 267
- P**
- Package Manager 78
 - Packaging-Tool 182, 183
 - Paket
 - ändern 181
 - anpassen 175
 - neu erstellen 177
 - Versionsnummer 181
 - Paketabhängigkeit 176
 - Paketerstellungsprozess 183
 - Paros 247
 - Partition
 - verschlüsselte 85
 - Verschlüsselung 63
 - Partitionierung 70
 - geführte 70
 - Partitionierungstool 85, 89
 - Partitionsmodus
 - manueller 74
 - Pass the hash 239
 - Passwort 322
 - knacken 238
 - zurücksetzen 243
 - Passwort-Attacke 172
 - Passwortcracker
 - online 234
 - Passwortcracker-Tool 325
 - Passwortcracking 240
 - lokal 240
 - Passwort-Dump 351
 - Passwörter
 - decodieren 281
 - Passwörter knacken
 - Linux 242
 - OS X 242
 - Windows 240
 - Passwort-Hash 234, 238
 - Windows 327
 - Passwort-Hash-Datei 239
 - Passwortsicherheit 371
 - Passwort-Wörterbuch 318
 - Patch 307
 - Problem beheben 232
 - Patch-Level 159
 - Patch-Management-System 185
 - Payload 233, 305, 306, 309
 - PCAnywhere 234
 - PCAP 294
 - PCI-Gerät 52

- PCMCIA-Karte 52
 - Penetrationstest 164
 - Ablauf 209
 - traditioneller 164
 - Vier-Schritte-Prozess 209
 - Penetrationstester 226
 - Penetration Testing Execution Standard 213
 - Permission to Attack 169
 - Persistence-Start 63
 - Persistenz 27, 62, 196, 197
 - verschlüsselt 199
 - Persistenzdateisystem 199
 - Persistenzpartition
 - verschlüsselt 201
 - Phishing 314, 315, 354
 - Web-Vorlage 315
 - Phishing-Seite 315
 - Phreaker 355
 - Physikalische Adresse 259
 - Physische Partition 87
 - PID 47
 - Ping 216, 219
 - Hacker-Werkzeug 220
 - Ping-Scan 259
 - Pipal 350
 - Port 217
 - Anzahl 221
 - ermitteln 217
 - offen 212, 216
 - Verkehrsaufkommen 217
 - Portscan 212, 216, 221, 222, 259, 312
 - PostgreSQL-Cluster 115
 - PPPoE 107
 - Primäres Betriebssystem 103
 - Programmausführungsfluss
 - steuern 171
 - Programmkonfiguration 36
 - Proof-of-Concept-Code 170
 - Protokoll
 - verbindungsloses 225
 - verbindungsorientiertes 225
 - Proxy 297
 - konfigurieren 297
 - ZAP 301
 - Proxy-Adresse 78
 - Prozess 39
 - verwalten 47
 - Prozess-ID 47
 - Prozessorarchitektur 159
 - Prozesspriorität 40
 - PTA 169
 - PTES 213
- Q**
- Quellpaket 177
 - aktualisieren 184
 - erstellen 185
 - Quellformat 182
- R**
- Race Conditions 157
 - RainbowCrack 27
 - Randgeräte 219
 - Raspberry Pi 95, 294
 - RDP 234
 - Recherche 213
 - Recon 209
 - Reconnaissance 209
 - RecordMyDesktop 351
 - Recovery 340
 - redfang 176
 - Redirection 46
 - Regelerstellung 138
 - Regression 279
 - Remote-Codeausführung 307
 - Remotecomputer 309
 - Remote Desktop Protocol 234
 - Remotedienst 234
 - Remote-Shell 241
 - Remotezugriff 112
 - Remotezugriffsdienst 234
 - Report 301
 - Repository 33, 98
 - Request for Comments 227
 - Ressourcenverbrauch 170
 - Reverse Engineering 373
 - Reverse-Payload 312
 - RFC 227
 - RFID-Tag 364
 - Richtlinien
 - Debian 32
 - Kali Linux 32
 - Richtlinien für Sicherheitsexperten 213
 - Ringbuffer 52
 - Risiko 155
 - Risikobewertung 129, 160, 163
 - Rolling Distribution 21
 - Root 39
 - Rootkit 251, 338, 355
 - Rootkit-Erkennung 338
 - Root-Konto 236
 - Root-Passwort 272
 - Root-Rechte 236
 - Router 283
 - RST-Paket 223

S

SAM 242

SAM-Datei 240, 326, 327

Samdump2 241, 326

SAM-Sperre 240

Scan

authentifizierter 160

automatisierter 159

leiser 258

Scannen 165

Schnittstelle 38, 282

dedizierte 189

Schnüffeln 376

Schwachstelle 155, 157, 306

ausnutzen 233

ermitteln 218

scannen 250

Webapplikationen 249

Schwachstellenanalyse 157, 158, 164

Tools 273

Schwachstellenanalyse-Tools

automatisierte 170

Schwachstellenerkennung 369

Schwachstellen-Scan 160, 212, 216,

218, 225, 232

automatisiert 303

Ergebnisse 159

Nikto 277

ZAP 304

Schwachstellen-Scanner 142, 160, 232

Metasploit 305

SD-Karte

startfähig 96

Searchsploit 316

Secure Shell 234

Service-Manager 119

Service-Unit 119

SET 176, 314, 315

setgid 48

SET-Power-User 184

setuid 48

SHA 242

SHA-256 340

shadow 242

Shell 42, 43, 222

Shell-Zugriff 154, 238

administrativ 222

Shrink Wrap Code 355

Sicherheitscheck 155

Sicherheitslücke 158, 233, 273, 355

Sicherheitsparameter 164

Sicherheitsprozesse 164

Sicherheitsrichtlinien

definieren 127

Sicherheitsupdate

Kernel 186

Siege 278

URL-Formate 279

Signatur 158

erstellen 159

Signaturset 161

Sitemap 300

Skipfish 300

Skript

ausführen 218, 230

Slackware 20

Sleuth Kit 333

Sniffing 280, 284

Sniffing Tools 160

SNMP 226

Social Engineering 214, 355, 375

Social-Engineering-Angriff 313

Social-Engineer Toolkit (SET) 26, 314

Software-RAID 85

Softwareversion 159

Source-Paket 98

Soziale Dienste 270

Soziales Engineering *siehe* Social
Engineering

Spam 355

Speicherbeschädigung 171

Speicher-Dumb 343

Speicherforensik 342, 343

Speichermedium

bootfähiges 64

Speicherverbrauch 186

Spider 303

automatisiert 248

Spiderangriff 298

ZAP 303

Spoofing 280, 355, 376

Spracheinstellung 65

Spyware 355

SQL-Befehle 157

SQL-Injection 154, 157, 172, 248, 355

SSH 112, 234

SSH-Host-Schlüssel 113

sshmitm 281

SSID 289

SSLstrip 147

SSL-Zertifikat 145

Stable Distribution 20

Stack Buffer Overflow 171

Standard-Angriffsziel 167

Standard-Anwendung 358

Standard-Assessment 166

Standardkonfiguration 195

optimieren 175

Standard-Linux-Kernel 67
 Standardnetzwerkkonfiguration 106
 Standardportnummer 217
 Standardports 224
 Standard-Shell 110
 Startmedium 203
 Statistiken 350
 Steganografie 362
 Subdomänen
 aufspüren 262
 Subnetz 224
 Superuser-Root-Konto 69
 SWAP-Partition 28, 77, 89, 156
 Switch 244, 283
 SYN/ACK 222
 SYN-Flag 229
 SYN-Scan 222, 259
 starten 223
 syskey 326
 SYSTEM 242
 Systemd 52
 Systeme
 kritische 211
 Systemressource 47
 Systemsicherung 331
 SysVinit-Methode 19

T

Target-Unit 119
 Tarnung 223
 Tastaturlayout 66
 TCP 225
 TCP-Port 221
 TCP-RFC 228
 TCP-Stack 260
 TCP-Verbindung 114
 TCP-Verbindungsscan 223, 224, 225
 Telnet 234
 Terminal 41
 Texteditor 46
 TFTP 226
 TheHarvester 215, 262
 Threat 355
 Threats pro Scan 160
 Tool
 Dsniff 280
 Exploitation 304
 Man-in-the-Middle-Angriffe 281
 Penetrationstest 295
 Schwachstellenanalyse 273
 Sniffing 280
 Spoofting 280

Tools

 automatisierte 161
 für Attacken 287
 zur Informationssammlung 257
 Torrent 59
 Traditioneller Penetrationstest 157, 164
 Transaktionsinformationen 279
 Trojaner 355
 True Negative 159
 True Positive 159

U

Überwachungsdienst 156
 Ubuntu 19
 UDP 225
 UDP-Port 221, 225
 UDP-Scan 225, 226
 UEFI 37
 UID-Variable 111
 Umgebungsvariable 44, 45
 Unified Sniffing 283
 Unix 38, 48
 Unix-basiertes Betriebssystem 61
 Unix-Crypt(3)-Hash 322
 Unix-Derivate 98
 Upstream 182
 Upstream-Git-Repository 182
 Upstream-Version 175
 packen 184
 urlsnarf 281
 USB-Gerät 52
 User-Account 110
 User-Agent 347
 User-Space 53
 User-Space-Bibliothek 192

V

Validierungsprozess
 Tools 169
 Variable 44
 Verbindungsaufbau 222
 Verbindungsloses Protokoll 225
 Verbindungsorientiertes Protokoll 225
 Verfügbarkeit 153
 Verschleierung 223
 Verschlüsselte Partition 85
 Verschlüsselter Hash 239
 Verschlüsselung 326
 Verschlüsselungs-Passphrase 86
 Verschlüsselungsschlüssel 86
 Vertraulichkeit 153
 Verzeichnis 39
 Verzeichnisbaum 42

VFAT 39
 Virtual Network Computing 234
 Virtuelle Konsole 41, 101
 Virtueller Host 116
 Virtuelles Dateisystem 52
 Virus 356
 VNC 309
 VNC-Injektion 311
 VNC-Payload 241
 Voice-over-IP-(VoIP-)System 366
 Volafx 342
 Volatility 343
 Volume-Gruppe 85
 Voreinstellungsdatei 203
 erstellen 204
 initrd 203
 Netzwerk 204
 Startmedium 203
 VPN 107
 VPN-Netzwerk 276
 Vulnerability 157
 Vulnerability Analysis 158
 Vulnerability-Scanner 273

W

w3af 247
 Webanwendung 158, 166, 300
 Webanwendungs-Assessment 156
 Webapplication 158
 Web Application Audit und Attack Framework 247
 Webapplikation
 Schwachstellen 249
 Webframework 300
 Webgestützter Angriff 246
 Webhacking 247, 249
 Webkit-Rendering 345
 webmitm 281
 Web-Penetrationstest 302, 347
 Webpräsenz
 Unternehmen 246
 Webscanner 250
 WebScarab 249, 295
 Web-Schwachstelle 171
 Webseite 267
 analysieren 249
 Offline-Kopie 267
 Webservers 116, 277
 Informationen gewinnen 221
 Web-Sicherheit 369
 websploit 281

WEP 287
 WEP-Schlüssel 287, 291
 knacken 288
 White-Box-Assessment 167
 Windows 367
 Windows-Eingabeaufforderung 235
 Windows-Installation 81
 Windows-LM-Passwörter 240
 Windows-NT-basierte Systeme 327
 Windows-Partition 74
 verkleinern 82
 Win-KeX 94
 Wireless-Assessments 156
 Wireless Injection 19
 Wireless Security Assessment 156
 Wireless Wide Area Network 107
 Wireshark 27, 246, 284
 WLAN-Hacking 287
 WLAN-Netzwerk
 aufspüren 294
 WLAN-Sicherheit 372
 Worst-Case-Szenario 165
 WSL-Distribution 91
 Wurm 356
 WWAN 107

X

XFCE 36
 Xmas-Scan 227, 228
 XSS 248, 356
 XSS-Angriff 172

Z

ZAP 301
 Zed Attack Proxy 301
 Zeitbombe 354
 Zenmap 221, 257
 Zero-Day-Exploit 273
 Zielnetzwerk 158
 Zielorganisation 215
 Ziel-PC
 steuern per Kommandozeile 222
 Zombie-Drohne 356
 Zonentransfer 264
 ZSH-Terminal 348
 Zugangspunkt 217
 Zugriff
 administrativer 233
 festigen 212
 Zugriffsbeschränkung 119