



Jürgen
Ebner

3. 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
1 Einführung	19
1.1 Unterschied zwischen Kali und Debian	19
1.2 Ein Stück Geschichte	19
1.3 Kali Linux – für jeden etwas	21
1.3.1 Varianten von Kali Linux	22
1.4 Die Hauptfeatures	25
1.4.1 Live-System	26
1.4.2 Ein maßgeschneideter Linux-Kernel	28
1.4.3 Komplett anpassbar	28
1.4.4 Ein vertrauenswürdiges Betriebssystem	30
1.4.5 Auf einer großen Anzahl von ARM-Geräten verwendbar ..	30
1.5 Richtlinien von Kali Linux	31
1.5.1 Benutzer ohne root-Rechte	31
1.5.2 Netzwerkdienste sind standardmäßig deaktiviert	32
1.5.3 Eine organisierte Sammlung von Tools	32
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-Bootloaders 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
4.4	Sichern und Überwachen mit Kali Linux	123
4.4.1	Sicherheitsrichtlinien definieren	124
4.4.2	Mögliche Sicherheitsmaßnahmen	126
4.4.3	Netzwerkservices absichern	127
4.4.4	Firewall- oder Paketfilterung	128
4.5	Weitere Tools installieren	136
4.5.1	Meta-Packages mit kali-tweaks installieren	136
4.5.2	Terminator statt Terminal	137
4.5.3	OpenVAS zur Schwachstellenanalyse	138
4.5.4	SSLstrip2	141
4.5.5	Dns2proxy	142
4.6	Kali Linux ausschalten	143
4.7	Zusammenfassung	143
Teil II Einführung in Penetration Testing		147
5	Einführung in Security Assessments	149
5.1	Kali Linux in einem Assessment	151
5.2	Arten von Assessments	152
5.2.1	Schwachstellenanalyse	154
5.2.2	Compliance-Test	159
5.2.3	Traditioneller Penetrationstest	160
5.2.4	Applikations-Assessment	162
5.3	Normierung der Assessments	164
5.4	Arten von Attacken	165
5.4.1	Denial of Services (DoS)	166
5.4.2	Speicherbeschädigungen	167
5.4.3	Schwachstellen von Webseiten	167
5.4.4	Passwort-Attacken	168
5.4.5	Clientseitige Angriffe	169
5.5	Zusammenfassung	169

6	Kali Linux für Security Assessments vorbereiten	171
6.1	Kali-Pakete anpassen	171
6.1.1	Quellen finden	173
6.1.2	Build-Abhängigkeiten installieren.	176
6.1.3	Änderungen durchführen	177
6.1.4	Build erstellen	181
6.2	Linux-Kernel kompilieren	181
6.2.1	Einführung und Voraussetzungen	182
6.2.2	Quellen finden	183
6.2.3	Kernel konfigurieren	184
6.2.4	Pakete kompilieren und erstellen	187
6.3	Erstellen eines individuellen Kali-Live-ISO-Images	188
6.3.1	Voraussetzungen	189
6.3.2	Erstellen von Live-Images mit verschiedenen Desktop-Umgebungen	190
6.3.3	Ändern der Liste installierter Pakete.	191
6.3.4	Verwenden von Hooks zum Optimieren des Live-Images	192
6.3.5	Hinzufügen von Dateien zum ISO-Image oder Live-Filesystem	192
6.4	Hinzufügen von Persistenz auf einem USB-Stick	193
6.4.1	Erstellen einer unverschlüsselten Persistenz auf einem USB-Stick	194
6.4.2	Erstellen einer verschlüsselten Persistenz auf einem USB-Stick	195
6.4.3	Verwenden von mehreren Persistenzspeichern.	197
6.5	»Automatisierte« Installation	198
6.5.1	Antworten auf Installationsabfragen vorbereiten.	198
6.5.2	Erstellen der Voreinstellungsdatei.	200
6.6	Zusammenfassung	201
6.6.1	Kali-Pakete ändern	201
6.6.2	Linux-Kernel neu kompilieren.	202
6.6.3	Benutzerdefinierte ISO-Images erstellen	203
7	Ablauf eines Penetrationstests	205
7.1	Informationen sammeln	209
7.1.1	Was nun?.	209
7.1.2	Kali-Tools zur Informationsbeschaffung	211
7.1.3	Informationen nach angreifbaren Zielen durchsuchen	211

7.2	Scannen	212
7.2.1	Pings	215
7.2.2	Portscan	217
7.2.3	Nmap Script Engine – Transformationen eines Tools	225
7.2.4	Schwachstellen-Scan	228
7.3	Eindringen über das lokale Netzwerk	229
7.3.1	Zugriff auf Remotedienste	230
7.3.2	Übernahme von Systemen	231
7.3.3	Passwörter hacken	234
7.3.4	Abrissbirnen-Technik – Passwörter zurücksetzen	239
7.3.5	Netzwerkverkehr ausspähen	240
7.4	Webgestütztes Eindringen	242
7.4.1	Schwachstellen in Webapplikationen finden	245
7.4.2	Webseite analysieren	245
7.4.3	Informationen abfangen	245
7.4.4	Auf Schwachstellen scannen	246
7.5	Nachbearbeitung und Erhaltung des Zugriffs	246
7.6	Abschluss eines Penetrationstests	248
7.7	Zusammenfassung	249
Teil III Tools in Kali Linux		251
8	Tools zur Informationsbeschaffung und Schwachstellenanalyse	253
8.1	Tools zur Informationssammlung	253
8.1.1	Nmap – Das Schweizer Taschenmesser für Portscanning	253
8.1.2	TheHarvester – E-Mail-Adressen aufspüren und ausnutzen	258
8.1.3	Dig – DNS-Informationen abrufen	260
8.1.4	Fierce – falls der Zonentransfer nicht möglich ist	260
8.1.5	MetaGooFil – Metadaten extrahieren	261
8.1.6	HTTrack – Webseite als Offline-Kopie	263
8.1.7	Maltego – gesammelte Daten in Beziehung setzen	265
8.1.8	Legion – Automation in der Informationsbeschaffung	267
8.2	Schwachstellenanalyse-Tools	269
8.2.1	OpenVAS – Sicherheitslücken aufdecken	269
8.2.2	Nikto – Aufspüren von Schwachstellen auf Webservern	273
8.2.3	Siege – Performance Test von Webseiten	274

8.3	Sniffing und Spoofing	276
8.3.1	Dsniff – Sammlung von Werkzeugen zum Ausspionieren von Netzwerksdatenverkehr	276
8.3.2	Ettercap – Netzwerkverkehr ausspionieren	277
8.3.3	Wireshark – der Hai im Datenmeer	280
9	Tools für Attacken	283
9.1	Wireless-Attacken	283
9.1.1	aircrack-ng	283
9.1.2	wifiphisher	287
9.1.3	Kismet	289
9.2	Webseiten-Penetration-Testing	291
9.2.1	WebScarab	291
9.2.2	Skipfish	296
9.2.3	Zed Attack Proxy	297
9.3	Exploitation-Tools	300
9.3.1	Metasploit	300
9.3.2	Armitage	308
9.3.3	Social Engineer Toolkit (SET)	309
9.3.4	Searchsploit	312
9.4	Passwort-Angriffe	314
9.4.1	Medusa	315
9.4.2	Hydra	317
9.4.3	John the Ripper	318
9.4.4	Samdump2	322
9.4.5	chntpw	323
10	Forensik-Tools	327
10.1	Dcfldd – Abbild für forensische Untersuchung erstellen	327
10.2	Autopsy	329
10.3	Binwalk	332
10.4	chkrootkit	334
10.5	Bulk_extractor	334
10.6	Foremost	335
10.7	Galleta	336
10.8	Hashdeep	336
10.9	Volafox	338
10.10	Volatility	339

11	Tools für Reports	341
11.1	Cutycapt	341
11.2	Faraday-IDE	343
11.3	Pipal	346
11.4	RecordMyDesktop	347
A	Terminologie und Glossar	349
B	Übersicht Kali-Meta-Pakete.	353
B.1	kali-linux.	353
B.2	kali-linux-full	353
B.3	kali-linux-all	354
B.4	kali-linux-top10	354
B.5	kali-linux-forensic	354
B.6	kali-linux-gpu.	355
B.7	kali-linux-pwtools.	355
B.8	kali-linux-rfid	355
B.9	kali-linux-sdr	355
B.10	kali-linux-voip.	355
B.11	kali-linux-web.	356
B.12	kali-linux-wireless	356
C	Checkliste: Penetrationstest	357
C.1	Scope	357
C.2	Expertise.	359
C.3	Lösung	359
D	Installation von Xfce und Undercover-Modus	361
	Stichwortverzeichnis	365

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 Metapakete 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 heißtt, 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 ressourcen hungriger. 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 weit verbreitet 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 Gerätedateien: 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 Gerätedatei 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 abruft. Dank der Abstraktion des Dateisystems muss man sich keine Gedanken machen, wie diese Festplatte ange schlossen 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 Kerneln, 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 größtentwents, 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 Zugriffsberichtigungen 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 Befehlsinterpretierer (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.

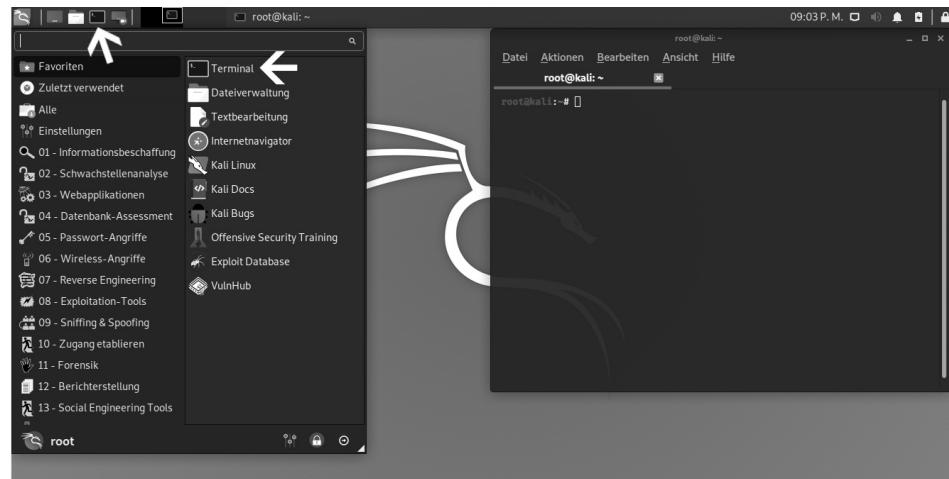


Abb. 2.2: Terminal aufrufen

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 **Bourne Again Shell**). 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 Quelldatei Zielformat`.

```
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 entfernbares 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 Befehlsinterpretator 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 aufzulisten 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 Verzeichnisse zu bereinigen. In dieser Spezifikation wurde vereinbart, dass Konfigurationsdateien unter *~/.config*, Cache-Dateien unter *~/.cache* und Anwendungsdateien unter *~/.local* (oder deren Unter-verzeichnissen) 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.

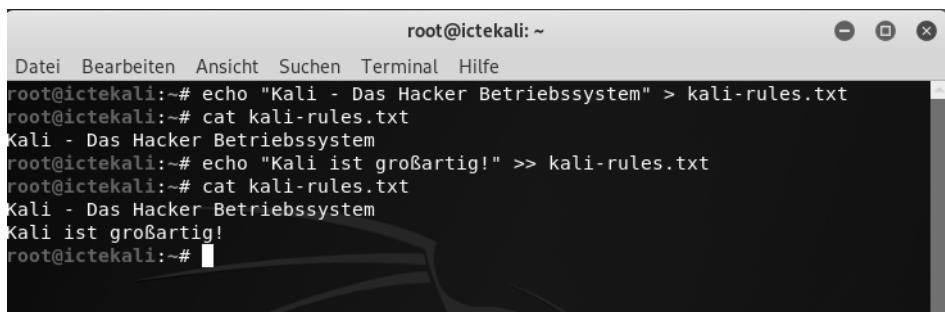
Grafische Desktops verfügen normalerweise über Verknüpfungen, mit denen Inhalte des Verzeichnisses `~/Desktop/` angezeigt werden können (oder auch entsprechende Übersetzungen für Systeme, die nicht auf Englisch konfiguriert sind).

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.



The screenshot shows a terminal window titled "root@ictekali: ~". The user has run several commands to demonstrate redirection:

```
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 Befehlsinterpretator 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
- **Besitzergruppe (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 Buchstabsymbole. 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 ($4 = \text{nur Leserechte}$)

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 Gruppenbesitzer, 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
Filesystem      1K-Blöcke  Benutzt  Verfügbar  Verw% Eingeht 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. Einsticken 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 Systemd⁶ 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

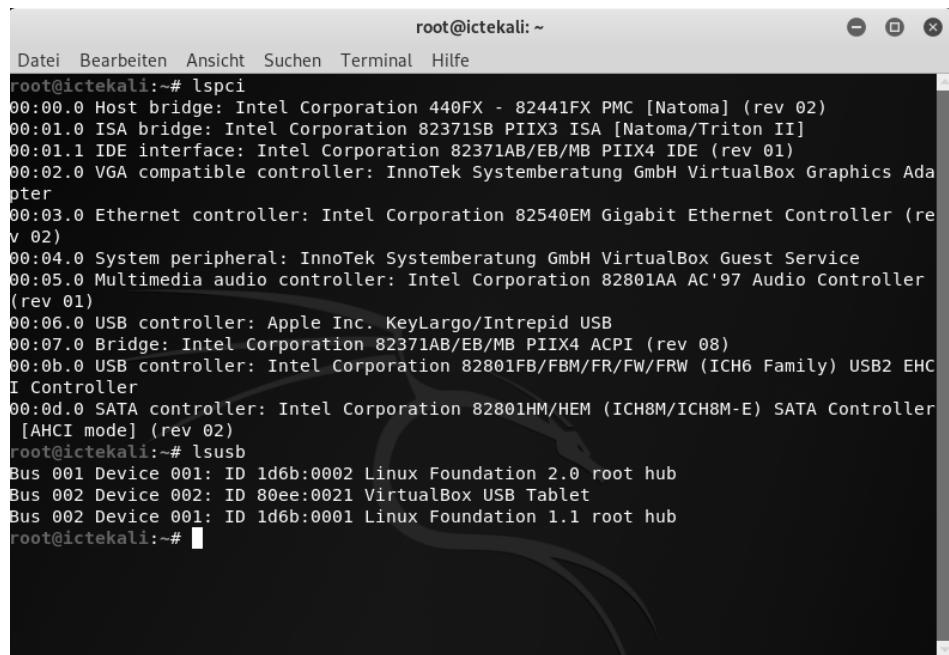
Der Kernel speichert viele Details über erkannte Hardware in den virtuellen Dateisystemen `/proc/` und `/sys/`. Mehrere Tools fassen diese Details zusammen. Dazu gehören

- `Ispci` (im Paket `pciutils`), das PCI-Geräte auflistet
- `Isusb` (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 relevanten Ergebnissen führt. Die Tools `pciutils` und `usbutils` werden bereits im Kali-Basisystem 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 EHC I 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 Kerdienstprogramme 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` – Verketten 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
 - `journaldctl` – 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
 - `lspcmia` – 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 272
Access Point 283
ACK-Paket 218, 219
address resolution protocol 254
Address Space Layout Randomization 154
Administrationsrecht 31
Administrativer Zugang 208
Administrativer Zugriff 229
Administratorkonto 325
knacken 234
Administratorpasswort
 zurücksetzen 324
Adresse
 physische 255
Adware 349
aircrack-ng 26, 283
Aktive Informationsbeschaffung 254
amd64-Plattformen 57
Analysieren
 von Kennwörtern 276
Android-Exploit 313
Anforderungen
 behördliche 159
 branchenspezifische 159
Angriff 349
 clientseitiger 169
 webgestützter 242
Angriffserkennungssystem 254
Anmeldeinformationen 278
Ansatz
 hybrider 164
Anti-Exploit-Technologie 154
Anwenderaktualisierung 169
Anwendungs-Assessment 164
Anwendungsdatei 45
Anwendungskonfigurationsdatei 45
Anwendungsverhalten 163
Apache-Konfigurationsanweisungen 118
Apache-Prozess 116
Apache-Standardmodule 116

Apache-Webserver
 konfigurieren 115
Applikations-Assessment 153, 162
APT 177
Arbeitsspeicher 86
ARM-Computer 95
Armel-Plattformen 57
Armhf-Plattformen 57
Armitage 232, 233, 308
ARM-Plattformen 57
ARP 254
ARP-Cache 255
ARPreplay-Attacke 286
ARP-Request 286
arpspoof 277
ARP-Spoofing 279
Assessment
 Arten von 152
 Installation 151
Assessment-Plattform 156
Aufklärung 205, 210, 259
Aufklärungsphase 206
Auslagerungsdatei 86
Auslagerungspartition 86
Auswirkungen 158
Authentifizierter Scan 156
Authentifizierung
 Access Point 286
 Basic 118
Authentifizierungsebene 243
Automatisierte Installation 151
Automatisierter Scan 155
Automatisierte Tools 157
Automatisierung 217
Autopsy 329
 Analyse 331
Availability 149

B

Backdoor 349
Back-End-Seitengenerierungslogik 167
BackTrack 19

- Banner 227
 Base64-Codierung 296
 Bash 42
 Bedrohung 151
 Bedrohungsstufe 273
 Befehle
 Übersicht 54
 Befehlsinterpret 47
 Befehlszeileninterpret 42
 Befehlszeilenwerkzeuge 92
 Befehlszeile *siehe* Kommandozeile
 Belastungstest 275
 Benchmarking 274
 Benutzerkennwort 70
 Berechtigungssystem 47
 Bereitstellungspunkt 63, 322
 Bericht
 erstellen 162, 208
 Berichterstattung 208
 Betriebssystemversion 154
 Bettercap 26
 BID-Nummer 303
 Bildanalyse 335
 Binärer Hook 192
 Binär-Image 332
 Binärpaket 181
 Bind-Payload 307
 Binwalk 332
 BIOS 37
 Black-Box-Assessment 163
 Bootfähiges Speichermedium 64
 Bootkey 322
 Bootloader 37, 79, 182
 Bootloader-Konfiguration
 ändern 199
 Boot-Parameter 198, 200
 Bot 349
 Botnet 349
 Breitband
 mobiles 107
 Bridged Sniffing 279
 Broadcast 226, 240
 Brute Force 275
 Brute-Force-Anmelde tool 315
 Brute-Force-Attacke 230, 318, 349
 Brute-Force-Methode 314, 316
 BSI 228
 BSSID 285
 Buffer-Overflow 153, 167, 349
 Bugtraq ID Database 303
 Build-Abhängigkeiten installieren 176
 Build-Environment 179
 Build-Essential-Paket 182
 Build-Option 179
 Build-Prozess 181
 Bulk_extractor 334
 Burp Suite 243
- C**
- Caching-Proxy 78
 CANVAS 300
 Capture-Filter 281
 CentOS 37
 chntpw 323
 Chromebook 59
 Chroot Hooks 192
 chrootkit 334
 Chroot-Umgebung 192
 CIA-Triade 149
 Clientseitiger Angriff 169
 Clone Phishing 349
 Closed-Source-Datei 28
 Cloud-Dienstanbieter 165
 Cloud-Installation 24
 Cloud-Service 165
 Codeausführung 228
 Code-Execution-Exploit 166
 Common Vulnerabilities Exposures 303
 Compliance 159
 Compliance-Framework 160
 Compliance-Test 153, 159, 160
 Confidentiality 149
 Connect-Scan 255
 Cookies 336
 CORE Impact 300
 Cracker 349
 Cracks pro Sekunde
 messen 319
 Crawler 264
 Crawling 245
 Cross Site Scripting (XSS) 168, 244, 352
 Cryptcat 247
 Cutycapt 341
 CVE 228
 CVE-Nummer 156, 303
 CVSS-Score 157
 Cyber-Hygiene 134
 Cyberuntersuchung 335
- D**
- Daemon-Daten 73
 Daemons 111
 Data Execution Prevention 154
 Dateisystem 38, 39
 virtuelles 52
 Dateisystemformat 39

- Datenbankserver
 PostgreSQL 113
- Datenintegrität 328
- Datenpaket
 suchen 281
 WLAN 284
- Datenstruktur
 wiederherstellen 335
- Datenverkehr 215, 241
- Dcfldd 327
- DDoS 350
- Debconf-Datenbank 102
- Debconf-Fragen 199
- Debconf-Voreinstellungen 198
- DEBEMAIL 178
- DEBFULLNAME 178
- Debian 19, 37
- Debian-Kernel-Handbuch 182
- Debian-Kernel-Paket 182
- Debian-Live-Systemhandbuch 190
- Debian-Packaging 177
- Debian-Paket 182, 187
- Debian-Quellverwaltungs-Datei 173
- Debian-Richtlinien 31
- Debian Unstable 28
- Debugging-Symbol 188
- Debug-Meldung 172
- Dedizierte Gruppe 49
- Dedizierte Schnittstelle 185
- Default Desktop 82
- Default Gateway 108
- Denial of Service 166, 350
- Denial-of-Service-Angriff 241
- Denial-of-Service-Bedingung 166
- Desktop-Anwendungen 162
- Desktop-Sitzung 41
- Desktop-Umgebung 29, 190
- Device-Mapper 85
- dget-Quellpaket 176
- DHCP 109, 222
- DHCP-Einstellungen 152
- Dienst
 aktiver 212
- Dig 211, 260
- Digitaler Fingerabdruck 263
- Display-Filter 281
- Distribution 19, 35
- DMZ 254
- DNS 222
- Dns2proxy 142
- DNS-Abfrage 260
- DNS-Server 108, 210
- dnsspoof 277
- Domain Controller 347
- Domänenadministratorkonto 234
- DoS 150, 166, 350
- DoS-Angriff 166
- DoS-Ergebnis 314
- dpkg-Dateien 172
- Drei-Wege-Handshake 218, 221
- Drohne 290
- Dsniff 241, 276
- dsniff 276

E

- EDB-ID 156
- Eindringen 162, 208
 netzwerkgestütztes 242
- Eingangsbuffer 131
- Einstellungs-Reiter 294
- Eintrittswahrscheinlichkeit 157
- E-Mail-Adressen
 aufspüren 258
- E-Mail-Passwort 278
- Embedded Device 59
- Encoder 301
- Endgeräte
 mobiles 163
- Enlightment 36
- Ermittler
 forensischer 327
- Erstellungszeitpunkt 179
- Ethernet-Netzwerk 277
- Ethischer Hacker 208
- Ettercap 277
 Sniff-Modi 279
- Exploit 153, 229, 247, 302, 350
 Definition 153
- Exploitation-Tools 300
- Exploit-Code 166
- Exploit-Datenbank 312
- Exploit-DB-Package 312
- Exploit-Framework 300
- Exploit Kit 350
- Exploit-Writer 166
- ext3-Filesystem 62
- ext4-Dateisystem 195

F

- Fail Open 241
- Fail-Open-Modell 241
- False Negative 155
- False Positive 155
- Faraday 343, 344
- Fedora-Linux 37

Fehlerbericht 103
Fehlkonfigurationen 269
FHS 25, 44
Fierce 211, 260
File Inclusion 153
filesnarf 276
Filesystem Hierarchy Standard *siehe* FHS
File Transfer Protocol 230
Filternetz-Gateway 128
Fingerabdruck
 digitaler 263
Firewall 222, 225, 254, 350
Firewall-Log 255
Firmware 332
Firmware-Datei 188
Firmware-Image
 analysieren und extrahieren 332
Foremost 335
Forensik 28
 Image erstellen 327
Forensik-Modus 27
Forensik-Tools 152
Forensischer Ermittler 327
Format String 167
FPING 216
FQDN 108
FTP 230
FTP-Datenverbindung 133
FTP-Protokoll 133
FTP-Server 242

G

Galleta 336
Garbage-String 334
Genehmigungsprozess 164
Gerichtsverfahren 330
Gesamtrisiko 158
GID-Variable 111
Git 172
Git-Workflows 178
GNOME3 36
GNOME-Desktop-Umgebung 60
GNOME-Shell 20
GnuPG-Schlüssel 181
Google Direktiven 210
GParted 82
GPS 290
GPU 168
Grafikprozessor 168
GRUB 79
GRUB-Bootmenü 84
GRUB-Konfiguration 79

Gruppe
 dedizierte 49
Gruppenvariable 111

H

Hacker
 ethischer 208
Hacker-Befehlsshell 232
Hacking 229
Hacking-Labor 121
Hail-Mary-Funktion 308
Hardwareerkennung 66
Hardwarekonfiguration 181
Hash 328
 verschlüsselter 235
Hash-Algorithmus 235
 Microsoft 319
Hashdeep 336
Hash-Wert 235, 336
Header-Datei 187
Heap Corruption 167
Heap-Speicher-Pointer 167
Heimlicher Scan 219
Herstellerhinweise 157
Heuristik 296
Hierarchie 46
Hintertür 228
Hintertürzugriff 218
Home-Verzeichnis 45
Hook 192
 binärer 192
Hop 283
Host 210
 virtueller 116
Host-Betriebssystem
 Shell-Zugriff 150
Hosterkennung 219, 220
HTTP-Anforderungen 341
 abfangen 294
 anpassen 297
HTTP-Proxy 293
HTTP-Regression 274
HTTPS-Regression 274
https-Verbindung
 protokollieren 291
http-Verbindung
 protokollieren 291
HTTrack 211, 263
Hub 240
Hybrider Ansatz 164
Hydra 317

I

- i386-Plattformen 57
- ICMP 131, 215, 256
- Identitätsuntersuchung 334
- Identität
 - verschleiern 276
- IDS 254
- Image
 - forensisches 327
 - Hash-Wert 330
- Informationen
 - sammeln 209
- Information Gathering 205
- Informationsbeschaffung 161, 205, 206, 207, 209, 253
 - aktive 254
 - automatisierte Werkzeuge 210
- Informationsquellen
 - mehrere 165
- Informationssicherheit 160
- Initialisierungsvektor 283
- initrd-Generator 182
- Installation
 - Fehlerbehebung 101, 103
 - Voraussetzungen 103
- Installationsprotokoll 103
- Integer Overflow 167
- Integrated Penetration-Test Environment 343
- Integrität 149, 328
- Integrity 149
- Internet Control Message Protocol 131
- Internetsimulation 275
- Intrusion-Detection-System 219, 254
- Intrusionuntersuchung 334
- IP-Adressbereich 232
- IP-Adresse 107, 152, 211, 215, 303
- IP-Adressraum 256
- ipe 343
- IRC-Client 232
- IRC-Programm 232
- ISO 29
- ISO-Image 36, 59
 - Dateien hinzufügen 192
 - herunterladen 58
- IV 283

J

- JavaScript 341
- John *siehe* John the Ripper
- John the Ripper 26, 237, 318
- JtR *siehe* John the Ripper

K

- Kali-Boot-USB-Stick 194
- Kali Bug Tracker 32
- Kali-Build
 - anpassen 188
- Kali Evil Wireless Access Point 189
- Kali-ISO 29
- Kali-ISO-Image
 - erstellen 189
- Kali Linux
 - Anpassungsmöglichkeiten 171
 - Kali Linux Image 62
 - Kali-Linux-Image 29
 - Kali Linux ISO of Doom 189
 - Kali Live 64
 - Kali-Live-ISO-Image 188
 - Kali-Live-System 197
 - Kali-Mirror 173
 - Kali Rolling 20
 - Kali Rolling ISO of Doom, Too 189
 - kali-rolling-Tool 177
 - kali-tweaks 136
 - Kali-USB-Stick 193
 - KDE 36
 - Kennwort
 - analysieren 346
 - für den Root-Benutzer 69
 - Kennwortangriff
 - offline 168
 - online 168
 - Kernel 37, 52
 - Konfigurationsdatei 184
 - konfigurieren 184
 - Neukompilierung 182
 - Quellen 183
 - Sicherheitsupdate 182
 - Standardkonfigurationen 184
 - Kernel-Code 182
 - Kernel-Image 187
 - Kernel-Konfigurationsoberfläche 185
 - Ketten 129
 - Keylogger 228
 - Keylogging 350
 - Kimon 290
 - Kismet 26, 289
 - Kismon 290
 - Klartext-Netzwerkprotokoll 277
 - Klartextpasswort 235
 - Klonvorgang 263
 - Kommandozeile 41, 341
 - Kommandozeilenbefehl 210
 - Konfigurationsdatei 45, 269

- Konfigurationseinstellung 112
Konfigurationsparameter 188
Konfigurationsverzeichnis 190
Konsole
 virtuelle 41, 101
Kreuzkontamination 151
Kritisches System 207
- L**
- LAN Manager 236, 319
Laufzeitinformation 52
Laufzeitkonfiguration 275
Leiser Scan 254
libfreefare 172
Linux 35
Linux-Befehle 54
Linux-Derivate 98
Linux-Kernel 128
 kompilieren 181
Linux-Systemstruktur 72
Linux Unified Key Set-up 85
Live-Boot Hooks 192
Live-Build 189
live-build Skript 28
Live-CD 36
Live-Dateisystem
 Dateien hinzufügen 192
Live-System 26
LM-Passwort 319
Logical Volume Management 85
Logikbombe 350
Login-Funktion 317
Login-Shell 44
LUKS 85, 86
LUKS-Container 195
LUKS-verschlüsselte Partition 195
LVM 85
LVM-Laufwerke 89
LVM-Tool 88
LXDE 36
- M**
- MAC-Adresse
 gefälschte 240
macof 241, 277
mailsnarf 277
Maltego 26, 265
Malware 350
 aufspüren 338
Malwareuntersuchung 334
Man-in-the-Middle-Angriff 142, 277
Massenangriff 229
Master Boot Record 80
- Master-Programm 350
MBR 79
MD4 336
MD5-Hash 328
Medusa 315
Memory-Dump 340
Metadaten 261, 332
Metadateneintrag 332
MetaGooFil 211, 261
Meta-Paket 30, 191, 353
Metasploit 26, 300
 Exploits 233
 Payloads 307
 Rang 303
Metasploitable 227
Metasploit-Dokumentation 305
Meterpreter 231, 247
Mobiles Breitband 107
Mobiles Endgerät 163
mount 39
Mounten 28
msfconsole 301
msgsnarf 277
- N**
- Nacharbeiten 208
Namensauflösung 108
Namenservers 108
Nessus 228
Netcat 247
NetworkManager 106
Netzwerk 229
 ohne Internetzugang 312
 scannen 232
Netzwerkanbindung 106
Netzwerkdateisystem 39
Netzwerkdatenverkehr 240
 ausspionieren 276
Netzwerkeinstellung
 überprüfen 152
Netzwerkgestütztes Eindringen 242
Netzwerkinfrastruktur 267
Netzwerk-Intrusion 152
Netzwerkkonfiguration 67, 106
Netzwerkkontrolle 169
Netzwerkpaket 215
Netzwerkprotokoll-Analysator 280
Netzwerkrand
 Geräte 215
Netzwerkschnittstelle 107
Netzwerk-Sniffer 277
Netzwerksniffing 240
Netzwerk-Sniffing-Attacke 277

- Netzwerkverkehr
 analysieren 276
 ausspähen 240
 ausspionieren 277
 erfassen 285
 überwachen 277
 NFC-Karte 172, 179
 NFS 39
 Nikto 245, 273
 NIST-Sonderpublikation 157
 Nmap 26, 212, 214, 217, 219, 221, 223, 232, 253, 303
 Befunde 233
 Portscan 218
 Script Engine 225
 Versionsscan 223
 NOPS 301
 Normierung
 Assessments 164
 NSE 212, 214, 225
 NSE-Skript 226
 NTLM 320
 NTP-Server 70
 NULL-Scan 223, 224, 225
 NVIDIA-Grafik 99
 NVIDIA-Karte 99
- O**
- Offener Port 208
 Offensive Security 22, 30
 Office-Dokument 261
 Online-Shop 295
 Open Source 25, 36
 Open-Source-Software 36
 OpenVAS 25, 138, 212, 228, 269, 302, 346
 Open Vulnerability Assessment System 228
 OpenWRT-Router 290
 OSVDB 228
 OWASP 297
 OWASP-ZAP 263
- P**
- Package Manager 78
 Packaging-Tool 178, 179
 Paket
 ändern 177
 anpassen 171
 neu erstellen 173
 Versionsnummer 177
 Paketabhängigkeit 172
 Paketerstellungsprozess 179
 Paros 243
- Partition
 verschlüsselte 85
 Verschlüsselung 63
 Partitionierung 70
 geführte 70
 Partitionierungstool 85, 89
 Partitionsmodus
 manueller 74
 Pass the hash 235
 Passwort 318
 knacken 234
 zurücksetzen 239
 Passwort-Attacke 168
 Passwortcracker
 online 230
 Passwortcracker-Tool 321
 Passwortcracking 236
 lokal 236
 Passwort-Dump 347
 Passwörter
 decodieren 277
 Passwörter knacken
 Linux 238
 OS X 238
 Windows 236
 Passwort-Hash 230, 234
 Windows 323
 Passwort-Hash-Datei 235
 Passwort-Wörterbuch 314
 Patch 303
 Problem beheben 228
 Patch-Level 155
 Patch-Management-System 181
 Payload 229, 301, 302, 305
 PCAnywhere 230
 PCAP 290
 PCI-Gerät 52
 PCMCIA-Karte 52
 Penetrationstest 160
 Ablauf 205
 traditioneller 160
 Vier-Schritte-Prozess 205
 Penetrationstester 222
 Penetration Testing Execution Standard 209
 Permission to Attack 165
 Persistence-Start 63
 Persistenz 26, 62, 192, 193
 verschlüsselt 195
 Persistenzdateisystem 195
 Persistenzpartition
 verschlüsselt 197
 Phishing 310, 311, 350
 Web-Vorlage 311
 Phishing-Seite 311

- Phreaker 351
Physikalische Adresse 255
Physische Partition 87
PID 47
Ping 212, 215
 Hacker-Werkzeug 216
Ping-Scan 255
Pipal 346
Port 213
 Anzahl 217
 ermitteln 213
 offen 208, 212
 Verkehrsaufkommen 213
Portscan 208, 212, 217, 218, 255, 308
PostgreSQL-Cluster 115
PPPoE 107
Primäres Betriebssystem 103
Programmausführungsfluss
 steuern 167
Programmkonfiguration 36
Proof-of-Concept-Code 166
Protokoll
 verbindungsloses 221
 verbindungsorientiertes 221
Proxy 293
 konfigurieren 293
 ZAP 297
Proxy-Adresse 78
Prozess 39
 verwalten 47
Prozess-ID 47
Prozessorarchitektur 155
Prozesspriorität 40
PTA 165
PTES 209
- Q**
Quellpaket 173
 aktualisieren 180
 erstellen 181
 Quellformat 178
- R**
Race Conditions 153
RainbowCrack 26
Randgeräte 215
Raspberry Pi 95, 290
RDP 230
Recherche 209
Recon 205
Reconnaissance 205
RecordMyDesktop 347
- Recovery 336
redfang 172
Redirection 46
Regelerstellung 134
Regression 275
Remote-Codeausführung 303
Remotecomputer 305
Remote Desktop Protocol 230
Remotedienst 230
Remote-Shell 237
Remotezugriff 112
Remotezugriffsdiest 230
Report 297
Repository 32, 98
Request for Comments 223
Ressourcenverbrauch 166
Reverse-Payload 308
RFC 223
Richtlinien
 Debian 31
 Kali Linux 31
Richtlinien für Sicherheitsexperten 209
Ringbuffer 52
Risiko 151
Risikobewertung 125, 156, 159
Rolling Distribution 21
Root 39
Rootkit 247, 334, 351
Rootkit-Erkennung 334
Root-Konto 232
Root-Passwort 268
Root-Rechte 232
Router 279
RST-Paket 219
- S**
SAM 238
SAM-Datei 236, 322, 323
Samdump2 237, 322
SAM-Sperre 236
Scan
 authentifizierter 156
 automatisierter 155
 leiser 254
Scannen 161
Schnittstelle 38, 278
 dedizierte 185
Schwachstelle 151, 153, 302
 ausnutzen 229
 ermitteln 214
 scannen 246
 Webapplikationen 245

- Schwachstellenanalyse 153, 154, 160
 Tools 269
 Schwachstellenanalyse-Tools
 automatisierte 166
 Schwachstellen-Scan 156, 208, 212, 214, 221, 228
 automatisiert 299
 Ergebnisse 155
 Nikto 273
 ZAP 300
 Schwachstellen-Scanner 138, 156, 228
 Metasploit 301
 SD-Karte
 startfähig 96
 Searchsploit 312
 Secure Shell 230
 Service-Manager 119
 Service-Unit 119
 SET 172, 310, 311
 setgid 48
 SET-Power-User 180
 setuid 48
 SHA 238
 SHA-256 336
 shadow 238
 Shell 42, 43, 218
 Shell-Zugriff 150, 234
 administrativ 218
 Shrink Wrap Code 351
 Sicherheitscheck 151
 Sicherheitslücke 154, 229, 269, 351
 Sicherheitsparameter 160
 Sicherheitsprozesse 160
 Sicherheitsrichtlinien
 definieren 124
 Sicherheitsupdate
 Kernel 182
 Siege 274
 URL-Formate 275
 Signatur 154
 erstellen 155
 Signatursatz 157
 Sitemap 296
 Skipfish 296
 Skript
 ausführen 214, 226
 Slackware 19
 Sleuth Kit 329
 Sniffing 276, 280
 Sniffing Tools 156
 SNMP 222
 Social Engineering 210, 351
 Social-Engineering-Angriff 309
 Social-Engineer Toolkit (SET) 26, 310
 Software-RAID 85
 Softwareversion 155
 Source-Paket 98
 Soziale Dienste 266
 Spam 351
 Speicherbeschädigung 167
 Speicher-Dumb 339
 Speicherforensik 338, 339
 Speichermedium
 bootfähiges 64
 Speicherverbrauch 182
 Spider 299
 automatisiert 244
 Spiderangriff 294
 ZAP 299
 Spoofing 276, 351
 Spracheinstellung 65
 Spyware 351
 SQL-Befehle 153
 SQL-Injection 150, 153, 168, 244, 351
 SSH 112, 230
 SSH-Host-Schlüssel 113
 sshmitm 277
 SSID 285
 SSLstrip 142
 SSL-Zertifikat 140
 Stable Distribution 20
 Stack Buffer Overflow 167
 Standard-Angriffsziel 163
 Standard-Assessment 162
 Standardkonfiguration 191
 optimieren 171
 Standard-Linux-Kernel 67
 Standardnetzwerkkonfiguration 106
 Standardportnummer 213
 Standardports 220
 Standard-Shell 110
 Startmedium 199
 Statistiken 346
 Subdomänen
 aufspüren 258
 Subnetz 220
 Superuser-Root-Konto 69
 SWAP-Partition 28, 77, 89, 152
 Switch 240, 279
 SYN/ACK 218
 SYN-Flag 225
 SYN-Scan 218, 255
 starten 219
 syskey 322
 SYSTEM 238
 Systemd 52

Systeme

- kritische 207
- Systemressource 47
- Systemsicherung 327
- SysVinit-Methode 19

T

- Target-Unit 119
- Tarnung 219
- Tastaturlayout 66
- TCP 221
- TCP-Port 217
- TCP-RFC 224
- TCP-Stack 256
- TCP-Verbindung 114
- TCP-Verbindungsscan 219, 220, 221
- Telnet 230
- Terminal 41
- Texteditor 46
- TFTP 222
- TheHarvester 211, 258
- Threat 351
- Threats pro Scan 156
- Tool

- Dsniff 276
- Exploitation 300
- Man-in-the-Middle-Angriffe 277
- Penetrationstest 291
- Schwachstellenanalyse 269
- Sniffing 276
- Spoofing 276

Tools

- automatisierter 157
- für Angriffe 283
- zur Informationssammlung 253

Torrent 59

- Traditioneller Penetrationstest 153, 160
- Transaktionsinformationen 275
- Trojaner 351
- True Negative 155
- True Positive 155

U

- Überwachungsdienst 152
- Ubuntu 19
- UDP 221
- UDP-Port 217, 221
- UDP-Scan 221, 222
- UEFI 37
- UID-Variable 111
- Umgebungsvariable 44, 45
- Unified Sniffing 279
- Unix 38, 48

Unix-basiertes Betriebssystem 61

- Unix-Crypt(3)-Hash 318
- Unix-Derivate 98
- Upstream 178
- Upstream-Git-Repository 178
- Upstream-Version 171
 - packen 180
- urlsnarf 277
- USB-Gerät 52
- User-Account 110
- User-Agent 343
- User-Space 53
- User-Space-Bibliothek 188

V

- Validierungsprozess
 - Tools 165
- Variable 44
- Verbindungsaufbau 218
- Verbindungsloses Protokoll 221
- Verbindungsorientiertes Protokoll 221
- Verfügbarkeit 149
- Verschleierung 219
- Verschlüsselte Partition 85
- Verschlüsselter Hash 235
- Verschlüsselung 322
- Verschlüsselungs-Passphrase 86
- Verschlüsselungsschlüssel 86
- Vertraulichkeit 149
- Verzeichnis 39
- Verzeichnisbaum 42
- VFAT 39
- VirtualBox 23
- Virtual Network Computing 230
- Virtuelle Konsole 41, 101
- Virtueller Host 116
- Virtuelles Dateisystem 52
- Virus 352
- VMware 23
- VNC 305
- VNC-Injektion 307
- VNC-Payload 237
- Volafox 338
- Volatility 339
- Volume-Gruppe 85
- Voreinstellungsdatei 199
 - erstellen 200
 - initrd 199
 - Netzwerk 200
 - Startmedium 199
- VPN 107
- VPN-Netzwerk 272
- Vulnerability 153

Vulnerability Analysis 154

Vulnerability-Scanner 269

W

w3af 243

Webanwendung 154, 162, 296

Webanwendungs-Assessment 152

Webapplication 154

Web Application Audit und Attack Framework 243

Webapplikation

Schwachstellen 245

Webframework 296

Webgestützter Angriff 242

Webhacking 243, 245

Webkit-Rendering 341

webmitm 277

Web-Penetrationstest 298, 343

Webpräsenz

Unternehmen 242

Webscanner 246

WebScarab 245, 291

Web-Schwachstelle 167

Webseite 263

analysieren 245

Offline-Kopie 263

Webserver 116, 273

Informationen gewinnen 217

webspy 277

WEP 283

WEP-Schlüssel 283, 287

knacken 284

White-Box-Assessment 163

Windows-Eingabeaufforderung 231

Windows-Installation 81

Windows-LM-Passwörter 236

Windows-NT-basierte Systeme 323

Windows-Partition 74

verkleinern 82

Win-KeX 94

Wireless-Assessments 152

Wireless Injection 19

Wireless Security Assessment 152

Wireless Wide Area Network 107

Wireshark 26, 242, 280

WLAN-Hacking 283

WLAN-Netzwerk

aufspüren 290

Worst-Case-Szenario 161

WSL-Distribution 91

Wurm 352

WWAN 107

X

XFCE 36

Xmas-Scan 223, 224

XSS 244, 352

XSS-Angriff 168

Z

ZAP 297

Zed Attack Proxy 297

Zeitbombe 350

Zenmap 217, 253

Zero-Day-Exploit 269

Zielnetzwerk 154

Zielorganisation 211

Ziel-PC

steuern per Kommandozeile 218

Zombie-Drohne 352

Zonentransfer 260

ZSH-Terminal 344

Zugangspunkt 213

Zugriff

administrativer 229

festigen 208

Zugriffsbeschränkung 119