



Jürgen
Ebner

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	23
1.4.1	Live-System	25
1.4.2	Ein maßgeschneiderter Linux-Kernel	27
1.4.3	Komplett anpassbar	27
1.4.4	Ein vertrauenswürdiges Betriebssystem	29
1.4.5	Auf einer großen Anzahl von ARM-Geräten verwendbar	29
1.5	Richtlinien von Kali Linux	30
1.5.1	Ein einzelner Root-Benutzer als Standard	30
1.5.2	Netzwerkdienste sind standardmäßig deaktiviert	30
1.5.3	Eine organisierte Sammlung von Tools	30
1.6	Zusammenfassung	31
2	Linux-Grundlagen	33
2.1	Was ist Linux und wie funktioniert es?	33
2.1.1	Hardwaresteuerung	35
2.1.2	Vereinheitlichtes Dateisystem	36
2.1.3	Prozesse verwalten	37
2.1.4	Rechtmanagement	38
2.2	Die Kommandozeile (Command Line)	39
2.2.1	Wie komme ich zur Kommandozeile?	39
2.2.2	Verzeichnisbaum durchsuchen und Dateien verwalten	40

2.3	Das Dateisystem	42
2.3.1	Dateisystem-Hierarchie-Standard	42
2.3.2	Das Home-Verzeichnis des Anwenders	43
2.4	Hilfreiche Befehle	44
2.4.1	Anzeigen und Ändern von Text-Dateien	44
2.4.2	Suche nach Dateien und innerhalb von Dateien	44
2.4.3	Prozesse verwalten	45
2.4.4	Rechte verwalten	45
2.4.5	Systeminformationen und Logs aufrufen	49
2.4.6	Hardware erkennen	50
2.5	Zusammenfassung	51
3	Installation von Kali	55
3.1	Systemanforderungen	55
3.2	Erstellen eines bootfähigen Mediums	56
3.2.1	Herunterladen des ISO-Images	56
3.2.2	Kopieren des Images auf ein bootfähiges Medium	57
3.2.3	Aktivieren der Persistenz auf dem USB-Stick	60
3.3	Stand-Alone-Installation	62
3.3.1	Partitionierung der Festplatte	68
3.3.2	Konfigurieren des Package Managers (apt)	75
3.3.3	GRUB-Bootloader installieren	76
3.3.4	Installation abschließen und neu starten	79
3.4	Dual-Boot – Kali Linux und Windows	79
3.5	Installation auf einem vollständig verschlüsselten Dateisystem	82
3.5.1	Einführung in LVM	82
3.5.2	Einführung in LUKS	83
3.5.3	Konfigurieren verschlüsselter Partitionen	83
3.6	Kali Linux auf Windows Subsystem for Linux	88
3.7	Kali Linux auf einem Raspberry Pi	91
3.8	Systemeinstellungen und Updates	94
3.8.1	Repositories	94
3.8.2	NVIDIA-Treiber für Kali Linux installieren	95
3.8.3	Terminal als Short-Cut (Tastenkombination)	98
3.9	Fehlerbehebung bei der Installation	99
3.9.1	Einsatz der Installer-Shell zur Fehlerbehebung	100
3.10	Zusammenfassung	101

4	Erste Schritte mit Kali	103
4.1	Konfiguration von Kali Linux	103
4.1.1	Netzwerkeinstellungen	104
4.1.2	Verwalten von Benutzern und Gruppen	107
4.1.3	Services konfigurieren	109
4.2	Managing Services.	117
4.3	Hacking-Labor einrichten	119
4.4	Sichern und Überwachen mit Kali Linux	121
4.4.1	Sicherheitsrichtlinien definieren.	122
4.4.2	Mögliche Sicherheitsmaßnahmen	124
4.4.3	Netzwerksservices absichern.	125
4.4.4	Firewall- oder Paketfilterung	126
4.5	Weitere Tools installieren	134
4.5.1	Terminator statt Terminal	134
4.5.2	OpenVAS zur Schwachstellenanalyse.	135
4.5.3	SSLstrip2.	138
4.5.4	Dns2proxy.	139
4.6	Kali Linux ausschalten.	139
4.7	Zusammenfassung	140

Teil II Einführung in Penetration Testing **143**

5	Einführung in Security Assessments	145
5.1	Kali Linux in einem Assessment	147
5.2	Arten von Assessments	148
5.2.1	Schwachstellenanalyse	150
5.2.2	Compliance-Test.	155
5.2.3	Traditioneller Penetrationstest	156
5.2.4	Applikations-Assessment.	158
5.3	Normierung der Assessments	160
5.4	Arten von Attacks	161
5.4.1	Denial of Services (DoS)	162
5.4.2	Speicherbeschädigungen	163
5.4.3	Schwachstellen von Webseiten	163
5.4.4	Passwort-Attacks	164
5.4.5	Clientseitige Angriffe	165
5.5	Zusammenfassung	165

6	Kali Linux für Security Assessments vorbereiten	167
6.1	Kali-Pakete anpassen	167
6.1.1	Quellen finden	169
6.1.2	Build-Abhängigkeiten installieren	172
6.1.3	Änderungen durchführen	173
6.1.4	Build erstellen	177
6.2	Linux-Kernel kompilieren	177
6.2.1	Einführung und Voraussetzungen	178
6.2.2	Quellen finden	179
6.2.3	Kernel konfigurieren	180
6.2.4	Pakete kompilieren und erstellen	182
6.3	Erstellen eines individuellen Kali-Live-ISO-Images	183
6.3.1	Voraussetzungen	184
6.3.2	Erstellen von Live-Images mit verschiedenen Desktop-Umgebungen	185
6.3.3	Ändern der Liste installierter Pakete	186
6.3.4	Verwenden von Hooks zum Optimieren des Live-Images	187
6.3.5	Hinzufügen von Dateien zum ISO-Image oder Live-Filesystem	187
6.4	Hinzufügen von Persistenz auf einem USB-Stick	188
6.4.1	Erstellen einer unverschlüsselten Persistenz auf einem USB-Stick	189
6.4.2	Erstellen einer verschlüsselten Persistenz auf einem USB-Stick	190
6.4.3	Verwenden von mehreren Persistenzspeichern	191
6.5	»Automatisierte« Installation	193
6.5.1	Antworten auf Installationsabfragen vorbereiten	193
6.5.2	Erstellen der Voreinstellungsdatei	195
6.6	Zusammenfassung	195
6.6.1	Kali-Pakete ändern	196
6.6.2	Linux-Kernel neu kompilieren	197
6.6.3	Benutzerdefinierte ISO-Images erstellen	198
7	Ablauf eines Penetrationstests	201
7.1	Informationen sammeln	205
7.1.1	Was nun?	205
7.1.2	Kali-Tools zur Informationsbeschaffung	207
7.1.3	Informationen nach angreifbaren Zielen durchsuchen	207

7.2	Scannen	208
7.2.1	Pings	211
7.2.2	Portscan.	213
7.2.3	Nmap Script Engine – Transformationen eines Tools	221
7.2.4	Schwachstellen-Scan	224
7.3	Eindringen über das lokale Netzwerk	225
7.3.1	Zugriff auf Remotedienste.	226
7.3.2	Übernahme von Systemen	227
7.3.3	Passwörter hacken	230
7.3.4	Abrissbirnen-Technik – Passwörter zurücksetzen	235
7.3.5	Netzwerkverkehr ausspähen	236
7.4	Webgestütztes Eindringen	238
7.4.1	Schwachstellen in Webapplikationen finden	241
7.4.2	Webseite analysieren	241
7.4.3	Informationen abfangen	241
7.4.4	Auf Schwachstellen scannen	242
7.5	Nachbearbeitung und Erhaltung des Zugriffs.	242
7.6	Abschluss eines Penetrationstests	244
7.7	Zusammenfassung	245

Teil III Tools in Kali Linux 247

8	Tools zur Informationsbeschaffung und Schwachstellenanalyse ...	249
8.1	Tools zur Informationssammlung	249
8.1.1	Nmap – Das Schweizer Taschenmesser für Portscanning.	249
8.1.2	TheHarvester – E-Mail-Adressen aufspüren und ausnutzen	254
8.1.3	Dig – DNS-Informationen abrufen.	256
8.1.4	Fierce – falls der Zonentransfer nicht möglich ist.	256
8.1.5	MetaGooFil – Metadaten extrahieren	257
8.1.6	HTTrack – Webseite als Offline-Kopie	258
8.1.7	Maltego – gesammelte Daten in Beziehung setzen.	260
8.1.8	Sparta – Automation in der Informationsbeschaffung	262
8.2	Schwachstellenanalyse-Tools	264
8.2.1	OpenVAS – Sicherheitslücken aufdecken	264
8.2.2	Nikto – Aufspüren von Schwachstellen auf Webservern ...	268
8.2.3	Siege – Performance Test von Webseiten	270

8.3	Sniffing und Spoofing	271
8.3.1	Dsniff – Sammlung von Werkzeugen zum Ausspionieren von Netzwerkdatenverkehr	272
8.3.2	Ettcap – Netzwerkverkehr ausspionieren	273
8.3.3	Wireshark – der Hai im Datenmeer	275
9	Tools für Attacken	279
9.1	Wireless-Attacken	279
9.1.1	aircrack-ng	279
9.1.2	Ghost Phisher	283
9.1.3	Kismet	284
9.2	Webseiten-Penetration-Testing	286
9.2.1	WebScarab	286
9.2.2	Skipfish	291
9.2.3	Zed Attack Proxy	292
9.3	Exploitation-Tools	295
9.3.1	Metasploit	295
9.3.2	Armitage	303
9.3.3	Social Engineer Toolkit (SET)	304
9.3.4	Searchsploit	307
9.4	Passwort-Angriffe	309
9.4.1	Medusa	310
9.4.2	Hydra	312
9.4.3	John the Ripper	313
9.4.4	Samdump2	317
9.4.5	chntpw	318
10	Forensik-Tools	321
10.1	Dcfldd – Abbild für forensische Untersuchung erstellen	321
10.2	Autopsy	323
10.3	Binwalk	326
10.4	Chkrootkit	328
10.5	Bulk_extractor	328
10.6	Foremost	329
10.7	Galleta	330
10.8	Hashdeep	330
10.9	Volafox	332
10.10	Volatility	333

11	Tools für Reports	335
11.1	Cutycapt	335
11.2	Faraday-IDE	337
11.3	Pipal	341
11.4	RecordMyDesktop	341
A	Terminologie und Glossar	343
B	Übersicht Kali-Meta-Pakete	347
B.1	kali-linux.	347
B.2	kali-linux-full	347
B.3	kali-linux all	348
B.4	kali-linux-top10	348
B.5	kali-linux-forensic	348
B.6	kali-linux-gpu	349
B.7	kali-linux-pwtools.	349
B.8	kali-linux-rfid	349
B.9	kali-linux-sdr	349
B.10	kali-linux-voip.	349
B.11	kali-linux-web	350
B.12	kali-linux-wireless	350
C	Checkliste: Penetrationstest	351
C.1	Scope	351
C.2	Expertise	353
C.3	Lösung	353
D	Installation von Xfce und Undercover-Modus	355
	Stichwortverzeichnis	359



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 wichtiger wiegt für viele die Tatsache, dass die meisten dieser Tools kostenlos erhältlich sind.

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.

Vorsicht

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

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.

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.

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 empfeh-

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

len, 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.

Teil I

Grundlagen von Kali Linux

Bevor Sie sich mit den Tools von Kali Linux und deren Einsatz beschäftigen, ist es wichtig, dass Sie verstehen, warum es dieses System gibt und was bei der Entwicklung eines Hacker-Betriebssystems bedacht wurde. Aus diesem Grund beschäftigen wir uns am Beginn des ersten Teils von Kali Linux mit der Geschichte von Kali und wie es sich von Debian unterscheidet.

Da es mehrere Versionen von Kali Linux gibt, damit es auch auf unterschiedlichen Plattformen genutzt werden kann, stelle ich Ihnen hier auch die unterschiedlichen Versionen kurz vor.

Für den Fall, dass Sie noch keine Erfahrungen mit Linux haben, habe ich auch die wichtigsten Grundlagen von Linux angeführt, die Ihnen aber auch als Auffrischung dienen können. In diesem Zusammenhang zeige ich Ihnen auch, wie Sie Kali Linux installieren und an Ihre Bedürfnisse anpassen. Anschließend ist das System bereit, damit Sie Ihren ersten Penetrationstest durchführen können.

In diesem Teil:

- **Kapitel 1**
Einführung19
- **Kapitel 2**
Linux-Grundlagen33
- **Kapitel 3**
Installation von Kali55
- **Kapitel 4**
Erste Schritte mit Kali103

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@ictekalı:/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@ictekalı:/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 Desktopsitzung 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.

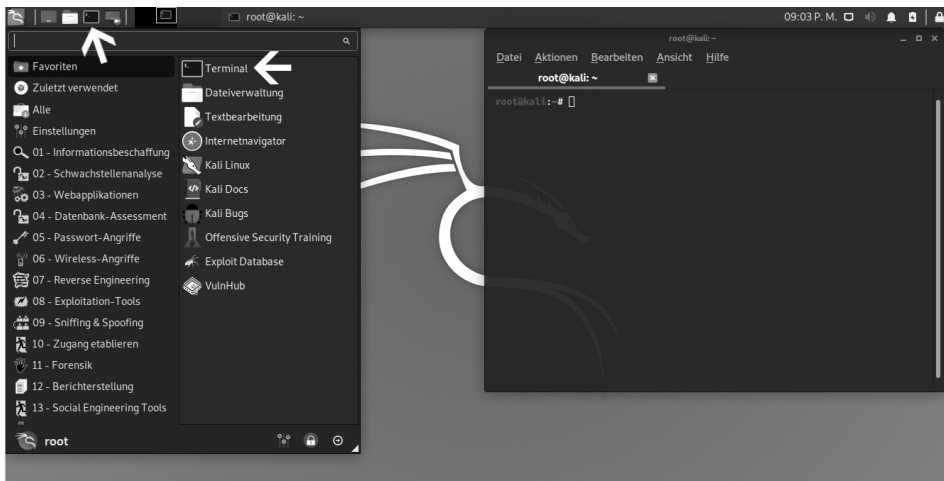


Abb. 2.2: Terminal aufrufen

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

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* *Zielfilei*.

```
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`, `/usr/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/**: Einhängpunkt für entfernbare Geräte – CD-ROM, USB-Stick usw.
- */mnt/*: vorübergehender Einhängpunkt
- */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 Unterzeichnissen) 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.

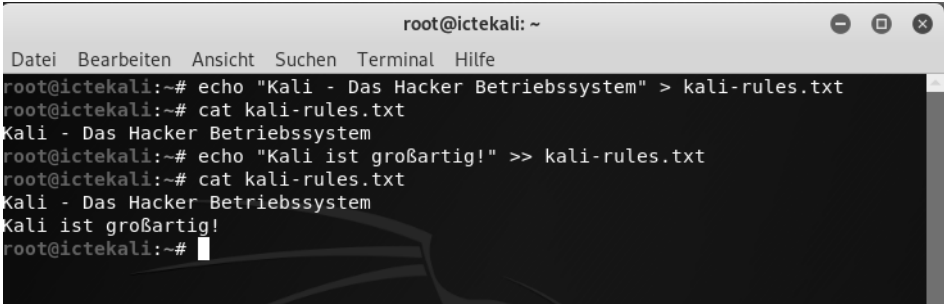
Grafische Desktops verfügen normalerweise über Verknüpfungen, mit denen Inhalte des Verzeichnisses `~/Desktop/` angezeigt werden können (oder auch entsprechende Übersetzungen für Systeme, die nicht auf Englisch konfiguriert sind).

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

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

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 mit `bg% job-number` neu gestartet werden.

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 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/sda1      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
- `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-get 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 Kern-dienstprogramme 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
 - `lspcmcia` – 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 36
- Abuse-Meldung 267
- Access Point 280
- ACK-Paket 214, 215
- address resolution protocol 250
- Address Space Layout Randomization 150
- Administrationsrecht 30
- Administrativer Zugriff 204
- Administrativer Zugriff 225
- Administratorkonto 320
 - knacken 230
- Administratorpasswort
 - zurücksetzen 318
- Adresse
 - physische 251
- Adware 343
- aircrack-ng 25, 279
- Aktive Informationsbeschaffung. 250
- amd64-Plattformen 55
- Analysieren
 - von Kennwörtern 272
- Android-Exploit 308
- Anforderungen
 - behördliche 155
 - branchenspezifische 155
- Angriff 343
 - clientseitiger 165
 - webgestützter 238
- Angriffserkennungssystem 250
- Anmeldeinformationen 273
- Ansatz
 - hybrider 160
- Anti-Exploit-Technologie 150
- Anwenderaktualisierung 165
- Anwendungs-Assessment 160
- Anwendungsdatei 43
- Anwendungskonfigurationsdatei 43
- Anwendungsverhalten 159
- Apache-Konfigurationsanweisungen 116
- Apache-Prozess 114
- Apache-Standardmodule 114
- Apache-Webserver
 - konfigurieren 113

- Applikations-Assessment 149, 158
- APT 172
- Arbeitsspeicher 83
- ARM-Computer 91
- Armel-Plattformen 55
- Armhf-Plattformen 55
- Armitage 228, 229, 303
- ARM-Plattformen 55
- ARP 250
- ARP-Cache 251
- ARPreplay-Attacke 282
- ARP-Request 282
- arpspoof 272
- ARP-Spoofing 275
- Assessment
 - Arten von 148
 - Installation 147
- Assessment-Plattform 152
- Aufklärung 201, 206, 255
- Aufklärungsphase 203
- Auslagerungsdatei 83
- Auslagerungspartition 83
- Auswirkungen 154
- Authentifizierter Scan 152
- Authentifizierung
 - Access Point 283
 - Basic 116
- Authentifizierungsebene 239
- Automatisierte Installation 147
- Automatisierte Tools 153
- Automatisierter Scan 151
- Automatisierung 213
- Autopsy 323
 - Analyse 325
- Availability 145

B

- Backdoor 343
- Back-End-Seitengenerierungslogik 163
- BackTrack 19
- Banner 222
- Base64-Codierung 291
- Bash 40
- Bedrohung 147

Bedrohungsstufe 268
 Befehle
 Übersicht 52
 Befehlsinterpretierer 45
 Befehlszeile *siehe* Kommandozeile
 Befehlszeileninterpretierer 40
 Befehlszeilenwerkzeugen 89
 Belastungstest 271
 Benchmarking 270
 Benutzerkennwort 67
 Berechtigungssystem 45
 Bereitstellungspunkt 61, 317
 Bericht
 erstellen 158, 204
 Berichterstattung 204
 Betriebssystemversion 150
 Bettercap 24
 BID-Nummer 298
 Bildanalyse 329
 Binärer Hook 187
 Binär-Image 326
 Binärpaket 177
 Bind-Payload 302
 Binwalk 326
 BIOS 35
 Black-Box-Assessment 159
 Bootfähiges Speichermedium 62
 Bootkey 317
 Bootloader 35, 76, 178
 Bootloader-Konfiguration
 ändern 193
 Boot-Parameter 193, 195
 Bot 343
 Botnet 343
 Breitband
 mobiles 104
 Bridged Sniffing 274
 Broadcast 222, 236
 Brute Force 270
 Brute-Force-Anmeldetool 310
 Brute-Force-Attacke 226, 313, 343
 Brute-Force-Methode 309, 311
 BSI 224
 BSSID 282
 Buffer-Overflow 149, 163, 343
 Bugtraq ID Database 298
 Build-Abhängigkeiten installieren 172
 Build-Environment 175
 Build-Essential-Paket 178
 Build-Option 175
 Build-Prozess 177
 Bulk_extractor 328
 Burp Suite 239

C

Caching-Proxy 76
 CANVAS 295
 Capture-Filter 277
 CentOS 35
 chntpw 318
 Chromebook 57
 Chroot Hooks 187
 chrootkit 328
 Chroot-Umgebung 187
 CIA-Triade 145
 Clientseitiger Angriff 165
 Clone Phishing 343
 Closed-Source-Datei 27
 Cloud-Dienstanbieter 161
 Cloud-Installation 23
 Cloud-Service 161
 Codeausführung 224
 Code-Execution-Exploit 162
 Common Vulnerabilities Exposures 298
 Compliance 155
 Compliance-Framework 156
 Compliance-Test 149, 155, 156
 Confidentiality 145
 Connect-Scan 251
 Cookies 330
 CORE Impact 295
 Cracker 343
 Cracks pro Sekunde
 messen 314
 Crawler 260
 Crawling 241
 Cross Site Scripting (XSS) 164, 240, 346
 Cryptcat 244
 Cutycapt 335
 CVE 223
 CVE-Nummer 152, 298
 CVSS-Score 153
 Cyber-Hygiene 132
 Cyberuntersuchung 329

D

Daemon-Daten 71
 Daemons 109
 Data Execution Prevention 150
 Dateisystem 36, 37
 virtuelles 50
 Dateisystemformat 37
 Datenbankserver
 PostgreSQL 111
 Datenintegrität 322

- Datenpaket
 - suchen 277
 - WLAN 280
- Datenstruktur
 - wiederherstellen 329
- Datenverkehr 211, 237
- Dcfldd 321
- DDoS 344
- debconf-Datenbank 100
- Debconf-Voreinstellungen 193
- DEBEMAIL 173
- DEBFULLNAME 173
- Debian 19, 35
- Debian Unstable 27
- Debian-Kernel-Handbuch 178
- Debian-Kernel-Paket 178
- Debian-Live-Systemhandbuch 185
- Debian-Packaging 173
- Debian-Paket 178, 182
- Debian-Quellverwaltungs-Datei 169
- Debian-Richtlinien 30
- Debonf-Fragen 193
- Debugging-Symbol 183
- Debug-Meldung 168
- Dedizierte Gruppe 47
- Dedizierte Schnittstelle 180
- Default Desktop 79
- Default Gateway 105
- Denial of Service 162, 344
- Denial-of-Service-Angriff 237
- Denial-of-Service-Bedingung 162
- Desktop-Anwendungen 158
- Desktopsitzung 39
- Desktop-Umgebung 28, 185
- Device-Mapper 83
- dget-Quellpaket 171
- DHCP 106, 218
- DHCP-Einstellungen 148
- Dienst
 - aktiver 208
- Dig 207, 256
- Digitaler Fingerabdruck 259
- Display-Filter 277
- Distribution 19, 33
- DMZ 250
- DNS 218
- Dns2proxy 139
- DNS-Abfrage 256
- DNS-Server 105, 206
- dnsspoof 272
- Domain Controller 341
- Domänenadministratorkonto 230
- DoS 146, 162, 344

- DoS-Angriff 162
- DoS-Ergebnis 308
- dpkg-Dateien 168
- Drei-Wege-Handshake 214, 217
- Drohne 285
- Dsniff 237, 272
- dsniff 272

E

- EDB-ID 152
- Eindringen 158, 204
 - netzwerkgestütztes 238
- Eingangsbuffer 129
- Einstellungs-Reiter 288
- Eintrittswahrscheinlichkeit 153
- E-Mail-Adressen
 - aufspüren 254
- E-Mail-Passwort 273
- Embedded Device 57
- Encoder 296
- Endgeräte
 - mobiles 159
- Enlightment 34
- Ermittler
 - forensischer 321
- Erstellungszeitpunkt 175
- Ethernet-Netzwerk 272
- Ethischer Hacker 204
- Ettercap 273
 - Sniff-Modi 274
- Exploit 149, 225, 243, 297, 344
 - Definition 149
- Exploit Kit 344
- Exploitation-Tools 295
- Exploit-Code 162
- Exploit-Datenbank 307
- Exploit-DB-Package 307
- Exploit-Framework 295
- Exploit-Writer 162
- ext3-Filesystem 60
- ext4-Dateisystem 190

F

- Fail Open 237
- Fail-Open-Modell 237
- Fake-Access-Point 284
- False Negative 151
- False Positive 151
- Faraday 337, 339
- Fedora-Linux 35
- Fehlerbericht 101
- Fehlkonfigurationen 264
- FHS 24, 42

Fierce 207, 256
 File Inclusion 149
 File Transfer Protocol 226
 filesnarf 272
 Filesystem Hierarchy Standard *siehe* FHS
 Filternetz-Gateway 126
 Fingerabdruck
 digitaler 259
 Firewall 218, 221, 250, 344
 Firewall-Log 251
 Firmware 326
 Firmware-Datei 183
 Firmware-Image
 analysieren und extrahieren 326
 Foremost 329
 Forensik 27
 Image erstellen 321
 Forensik-Modus 25, 26
 Forensik-Tools 148
 Forensischer Ermittler 321
 Format String 163
 FPING 212
 FQDN 106
 FTP 226
 FTP-Datenverbindung 131
 FTP-Protokoll 131
 FTP-Server 238

G

Galleta 330
 Garbage-String 328
 Genehmigungsprozess 160
 Gerichtsverfahren 324
 Gesamtrisiko 154
 Ghost Phisher 283
 GID-Variable 109
 Git 168
 Git-Workflows 174
 GNOME3 34
 GNOME-Desktop-Umgebung 58
 GNOME-Shell 20
 GnuPG-Schlüssel 177
 Google Direktiven 206
 GParted 80
 GPS 285
 GPU 164
 Grafikprozessor 164
 GRUB 77
 GRUB-Bootmenü 81
 GRUB-Konfiguration 77
 Gruppe
 dedizierte 47
 Gruppenvariable 109

H

Hacker
 ethischer 204
 Hacker-Befehlsshell 228
 Hacking 225
 Hacking-Labor 119
 Hail-Mary-Funktion 303
 Hardwareerkennung 64
 Hardwarekonfiguration 177
 Hash 322
 verschlüsselter 231
 Hash-Algorithmus 231
 Microsoft 314
 Hashdeep 330
 Hash-Wert 231, 330
 Header-Datei 182
 Heap Corruption 163
 Heap-Speicher-Pointer 163
 Heimlicher Scan 215
 Herstellerhinweise 153
 Heuristik 291
 Hierarchie 44
 Hintertür 224
 Hintertürzugriff 214
 Home-Verzeichnis 43
 Hook
 binärer 187
 Hooks 187
 Hop 279
 Host 206
 virtueller 114
 Host-Betriebssystem
 Shell-Zugriff 146
 Hosterkennung 215, 216
 HTTP-Anforderungen 336
 abfangen 289
 anpassen 292
 HTTP-Proxy 288
 HTTP-Regression 270
 HTTPS-Regression 270
 https-Verbindung
 protokollieren 286
 http-Verbindung
 protokollieren 286
 HTTrack 207, 258
 Hub 236
 Hybrider Ansatz 160
 Hydra 312

I

i386-Plattformen 55
 ICMP 129, 211, 252
 Identitätsuntersuchung 328

- Identität
 - verschleiern 271
- IDS 250
- Image
 - forensisches 321
 - Hash-Wert 324
- Information Gathering 201
- Informationen
 - sammeln 205
- Informationsbeschaffung 157, 201, 202, 203, 205, 249
 - aktive 250
 - automatisierte Werkzeuge 206
- Informationsquellen
 - mehrere 161
- Informationssicherheit 156
- Initialisierungsvektor 279
- initrd-Generator 178
- Installation
 - Fehlerbehebung 99, 101
 - Voraussetzungen 101
- Installationsprotokoll 101
- Integer Overflow 163
- Integrated Penetration-Test Environment 337
- Integrität 145, 322
- Integrity 145
- Internet Control Message Protocol 129
- Internetsimulation 270
- Intrusion-Detection-System 215, 250
- Intrusionuntersuchung 328
- IP-Adressbereich 228
- IP-Adresse 105, 148, 207, 211, 297
- IP-Adressraum 252
- IPE 337
- IRC-Client 227
- IRC-Programm 227
- ISO 28
- ISO-Image 34, 56
 - Dateien hinzufügen 187
 - herunterladen 56
- IV 279
- J**
 - JavaScript 336
 - John *siehe* John the Ripper
 - John the Ripper 25, 233, 313
 - JtR *siehe* John the Ripper
- K**
 - Kali Bug Tracker 31
 - Kali e17 22
 - Kali Evil Wireless Access Point 184
 - Kali Linux
 - Anpassungsmöglichkeiten 167
 - Kali Linux Image 60
 - Kali Linux ISO of Doom 184
 - Kali Live 62
 - Kali Mate 22
 - Kali Rolling 20
 - Kali Rolling ISO of Doom, Too 184
 - Kali-Boot-USB-Stick 189
 - Kali-Build
 - anpassen 183
 - Kali-ISO 28
 - Kali-ISO-Image
 - erstellen 184
 - Kali-Linux-Image 28
 - Kali-Live-ISO-Image 183
 - Kali-Live-System 191
 - Kali-Mirror 169
 - kali-rolling-Tool 172
 - Kali-USB-Stick 188
 - KDE 34
 - Kennwort
 - analysieren 341
 - für den Root-Benutzer 67
 - Kennwortangriff
 - offline 164
 - online 164
 - Kernel 35, 50
 - Konfigurationsdatei 180
 - konfigurieren 180
 - Neukompilierung 178
 - Quellen 179
 - Sicherheitsupdate 178
 - Standardkonfigurationen 180
 - Kernel-Code 177
 - Kernel-Image 182
 - Kernel-Konfigurationsoberfläche 180
 - Ketten 127
 - Keylogger 224
 - Keylogging 344
 - Kimon 285
 - Kismet 24, 284
 - Kismon 285
 - Klartext-Netzwerkprotokoll 272
 - Klartextpasswort 231
 - Klonvorgang 259
 - Kommandozeile 39, 335
 - Kommandozeilenbefehl 206
 - Konfigurationsdatei 43, 264
 - Konfigurationseinstellung 110
 - Konfigurationsparameter 183
 - Konfigurationsverzeichnis 185

Konsole
 virtuelle 39, 99
 Kontrollzentrum 104
 Kreuzkontamination 147
 Kritisches System 203

L

LAN Manager 232, 314
 Laufzeitinformation 50
 Laufzeitkonfiguration 270
 Leiser Scan 250
 libfreefare 168
 Linux 33
 Linux Unified Key Set-up 82
 Linux-Befehle 52
 Linux-Derivate 94
 Linux-Kernel 126
 kompilieren 177
 Linux-Systemstruktur 70
 Live-Boot Hooks 187
 Live-Build 184
 live-build Skript 27
 Live-CD 34
 Live-Dateisystem
 Dateien hinzufügen 187
 Live-System 25
 LM-Passwort 314
 Logical Volume Management 82
 Logikbombe 344
 Log-in-Bildschirm 97
 Login-Funktion 312
 Login-Shell 42
 LUKS 82, 83
 LUKS-Container 190
 LUKS-verschlüsselte Partition 190
 LVM 82
 LVM-Laufwerke 86
 LVM-Tool 86
 LXDE 34

M

MAC-Adresse
 gefälschte 236
 macof 237, 272
 mailsnarf 272
 Maltego 24, 260
 Malware 344
 aufspüren 332
 Malwareuntersuchung 328
 Man-in-the-Middle-Angriff 139, 273, 283
 Massenangriff 225
 Master Boot Record 78

Master-Programm 344
 MBR 77
 MD4 330
 MD5-Hash 322
 Medusa 310
 Memory-Dump 334
 Metadaten 257, 326
 Metadateneintrag 326
 MetaGooFil 207, 257
 Meta-Paket 29, 186, 347
 Metasploit 25, 295
 Exploits 229
 Payloads 302
 Rang 298
 Metasploitable 223
 Metasploit-Dokumentation 300
 Meterpreter 227, 243
 Mobiles Breitband 104
 Mobiles Endgerät 159
 mount 37
 Mounten 26
 msfconsole 296
 msgsnarf 272

N

Nacharbeiten 204
 Namensauflösung 106
 Namensservers 106
 Nessus 224
 Netcat 244
 NetworkManager 104
 Netzwerk 225
 ohne Internetzugang 307
 scannen 228
 Netzwerkanbindung 104
 Netzwerkdateisystem 37
 Netzwerkdatenverkehr 236
 ausspionieren 272
 Netzwerkeinstellung
 überprüfen 148
 Netzwerkgestütztes Eindringen 238
 Netzwerkinfrastruktur 262
 Netzwerk-Intrusion 148
 Netzwerkkonfiguration 65, 104
 Netzwerkkontrolle 165
 Netzwerkpaket 211
 Netzwerkprotokoll-Analysator 275
 Netzwerkrand
 Geräte 211
 Netzwerkschnittstelle 105
 Netzwerk-Sniffer 272
 Netzwerksniffing 236
 Netzwerk-Sniffing-Attacke 273

- Netzwerkverkehr
 - analysieren 272
 - ausspähen 236
 - ausspionieren 273
 - erfassen 282
 - überwachen 273
- NFC-Karte 168, 176
- NFS 37
- Nikto 241, 268
- NIST-Sonderpublikation 153
- Nmap 24, 208, 210, 213, 215, 217, 219, 228, 249, 297
 - Befunde 229
 - Portscan 214
 - Script Engine 221
 - Versionsscan 219
- NOPS 296
- Normierung
 - Assessments 160
- NSE 208, 210, 221
- NSE-Skript 222
- NTLM 315
- NTP-Server 68
- NULL-Scan 219, 220, 221
- NVIDIA-Grafik 95
- NVIDIA-Karte 96
- NVIDIA-Treiber 96
- O**
 - Offener Port 204
 - Offensive Security 21, 29
 - Office-Dokument 257
 - Online-Shop 290
 - Open Source 24
 - Open Vulnerability Assessment System 224
 - Open-Source 34
 - Open-Source-Software 34
 - OpenVAS 24, 135, 208, 224, 264, 297, 339
 - OpenWRT-Router 285
 - OSVDB 223
 - OWASP 292
 - OWASP-ZAP 259
- P**
 - Package Manager 76
 - Packaging-Tool 173, 174
 - Paket
 - ändern 173
 - anpassen 167
 - neu erstellen 169
 - Versionsnummer 173
 - Paketabhängigkeit 168
 - Paketerstellungsprozess 175
 - Paros 239
 - Partition
 - verschlüsselte 82
 - Verschlüsselung 61
 - Partitionierung 68
 - geführte 68
 - Partitionierungstool 82, 86
 - Partitionsmodus
 - manueller 72
 - Pass the hash 231
 - Passwort 313
 - knacken 230
 - zurücksetzen 235
 - Passwort-Attacke 164
 - Passwortcracker
 - online 226
 - Passwortcracker-Tool 316
 - Passwortcracking 232
 - lokal 232
 - Passwort-Dump 341
 - Passwörter
 - decodieren 272
 - Passwörter knacken
 - Linux 234
 - OS X 234
 - Windows 232
 - Passwort-Hash 226, 230
 - Windows 318
 - Passwort-Hash-Datei 231
 - Passwort-Wörterbuch 309
 - Patch 298
 - Problem beheben 224
 - Patch-Level 151
 - Patch-Management-System 177
 - Payload 225, 296, 297, 299
 - PCAnywhere 226
 - PCAP 285
 - PCI-Gerät 50
 - PCMCIA-Karte 50
 - Penetration Testing Execution Standard 205
 - Penetrationstest 156
 - Ablauf 201
 - traditioneller 156
 - Vier-Schritte-Prozess 201
 - Penetrationstester 218
 - Permission to Attack 161
 - Persistence-Start 61
 - Persistenz 25, 60, 187, 188
 - verschlüsselt 190
 - Persistenzdateisystem 190
 - Persistenzpartition
 - verschlüsselt 192

Phishing 305, 306, 344
 Web-Vorlage 306
 Phishing-Seite 306
 Phreaker 345
 Physikalische Adresse 251
 Physische Partition 84
 PID 45
 Ping 208, 211
 Hacker-Werkzeug 212
 Ping-Scan 251
 Pipal 341
 Port 209
 Anzahl 213
 ermitteln 209
 offen 204, 208
 Verkehrsaufkommen 209
 Portscan 204, 208, 213, 214, 251, 303
 PostgreSQL-Cluster 113
 PPPoE 104
 Primäres Betriebssystem 101
 Programmausführungsfluss
 steuern 163
 Programmkonfiguration 34
 Proof-of-Concept-Code 162
 Protokoll
 verbindungsorientiertes 217
 Proxy 287
 konfigurieren 288
 ZAP 292
 Proxy-Adresse 76
 Prozess 37
 verwalten 45
 Prozess-ID 45
 Prozessorarchitektur 151
 Prozesspriorität 38
 PTA 161
 PTES 205

Q

Quellpaket 169
 aktualisieren 176
 erstellen 177
 Quellformat 174

R

Race Conditions 149
 RainbowCrack 25
 Randgeräte 211
 Raspberry Pi 91, 285
 RDP 226
 RDP-Client 91
 Recherche 205
 Recon 201

Reconnaissance 201
 RecordMyDesktop 341
 Recovery 330
 redfang 168
 Redirection 44
 Regelerstellung 132
 Regression 270
 Remote Desktop Protocol 226
 Remote-Codeausführung 298
 Remotecomputer 300
 Remotedesktopverbindung 91
 Remotedienst 226
 Remote-Shell 233
 Remotezugriff 110
 Remotezugriffsdienst 226
 Report 292
 Repository 31, 94
 Request for Comments 219
 Ressourcenverbrauch 162
 Reverse-Payload 302
 RFC 219
 Richtlinien
 Debian 30
 Kali Linux 30
 Richtlinien für Sicherheitsexperten 205
 Ringbuffer 50
 Risiko 147
 Risikobewertung 123, 152, 155
 Rolling Distribution 21
 Root 37
 Rootkit 243, 328, 345
 Rootkit-Erkennung 328
 Root-Konto 228
 Root-Passwort 264
 Root-Rechte 227
 Router 274
 RST-Paket 215

S

SAM 234
 SAM-Datei 232, 317, 318
 Samdump2 232, 317
 SAM-Sperre 232
 Scan
 authentifizierter 152
 automatisierter 151
 leiser 250
 Scannen 157
 Schnittstelle 36, 273
 dedizierte 180
 Schwachstelle 147, 149, 297
 ausnutzen 225
 ermitteln 210

- scannen 242
- Webapplikationen 241
- Schwachstellenanalyse 149, 150, 156
 - Tools 264
- Schwachstellenanalyse-Tools
 - automatisierte 162
- Schwachstellen-Scan 152, 204, 208, 210, 217, 224
 - automatisiert 294
 - Ergebnisse 151
 - Nikto 269
 - ZAP 295
- Schwachstellen-Scanner 135, 152, 224
 - Metasploit 296
- SD-Karte
 - startfähig 93
- Searchsploit 307
- Secure Shell 226
- Service-Manager 117
- Service-Unit 117
- SET 168, 305, 306
- setgid 46
- SET-Power-User 176
- setuid 46
- SHA 234
- SHA-256 330
- shadow 234
- Shell 40, 41, 214
- Shell-Zugriff 146, 229
 - administrativ 214
- Shrink Wrap Code 345
- Sicherheitscheck 147
- Sicherheitslücke 150, 225, 264, 345
- Sicherheitsparameter 156
- Sicherheitsprozesse 156
- Sicherheitsrichtlinien
 - definieren 122
- Sicherheitsupdate
 - Kernel 178
- Siege 270
 - URL-Formate 271
- Signatur 150
 - erstellen 151
- Signaturset 153
- Sitemap 291
- Skipfish 291
- Skript
 - ausführen 210, 222
- Slackware 19
- Sleuth Kit 323
- Sniffing 271, 275
- Sniffing Tools 152
- SNMP 218
- Social Engineer Toolkit (SET) 24, 305
- Social Engineering 207, 345
- Social-Engineering-Angriff 304
- Software-RAID 83
- Softwareversion 151
- Source-Paket 95
- Soziale Dienste 262
- Spam 345
- Sparta 262
- Speicherbeschädigung 163
- Speicher-Dumb 333
- Speicherforensik 332, 333
- Speichermedium
 - bootfähiges 62
- Speicherverbrauch 177
- Spider 294
 - automatisiert 240
- Spiderangriff 289
 - ZAP 294
- Spoofing 271, 345
- Spracheinstellung 63
- Spyware 345
- SQL-Befehle 149
- SQL-Injection 146, 149, 164, 240, 345
- SSH 110, 226
- SSH-Host-Schlüssel 110
- sshmitm 272
- SSID 282
- SSLstrip 138
- SSL-Zertifikat 137
- Stable Distribution 20
- Stack Buffer Overflow 163
- Standard-Angriffsziel 159
- Standard-Assessment 158
- Standardkonfiguration 186
 - optimieren 167
- Standard-Linux-Kernel 65
- Standardnetzwerkkonfiguration 104
- Standardportnummer 209
- Standardports 216
- Standard-Shell 108
- Startmedium 194
- Statistiken 341
- Subdomänen
 - aufspüren 254
- Subnetz 216
- Superuser-Root-Konto 66
- SWAP-Partition 26, 74, 86, 148
- Switch 236, 274
- SYN/ACK 214
- SYN-Flag 221
- SYN-Scan 214, 251
 - starten 215

syskey 317
 SYSTEM 234
 Systemd 50
 Systeme
 kritische 203
 Systemressource 45
 Systemsicherung 321
 SysVinit-Methode 19

T

Target-Unit 117
 Tarnung 215
 Tastaturlayout 64
 TCP 217
 TCP-Port 213
 TCP-RFC 220
 TCP-Stack 252
 TCP-Verbindung 111
 TCP-Verbindungsscan 215, 216, 217
 Telnet 226
 Terminal 39
 Texteditor 44
 TFTP 218
 TheHarvester 207, 254
 Threat 345
 Threats pro Scan 152
 Tool
 Dsniff 272
 Exploitation 295
 Man-in-the-Middle-Angriffe 272
 Penetrationstest 286
 Schwachstellenanalyse 264
 Sniffing 271
 Spoofing 271
 Tools
 automatisierter 153
 für Attacken 279
 zur Informationssammlung 249
 Torrent 57
 Traditioneller Penetrationstest 149, 156
 Transaktionsinformationen 270
 Trojaner 345
 True Negative 151
 True Positive 151

U

Überwachungsdienst 148
 Ubuntu 19
 UDP 217
 UDP-Port 213, 217
 UDP-Scan 217, 218
 UEFI 35
 UID-Variable 109

Umgebungsvariable 42, 43
 Unified Sniffing 274
 Unix 36, 46
 Unix-basiertes Betriebssystem 59
 Unix-Crypt(3)-Hash 313
 Unix-Derivate 94
 Upstream 174
 Upstream-Git-Repository 174
 Upstream-Version 167
 packen 176
 urlsnarf 272
 USB-Gerät 50
 User-Account 107
 User-Agent 337
 User-Space 51
 User-Space-Bibliothek 183

V

Validierungsprozess
 Tools 161
 Variable 42
 Verbindungsaufbau 214
 Verbindungsloses Protokoll 217
 Verbindungsorientiertes Protokoll 217
 Verfügbarkeit 145
 Verschleierung 215
 Verschlüsselte Partition 82
 Verschlüsselter Hash 231
 Verschlüsselung 317
 Verschlüsselungs-Passphrase 83
 Verschlüsselungsschlüssel 83
 Vertraulichkeit 145
 Verzeichnis 37
 Verzeichnisbaum 40
 VFAT 37
 Virtual Network Computing 226
 VirtualBox 22
 Virtuelle Konsole 39, 99
 Virtueller Host 114
 Virtuelles Dateisystem 50
 Virus 346
 VMware 22
 VNC 300
 VNC-Injektion 301
 VNC-Payload 233
 Volafox 332
 Volatility 333
 Volume-Gruppe 83
 Voreinstellungsdatei 194
 erstellen 195
 initrd 194
 Netzwerk 194
 Startmedium 194

VPN 105
 VPN-Netzwerk 267
 Vulnerability 149
 Vulnerability Analysis 150
 Vulnerability-Scanner 265

W

w3af 239
 Web Application Audit und Attack Framework 239
 Webanwendung 150, 158, 291
 Webanwendungs-Assessment 148
 Webapplication 150
 Webapplikation
 Schwachstellen 241
 Webframework 291
 Webgestützter Angriff 238
 Webhacking 239, 241
 Webkit-Rendering 335
 webmitm 272
 Web-Penetrationstest 293, 337
 Webpräsenz
 Unternehmen 238
 Webscanner 242
 WebScarab 241, 286
 Web-Schwachstelle 163
 Webseite 258
 analysieren 241
 Offline-Kopie 258
 Webserver 114, 268
 Informationen gewinnen 213
 websploit 272
 WEP 279
 WEP-Schlüssel 279, 280, 283
 knacken 280
 White-Box-Assessment 159
 Windows Subsystem for Linux 23
 Windows-Eingabeaufforderung 227
 Windows-Installation 79
 Windows-LM-Passwörter 232
 Windows-NT-basierte Systeme 318
 Windows-Partition 72
 verkleinern 80

Wireless Injection 19
 Wireless Security Assessment 148
 Wireless Wide Area Network 104
 Wireless-Assessments 148
 Wireshark 24, 238, 275
 WLAN-Hacking 279
 WLAN-Netzwerk
 aufspüren 285
 Worst-Case-Szenario 157
 WSL-Distribution 88
 Wurm 346
 WWAN 104

X

XFCE 34
 XFCE-Desktop 22
 Xmas-Scan 219, 220
 XSS 240, 346
 XSS-Angriff 164

Z

ZAP 292
 Zed Attack Proxy 292
 Zeitbombe 344
 Zenmap 213
 zenMap 249
 Zero-Day-Exploit 265
 Zielnetzwerk 150
 Zielorganisation 207
 Ziel-PC
 steuern per Kommandozeile 214
 Zombie-Drohne 346
 Zonentransfer 256
 ZSH-Terminal 339
 Zugangspunkt 209
 Zugriff
 administrativer 225
 festigen 204
 Zugriffsbeschränkung 117