Jürgen
Ebner

2. Auflage

Einstieg in Kali Linux

Penetration Testing und Ethical Hacking mit Linux

Inhaltsverzeichnis

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 Buch[1] 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.

Kapitel 2

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
- Enlightment
- LXDE
- XFCE

Die beiden ersten sind deutlich ressourcenhungriger. Enlightment, 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 Space[1].

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 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 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 größtenteils untätiges, hat fast immer Dutzende laufende Prozesse.

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

2.1.4 Rechtemanagement

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

2.2 Die Kommandozeile (Command Line)

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

2.2.1 Wie komme ich zur Kommandozeile?

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

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

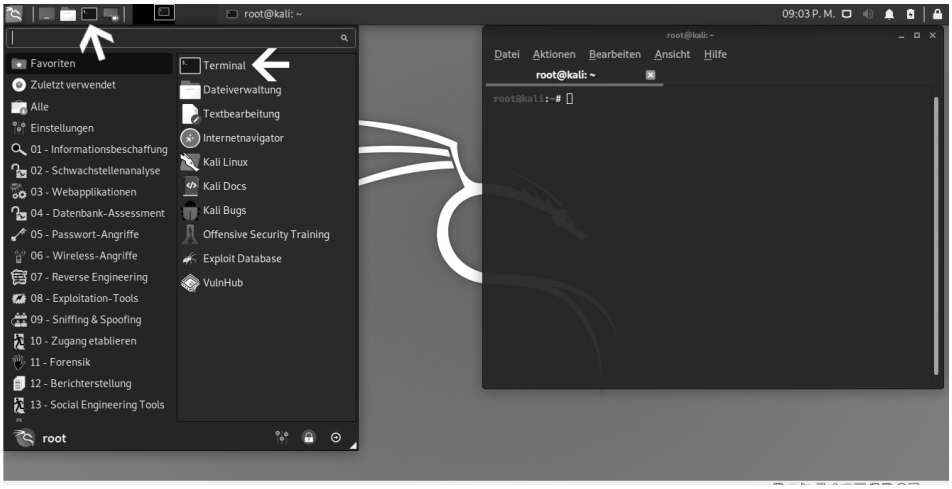

Abb. 2.2: Terminal aufrufen

Für den Fall, dass die grafische Benutzeroberfläche beschädigt ist, können Sie immer noch eine Befehlszeile auf virtuellen Konsolen erhalten (bis zu sechs davon sind über die sechs Tastenkombinationen Strg+Alt+F1 bis Strg+Alt+F6 aufrufbar, die Strg-Taste kann weggelassen werden, wenn Sie sich bereits im Textmodus außerhalb der grafischen Benutzeroberfläche von Xorg[2] oder Wayland[3] 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 **Sh**ell). 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 Zieldatei`*.

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

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

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

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

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

Hinweis

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

Umgebungsvariablen

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

2.3 Das Dateisystem

2.3.1 Dateisystem-Hierarchie-Standard

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

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

- */usr/*: Applikationen – das Verzeichnis wird in weitere Verzeichnisse geteilt, *bin*, *sbin*, *lib*, und folgt der gleichen Logik wie das Root-Verzeichnis. Des Weiteren enthält das Verzeichnis */usr/share/* Architektur-unabhängige Daten. Das Verzeichnis */usr/local/* wird vom Administrator für die manuelle Installation von Programmen verwendet, ohne dass Dateien überschrieben werden, die vom Paketsystem (dpkg) verwendet werden.
- */var/*: variable Daten, die von Daemon[4] verarbeitet werden. Das umfasst Protokolldateien, Warteschlangen, Spools und Caches.
- */proc/* und */sys/*: sind spezifische Linux-Kernel (und nicht Teil des FHS). Diese werden vom Kernel für den Export von Daten in den User-Space benötigt.

2.3.2 Das Home-Verzeichnis des Anwenders

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

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

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

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

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

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

2.4 Hilfreiche Befehle

2.4.1 Anzeigen und Ändern von Text-Dateien

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

Der Editor-Befehl (abhängig vom Editor) startet einen Texteditor (wie Vi oder Nano) und ermöglicht das Erstellen, Ändern und Lesen von Textdateien. Einfache Dateien können manchmal dank Redirection[5] 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.

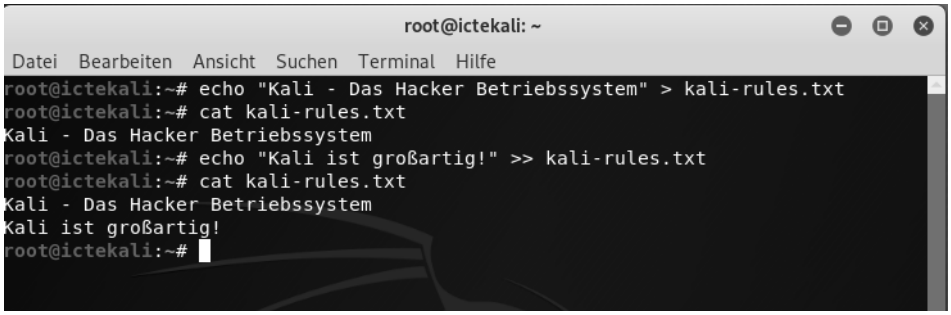

Abb. 2.6: Ausgabe von Befehlen in Datei umleiten

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.

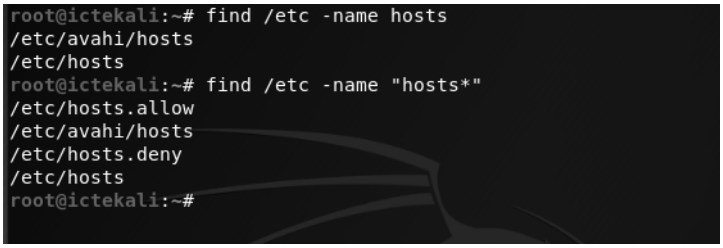

Abb. 2.7: Der Befehl `find` mit dem Suchkriterium `-name` in unterschiedlichen Varianten

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 e**x**ecute, 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
- `chggrp` *`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

- lspci (im Paket pciutils), das PCI-Geräte auflistet
- lsusb (im Paket usbutils), das USB-Geräte auflistet
- lspcmcia (im Paket pcmciautils), das PCMCIA-Karten auflistet

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

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

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

2.5 Zusammenfassung

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

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

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

- Zu den allgemeinen Befehlen gehören:
  - `pwd` – Arbeitsverzeichnis drucken
  - `cd` – Verzeichnis ändern
  - `ls` – Datei- und Verzeichnisinhalt auflisten
  - `mkdir` – Verzeichnis erstellen
  - `rmdir` – Verzeichnis entfernen
  - `mv`, `rm` und `cp` – Verschieben, Entfernen und Kopieren von Dateien bzw. Verzeichnissen
  - `cat` – 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
  - `journalcttl` – 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

C

D

E

F

G

H

I

J

K

O

P

Q

R

S