Bastian Ballmann

Network Hacks – Intensivkurs

Angriff und Verteidigung mit Python 3

2. Auflage

Springer Vieweg

Network Hacks – Intensivkurs

Bastian Ballmann

Network Hacks – Intensivkurs

Angriff und Verteidigung mit Python 3

2. Auflage

Springer Vieweg

Bastian Ballmann
Uster, Schweiz

ISBN 978-3-662-61635-2 ISBN 978-3-662-61636-9 (eBook)
https://doi.org/10.1007/978-3-662-61636-9

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Springer Vieweg

Planung: Martin Börger
Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer-Verlag GmbH, DE und ist ein Teil von Springer Nature.
Die Anschrift der Gesellschaft ist: Heidelberger Platz 3, 14197 Berlin, Germany

Für Datenreisende, Wissenshungrige und neugierige, netzwerkbegeisterte Lebewesen, die Spaß daran haben den Dingen auf den Grund zu gehen.

Geleitwort

Erklärt dieses Buch nicht nur wie man in Systeme einbricht? Ist das nicht illegal?

Der Autor möchte beide Fragen verneinen. Wissen per se ist nicht illegal, sondern höchstens die Handlungen, die man mit diesem Wissen begeht.

Sie als Admin, Programmierer, IT-Beauftrager oder interessierter User können sich nicht wirkungsvoll vor einem Angreifer schützen, wenn Sie dessen Methoden nicht kennen! Sie können die Wirksamkeit Ihrer Firewalls, Intrusion-Detection-Systeme und sonstiger Sicherheitssoftware nicht überprüfen und beurteilen, wenn Sie nicht in der Lage sind Ihr Netz aus der Sicht eines Angreifer zu sehen. Sie können nicht die Gefahren gegen die Aufwände möglicher Schutzvorkehrungen abwägen, wenn Sie die Auswirkungen eines Angriffs nicht oder nur unzureichend kennen. Deswegen ist es wichtig zu verstehen wie Angriffe auf Computernetzwerke funktionieren.

Eine Auswahl an Angriffsmöglichkeiten wird Ihnen im Buch anhand von kurzen, praktischen Code-Beispielen erklärt, die Ihnen wirkungsvolle Demonstrationsmöglichkeiten an die Hand geben, mit denen Sie IT-Entscheider davon überzeugen können, dass es sinnvoll wäre etwas mehr Budget für Sicherheit zu investieren. Sie sollten am Ende des Buches in Lage sein diese Beispiele nicht nur zu verstehen, sondern auch an Ihre eigenen Bedürfnisse anzupassen.

Natürlich lehrt dieses Buch ebenfalls den bösen Buben, wie er eigene Angriffstools schreiben kann. IT-Security ist ein zweischneidiges Schwert und ein ständiger Wettkampf, der von der absichernden Seite niemals gewonnen werden kann, wenn sie sich selbst ihres Wissens beraubt!

Einleitung

Für wen ist dieses Buch?

Dieses Buch richtet sich an interessierte Python-Programmierer, die ihr Grundwissen mit einer gehörigen Portion Netzwerk-Code erweitern möchten, und an versierte Administratoren, die aktiv die Sicherheit ihrer Systeme und Netze überprüfen wollen. Der Inhalt dürfte ebenfalls White-, Gray- und BlackHat-Hacker interessieren, die wie ich Python als ihre bevorzugte Programmiersprache für kleine und große Hacks und Exploits entdeckt haben. Interessierte Computerbenutzer, die selber einmal lernen möchten ihr Netzwerk mit den Augen eines Angreifers zu sehen, werden genauso auf ihre Kosten kommen.

Es werden weder Kenntnisse in Python noch in Netzwerktechnologie vorausgesetzt. Das Wissen, das für dieses Buch benötigt wird, wird in den Kap. 2 und 3 vermittelt. Leser, die schon über ausreichende Python- und Netzwerk-Kenntnisse verfügen und eine bevorzugte Python-IDE ihr Eigen nennen, können sofort zu Kap. 5 springen und sich umgehend in die Hacking-Techniken stürzen.

Sie sollten das erlernte Wissen selbstverständlich nur auf ihre eigenen Systeme und Netzwerke bzw. nur mit ausdrücklicher Erlaubnis der Betreiber anwenden, da Sie ansonsten wahrscheinlich eine strafbare Handlung begehen!

Der Umfang dieses Buches erlaubt es nicht, die behandelten Themengebiete in voller Tiefe zu ergründen. Es will Basiswissen auf den wichtigsten netzwerkspezifischen Gebieten aufbauen. Falls Sie sich anschließend mit einem oder mehreren Bereichen eingehender befassen wollen, sollten Sie sich für diese Bereiche extra Fachliteratur anschaffen.

Wie ist dieses Buch aufgebaut?

Die verschiedenen Hacks sind nach Netzwerkprotokollen gruppiert und innerhalb der Kapitel nach Schwierigkeitsgrad geordnet. Abgesehen von den beiden Grundlagenkapiteln über Netzwerke (Kap. 2) und Python (Kap. 3) können die Kapitel in beliebiger Reihenfolge gelesen werden.

Die Codebeispiele sind ungekürzt abgedruckt, damit sie komplett abgetippt werden können. Sie können sie alternativ auch über Github downloaden unter https://github.com/balle/python-network-hacks.

Am Ende eines jeden Kapitels werden Tools vorgestellt, die in Python programmiert sind und das jeweilige Protokoll angreifen, das in dem Kapitel behandelt wurde. Mit dem fundierten Vorwissen sollte es Ihnen dann nicht allzu schwer fallen, die Source Codes dieser Programme zu lesen und zu verstehen.

Die wichtigsten Sicherheitsprinzipien

Die wichtigsten Prinzipien beim Aufbau eines sicheren Netzes sind nach Auffassung des Autors:

1. Sicherheitslösungen sollten simpel sein. Firewall-Regeln, die niemand mehr verstehen kann, sind eine Garantie für Sicherheitslücken. Software, die kompliziert ist, hat mehr Bugs als simpler Code.
2. Weniger ist mehr. Mehr Code, mehr Systeme, mehr Server bieten mehr Angriffsfläche.
3. Sicherheitslösungen sollten Open Source sein. Andere Mitmenschen können nicht so effektiv nach Sicherheitslücken suchen, wenn der Source Code nicht verfügbar ist. Falls der Hersteller eine Sicherheitslücke gar nicht oder erst in ein paar Monaten beheben will, haben Sie kaum eine bis gar keine Möglichkeit, die Lücke selbst zu beheben. Proprietäre Software beinhaltet außerdem des öfteren Hintertüren (manchmal Law-Interception-Interface genannt). Firmen wie Cisco (RFC 3924), Skype (US-Patent-Nr 20110153809) und Microsoft (z. B. _NSAKEY siehe http://www.heise.de/tp/artikel/5/5263/1.html) belegen dies.
4. Eine Firewall ist nur ein Teil eines Sicherheitkonzepts, keine Box, die man hinstellt und dann ist man sicher.
5. Bleiben Sie auf dem neuesten Stand! Was heute als sicher gilt, kann in ein paar Stunden schon als Einfallstor missbraucht werden. Halten Sie deshalb alle Software und alle Systeme auf dem neuesten Stand, auch Drucker, Switches und Smartphones!
6. Die am schwächsten abgesicherte Komponente bestimmt die Qualität der Gesamtsicherheit, und das ist manchmal kein Computer, Telefon oder Drucker, sondern ein Mensch (Stichwort Social Engineering).
7. Es gibt keine 100%ige Sicherheit. Selbst ein ausgeschalteter Computer kann durch einen raffinierten Social Engineer noch missbraucht werden. Sie können es einem Angreifer nur so schwer machen, dass es seine Fähigkeiten übersteigt oder es sich für ihn nicht lohnt, doch dafür müssen Sie die Techniken und die Motivation eines Angreifers kennen.

Inhaltsverzeichnis

Installation

1

Zusammenfassung

In diesem Kapitel erfahren Sie, für welche Betriebssysteme die Source Codes entwickelt wurden und auf welchen sie lauffähig sind, welche Python-Version Sie benötigen und wie man Python-Module bequem suchen, installieren und updaten kann. Des Weiteren werden eine Reihe von Entwicklungsumgebungen vorgestellt, um Ihnen eine Übersicht samt Entscheidungshilfen für eine moderne Entwicklungsumgebung zu geben, die Ihnen einen Teil der Arbeit abnimmt, und Sie bei der Fehlersuche unterstützt. Sie können natürlich die Quellcodes auch mit einem einfachen Texteditor eingeben.

1.1 Das richtige Betriebssystem

Alle Quellcodes dieses Buches wurden unter GNU/Linux Kernelversion 5.x geschrieben und sind auch nur unter Linux und OpenBSD getestet worden; sie sollten allerdings ebenfalls unter jeder anderen Linux Version lauffähig sein. Vom Kapitel über Bluetooth abgesehen sollten die Code-Beispiele auch unter anderen BSD-Derivaten und unter Mac OS X einwandfrei funktionieren. Der Autor freut sich über Erfolgsmeldungen per Mail. Von Netzwerk-Hacking unter Windows hält der Autor allerdings nicht sehr viel und kann deswegen keinerlei Aussage über die Lauffähigkeit der Scripte unter diesem Betriebssystem machen.

Falls Sie kein Linux- oder BSD-System installiert haben, reicht es aus ein Image in einer VirtualBox- (www.virtualbox.org) oder Vmware-Virtualisierungslösung (www.vmware.com) zu installieren. Entsprechende vorinstallierte Images finden Sie für VirtualBox unter virtualboxes.org und für Vmware unter www.vmware.com/appliances.

B. Ballmann, *Network Hacks – Intensivkurs*,
https://doi.org/10.1007/978-3-662-61636-9_1

1.2 Die richtige Python-Version

Alle Quellcodes wurden für Python Version 3 entwickelt und mit Python 3.7 getestet.

Um zu überprüfen, welche Version von Python auf Ihrem System installiert ist, führen Sie den nachfolgenden Befehl aus:

```
python3 --version
Python 3.7.4
```

1.3 Entwicklungsumgebung

Der Autor bevorzugt GNU/Emacs (www.gnu.org/software/emacs) als Entwicklungsumgebung, weil er die Editier- und Erweiterungsmöglichkeiten für unschlagbar hält. Emacs bietet alle gängigen Features wie Syntax Highlighting, Code Completion, Code Templates, Debugger Support, PyLint Integration und hat dank dem Gespann Rope, Pymacs und Ropemacs mit eine der besten Refactoring-Unterstützungen für Python. Um sofort in den Genuss all dieser Features zu kommen, empfiehlt der Autor die Installation der Erweiterung Emacs-for-Python, zu finden unter gabrielelanaro.github.com/emacs-for-python. Dank einer Menge an Plugins kann Emacs noch erweitert werden, z. B. zum E-Mail- und News-Client, IRC-Chat-Client oder Music-Player, und weitere Features bieten wie Sprachunterstützung, eingebaute Shells und Datei-Explorer bis hin zu Spielen wie Tetris. Manche Mitmenschen meinen, Emacs sei eher ein Betriebssystem als eine IDE.

Als alternativer Consolen-Editor sei hier natürlich ebenso Vi bzw. Vim (www.vim.org) erwähnt, um keine Glaubenskriege auszulösen oder zu unterstützen. Vi bietet ebenfalls alle gängigen Features einer modernen IDE. Wie gut die Python-Unterstützung ist, kann der Autor mangels Erfahrung allerdings nicht beurteilen.

Wer lieber mit einer grafischen Entwicklungsumgebung arbeiten möchte, dem sei als erstes die Entwicklung Eclipse (www.eclipse.org) und PyDev (pydev.org) nahegelegt. Eclipse bietet neben den üblichen Features Code Outline, einen verbesserten Debugger Support und eine schier unglaubliche Anzahl an weiteren Plugins wie z. B. UMLet für UML-Diagramme und Mylyn für die Integration eines Bugtracking-Systems.

Als alternative GUI-IDE möchte der Autor noch Eric4 (eric-ide.python-projects.org) und Spyder (code.google.com/p/spyderlib) aufführen, die ebenfalls die Standardeigenschaften plus Debugger, PyLint Support und Refactoring bieten.

Wer nicht viele Ressourcen zum Programmieren zur Verfügung hat, aber eine GUI bevorzugt, dem empfiehlt der Autor Gedit mit den Plugins Class Browser, Externe Werkzeuge, PyLint, Python Code Completion, Python Doc String Wizard, Python Outline, Quelltext Kommentar und Rope Plugin. Die Installation ist etwas aufwändiger und im Funktionsumfang ein wenig eingeschränkter als bei den vorgenannten Umgebungen, allerdings verbraucht Gedit auch nur etwas ein Zehntel der Ressourcen von Eclipse.

Die Qual der Wahl sei dem Leser überlassen. Wer nicht wählen und mit möglichst geringem Aufwand einsteigen will, der installiert Eclipse und PyDev als Bundle von Aptana (aptana.com/products/studio3).

1.4 Python-Module

Python-Module findet man im Python-Packet-Index, der über pypi.python.org erreichbar ist. Neue Module können in drei Varianten installiert werden:

1. Download des Source-Archives, Entpacken und anschließendes Ausführen der magischen Zeile

   ```
   python3 setup.py install
   ```

2. Verwenden von easy_install mittels

   ```
   easy_install <modulname>
   ```

3. Mit Hilfe von pip (hierfür muss ggf. das Paket `python-pip` nachinstalliert werden)

   ```
   pip3 install <modulname>
   ```

Der Autor bevorzugt die Verwendung von `pip`, denn mit `pip` können Sie nicht nur bequem neue Module installieren und alte deinstallieren, sondern auch vorhandene updaten, in Listen exportieren, um sie andernorts alle zu reinstallieren, Module suchen und mehr.

Alternativ können Sie pip auch mitteilen, dass es die Module in einem Unterordner in Ihrem Home-Verzeichnis installieren soll, indem Sie den Parameter `-user` hinzufügen.

Welche Python-Module für welche Tools und Scripte gebraucht werden, steht entweder am Anfang eines Kapitels oder vor dem jeweiligen Codeabschnitt, damit Sie nur die Module installieren müssen, die Sie auch wirklich verwenden wollen.

1.5 Pip

Pip kann nicht nur Module installieren, es bietet ebenfalls die Möglichkeit mit dem Kommando `search` nach Modulen zu suchen.

```
pip3 search <modulname>
```

Um ein Modul zu deinstallieren, verwenden Sie das Kommando `uninstall`. Eine Auflistung aller installierter Module und deren Versionen liefert der Parameter `freeze`. Die Liste kann in eine Datei geschrieben und Anschließend wieder eingespielt werden.

```
pip3 freeze > requirements.txt
pip3 install -r requirements.txt
```

Welche Module veraltet sind, verrät der Befehl `pip list -outdated`, ein einzelnes Modul kann mittels `pip3 install -upgrade <modulname>` aktualisiert werden.

1.6 Virtualenv

Sie haben auch die Möglichkeit alle für dieses Buch benötigten Python-Module in einen Unterordner in einer sogenannten Virtualenv zu speichern, so dass sie nicht mit den Modulen des Betriebssystems in Konflikt geraten. Als Beispiel legen wir die Virtualenv `python-network-hacks` an, installieren das Modul `scapy` und verlassen die virtuelle Umgebung wieder.

```
python3 -m venv python-network-hacks
source python-network-hacks/bin/activate
(python-network-hacks) $ pip3 install scapy
(python-network-hacks) $ deactivate
$ _
```

Stellen Sie nachdem dem Ausführen des Kommandos `deactivate` sicher, dass Sie sich nicht mehr in der Virtualenv befinden. Der Prompt muss wieder dem Default-Prompt entsprechen.

Netzwerk 4 Newbies

2

Zusammenfassung

Computernetzwerke sind die Adern des Informationszeitalters, Protokolle die Sprache des Netzes.

Dieses Kapitel vermittelt Grundkenntnisse in Sachen Networking von der Hardware und dem Aufbau über die Funktionsweise aller gängigen Protokolle eines Ethernet-IP-TCP-Netzwerks bis hin zu Topologien und Man-in-the-middle-Attacken. Für alle, die ihr Wissen in Sachen Netzwerke erneuern oder neu aufbauen wollen.

2.1 Komponenten

Um überhaupt ein Computernetzwerk aufbauen zu können, braucht man eine Reihe von Hardware-Komponenten. Je nach Netzart umfassen diese neben Computern und Netzwerkkarten noch Kabel, Modems, altmodische Akkustikkoppler in Bananenkisten, Richtfunkantennen oder Satellitenschüsseln, sowie Router (Abschn. 2.14), Gateways (Abschn. 2.13), Firewalls (Abschn. 2.18), Bridges (Abschn. 2.15), Hubs und Switches.

Ein **Hub** ist einfach nur ein Kasten, in den viele Netzwerkkabel gesteckt werden und der alle eingehenden Signale an alle angeschlossenen Kabel weiterschickt. Diese Eigenschaft führt nicht nur zu einer Explosion des Netzwerktraffics, sondern auch dazu, dass Hubs heutzutage nicht mehr verbaut werden. Stattdessen setzt man **Switches** ein, um Netzwerkverbindungen zu bündeln. Der Unterschied zum Hub besteht darin, dass ein Switch sich die MAC-Adresse der Netzwerkkarte am Ende des Kabels merkt und Traffic gezielt nur an den Port verschickt, an dem der Zielrechner angeschlossen ist. Was MAC-Adressen sind und wie die Adressierung genau funktioniert, wird im Abschn. 2.4 erklärt.

B. Ballmann, *Network Hacks – Intensivkurs*,
https://doi.org/10.1007/978-3-662-61636-9_2

2.2 Topologien

Computernetzwerke kann man auf unterschiedliche Arten verkabeln. Die heutzutage üblichste Variante sind **Stern-Netzwerke** (siehe Abb. 2.1), bei denen alle angeschlossenen Computer über ein zentrales Verbindungsgerät miteinander verbunden sind. Der Nachteil dieser Verkabelungsart ist, dass ein Single-Point-of-Failure besteht und, dass das gesamte Netz zusammenbricht, sobald die zentrale Komponente ausfällt. Dieser Nachteil kann allerdings durch redundant (mehrfach) ausgelegte Komponenten umgangen werden.

Eine weitere Möglichkeit ist, alle Computer in einer Reihe miteinander zu verbinden, das sogenannte **Bus-Netzwerk** (siehe Abb. 2.2). Nachteil dieser Topologie ist, dass jeder angeschlossene Computer über zwei Netzwerkkarten verfügen muss und die Daten ggf. über viele Rechner verschickt werden. Sollte einer von ihnen ausfallen, können die dahinter liegenden Computer nicht mehr erreicht werden, und wenn ein Computer in der Kette unter hoher Last leidet, wird er zwangsweise zum Flaschenhals der Netzwerk-Kommunikation.

Der Autor hat in seiner beruflichen Laufbahn bisher nur wenige Bus-Netzwerke zu Gesicht bekommen und alle bestanden aus zwei Computern, die über diese Direktverbindung zeitkritische oder Traffic-intensive Dienste gefahren haben, wie das Replizieren von großen Datenbanken, Clustering von Applikation-Servern oder Syncen von Backupdaten auf einen weiteren Server. In allen Fällen diente das Bus-Netzwerk dazu, das Stern-Netz zu entlasten.

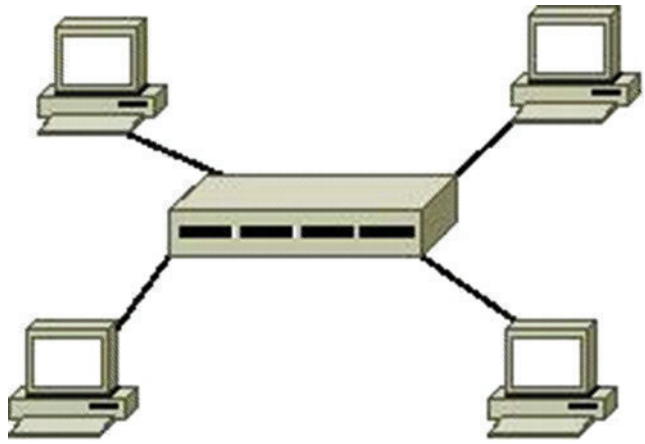


Abb. 2.1 Stern-Netzwerk

Abb. 2.2 Bus-Netzwerk

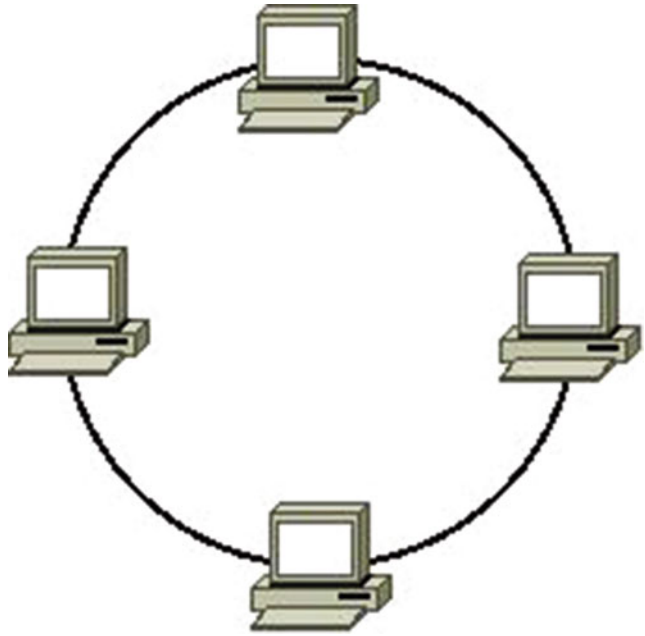


Abb. 2.3 Ring-Netzwerk

Als letzte Variante sei der Vollständigkeit halber noch das **Ring-Netzwerk** (Abb. 2.3) erwähnt, bei dem, wie der Name schon sagt, alle Computer im Kreis angeschlossen werden. Das Ring-Netz hat dieselben Nachteile wie ein Bus-Netzwerk mit dem Unterschied, dass das Netz nicht teilweise zusammenbricht, sobald ein Computer ausfällt. In diesem Fall kann der Traffic in einem Ring-Netz einfach in die entgegengesetzte Richtung umgeleitet werden. Der Autor hat selber noch kein Ringnetz implementiert gesehen, hat sich aber sagen lassen, dass diese Topologie für Backbones (Netzwerkrückgrat) bei ISPs und großen Firmen verwendet wird.

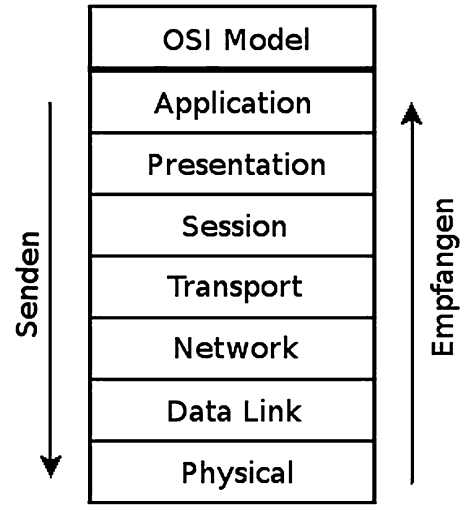
Des Weiteren hört oder liest man öfters von **LAN** (Local Area Network), **WAN** (Wide Area Network) und manchmal auch von **MAN** (Middle Area Network). Ein LAN ist ein lokales Netzwerk, das meist auf ein Gebäude, ein Stockwerk oder ein Zimmer begrenzt ist. In modernen Netzen sind die Computer eines LANs mittels eines oder mehrerer Switches miteinander verbunden. Verbindet man mehrere LANs über Router oder VPNs (siehe Abschn. 2.17), so erhält man ein MAN. Umspannt das Netzwerk, wie beim Internet, gar mehrere Länder oder die ganze Welt, spricht man von einem WAN.

2.3 ISO/OSI Schichtenmodell

Nach der reinen Lehre, dem sogenannten ISO/OSI-Schichtenmodell, besteht ein Computernetzwerk technisch aus sieben verschiedenen Ebenen, sogenannten Layern (siehe Abb. 2.4).

Jeder dieser Layer besitzt eine klar abgegrenzte Aufgabe und jedes Datenpaket passiert im Betriebssystemkernel nach und nach alle Layer, bis zu dem Layer, auf dem es arbeitet (Tab. 2.1).

Abb. 2.4 OSI-Modell



Tab. 2.1 OSI-Layer

OSI-Layer	Layer-Name	Funktionalität
1	Physical	Kabel, Richtfunkantennen, etc.
2	Data-Link	Stellt eine Punkt-zu-Punkt-Verbindung zwischen zwei Computern her
3	Network	Sorgt für die Adressierung des Zielsystems
4	Transport	Sorgt dafür, dass Daten in der richtigen Reihenfolge ankommen und bei Verlust erneut geschickt werden
5	Session	Dient dazu, eine Kommunikation zwischen Anwendungen aufzubauen (z. B. mittels Ports)
6	Presentation	Umwandlung von Datenformaten und -codierungen (z. B. Komprimierung / Verschlüsselung)
7	Application	Protokolle, die die eigentliche Anwendung implementieren, z. B. HTTP

2.4 Ethernet

Sind Sie schon einmal in einen Laden gegangen und haben Netzwerkkabel und -karten gekauft? Dann besitzen Sie mit ziemlicher Sicherheit Ethernet-Hardware, denn Ethernet ist das mit großem Abstand am weitesten verbreitete Netzwerk. Die Netzkomponenten gibt es in unterschiedlichen Geschwindigkeitsstufen wie 1, 10, 100 MBit oder Gigabit, und ein Ethernet kann über verschiedene Kabelarten wie Koaxial (veraltet), Twisted-Pair (die üblichen Netzwerkkabel aus dem Laden Ihres Vertrauens) oder Glasfaser (für Datenhungrige) aufgebaut werden. Bei Twisted-Pair-Kabeln unterscheidet man sowohl **STP** – (Single-Twisted-Pair) und **UTP** – (Unshielded-Twisted-Pair) Kabel, als auch Patch- und Crossover-Kabel.

Der Unterschied zwischen STP- und UTP-Kabeln ist, dass die Adern in den UTP-Kabeln nicht abgeschirmt sind, was zur Folge hat, dass sie eine schlechtere Qualität haben als STP-Kabel. Heutzutage findet man fast nur noch STP-Kabel im Laden.

Patch- und Cross-Kabel unterscheidet man, indem man die beiden Steckerköpfe des Kabels nebeneinanderhält. Ist die Farbreihenfolge der **Adern gekreuzt**, also andersherum, handelt es sich um ein **Cross-Kabel** , welches dazu verwendet wird zwei Computer direkt miteinander zu verbinden. Ist die **Reihenfolge gleich**, handelt es sich um ein **Patch-Kabel**, mit dem man einen Computer und einen Switch oder Hub verbinden kann.

Jede Netzwerkkarte in einem Ethernet-Netzwerk besitzt eine weltweit eindeutige MAC-Adresse, die dazu dient, einen Computer in einem Ethernet-Netzwerk zu adressieren. Eine **MAC-Adresse** besteht aus 6 zweistelligen Hexadezimalzahlen, die durch Doppelpunkte getrennt werden (z. B. `aa:bb:cc:11:22:33`).

Es ist ein weit verbreiteter Irrglaube, dass Computer in einem lokalen TCP/IP-Netzwerk über die IP-Adresse angesprochen werden; in Wirklichkeit wird dazu die MAC-Adresse verwendet. Eine weitere falsche Annahme ist, dass MAC-Adressen nicht gefälscht werden können. Tatsächlich wird die MAC-Adresse im Betriebssystemkernel in das Netzwerkpaket geschrieben, und Betriebssysteme wie GNU/Linux oder *BSD bieten die Möglichkeit, mit einem einzigen Befehl die MAC-Adresse zu ändern.

```
ifconfig enp3s0f1 hw ether c0:de:de:ad:be:ef
```

Ein Ethernet-Header (siehe Abb. 2.5) enthält neben einer Source- und Destination-MAC-Adresse nur noch ein Typ-Feld und eine Checksumme. Das Typ-Feld gibt das übergeordnete Protokoll an, z. B. 0x0800 für IP oder 0x0806 für ARP.

Als Letztes sei noch der Begriff CSMA/CD erwähnt. CSMA/CD steht für Carrier Sense Multiple Access/Collision Detect und beschreibt, wie ein Computer Daten in einem Ethernet sendet. Zuerst horcht die Netzwerkkarte auf dem Kabel, ob gerade schon Daten gesendet werden. Ist dies der Fall, wartet sie eine zufällige Zeit und versucht es dann erneut. Ist die Leitung frei, sendet sie die Daten ins Netzwerk. Sollten zwei oder mehr Komponenten gleichzeitig senden, kommt es zu einer Kollision. Komponenten, die Daten senden, horchen weiterhin auf der Leitung, erkennen dadurch die Kollision,