Bibliothek des technischen Wissens

Netzwerkanalyse mit Wireshark 2.0

Einführung in die Protokollanalyse

Bernhard J. Hauser

1. Auflage

VERLAG EUROPA-LEHRMITTEL · Nourney, Vollmer GmbH & Co. KG
Düsselberger Straße 23 · 42781 Haan-Gruiten

Europa-Nr.: 54081

Autor:
Hauser, Bernhard J. Dipl.-Ing. Bisingen

Verlagslektorat:
Alexander Barth Dipl.-Ing. Haan

Bildentwürfe: Der Autor

Bildbearbeitung:
tiff.any GmbH, Berlin

1. Auflage 2016

Druck 5 4 3 2 1

Alle Drucke derselben Auflage sind parallel einsetzbar, da sie bis auf die Behebung von Druckfehlern untereinander unverändert sind.

ISBN 978-3-8085-5408-1

Satz: tiff.any GmbH, Berlin
Umschlaggestaltung: braunwerbeagentur, 42477 Radevormwald
Umschlagfoto: © Péter Mács – Fotolia.com
Druck: BALTO Print, Vilnius LT-08217, Litauen

Vorwort

Gerald Combs begann 1997 mit der Entwicklung eines Netzwerkprotokoll-Analysators namens *Ethereal*, dessen erste Version im Juli 1998 erschien. Der *Sniffer* (also „Schnüffler") verbreitete sich rasch und wurde zum Standardwerkzeug vieler Netzwerker, von denen einige Gerald Combs zu unterstützen begannen und das Programm sukzessive erweiterten.

Seit 2006 wird das Projekt unter dem Namen *Wireshark* geführt. Es ist in laufender Entwicklung – ständig kommen neue Protokolle und Funktionen hinzu. Im November 2015 wurde nach etwa 2-jähriger Entwicklungszeit die Version 2.0 freigegeben. Der Sprung in der Versionsnummer von 1.12 auf 2.0 verdeutlicht, dass dabei einiges verändert wurde.

Welche Aufgaben aber hat eine Sniffer-Software wie Wireshark? Kurz: Sie zeichnet den Datenverkehr in Rechnernetzen auf und analysiert ihn. Sie ist eine Art Fenster in die Netze, durch das wir sehen, welche Daten über Netzwerkleitungen übertragen werden.

Das Open-Source-Produkt Wireshark ist in dieser Kategorie ein sehr mächtiges Werkzeug und bringt die bekannten Vorteile freier Software mit: Nichts ist geheim oder geschieht im Verborgenen – jeder kann den Quelltext der Software einsehen und ggf. an eigene Bedürfnisse anpassen. Sie steht jedem kostenlos zu Verfügung und braucht den Vergleich mit kommerziellen Sniffern und Packet-Analyzern nicht scheuen. So ist Wireshark mit über 500 000 Downloads pro Monat unbestreitbar das bekannteste Werkzeug zur Netzwerkanalyse.

Wireshark interpretiert alle gängigen Netzwerkprotokolle. Dazu gehören neben der TCP/IP-Familie selbstverständlich auch DSL, ATM und WLAN. Zudem ist es für alle bekannten Betriebssysteme verfügbar: Linux, Solaris, NetBSD, OpenBSD, Mac OS, Windows, Android usw.

An wen wendet sich dieses Buch? Oder anders: Wer nutzt Wireshark und für welchen Zweck?

- Netzwerk-Administratoren zur Fehlersuche im Netzwerk
- Netzwerk-Administratoren zum Aufspüren von unnötigem Datenverkehr im Netzwerk
- Netzwerk-Administratoren zum Aufspüren von Störern und Eindringlingen
- Netzwerk-Sicherheitsingenieure bei der Suche und Analyse von Sicherheitsproblemen
- Software-Entwickler für Tests und Protokoll-Implementationen
- Studenten und angehende Netzwerker für ein eingehendes Verständnis von Netzwerkprotokollen

Dieses Buch ist aus der Notwendigkeit heraus entstanden, Schülern und Studenten die Grundlagen der Netzwerktechnik näherzubringen. Die verschiedenen Protokollfunktionen und deren Ineinandergreifen auf den Schichten der Netzwerkkommunikation sind durch eigenes Erforschen deutlich einfacher nachzuvollziehen und damit nachhaltiger zu verstehen als durch bloße Lektüre der Fachliteratur. Dabei ist Wireshark ein Werkzeug von unschätzbarem Wert.

Der Praxisbezug, insbesondere zu Themen wie „Filter" oder zur Anpassung von Wireshark an spezielle Aufgaben, hilft auch dem erfahrenen Netzwerker, sich mit Wireshark ein starkes Analysetool für die tägliche Überwachung und die Fehlersuche zu erschließen.

Dieses Buch versteht sich lediglich als Einführung in das komplexe Thema der Netzwerktechnik. Danach stehen Ihnen über die Angebote der Projekt-Webseite (z. B. die FAQ) oder die Aktivitäten der Community (z. B. in Form von Wiki oder Mailinglisten) weitere Hilfen zur Verfügung, die Sie für sich entdecken und nutzen sollten.

Gegenüber älteren Versionen von Wireshark (*legacy Wireshark* genannt) wartet die Version 2 mit einigen Neuerungen auf. Die Benutzerfreundlichkeit wurde noch weiter verbessert, die grafischen Ausgaben wurden überarbeitet. So kann jetzt im Fenster des Graphen die Darstellungsform geändert werden, x- und y-Achse sind zoombar und vieles mehr. Des Weiteren wurden die Menüs in viele Sprachen übersetzt; neben Englisch, Polnisch, Spanisch und vielen weiteren Sprachen ist Wireshark 2 auch ins Deutsche übersetzt worden.

Über weite Bereiche sind aber alte und neue Version gleich, sodass hier auch ältere Screenshots verwendet werden konnten.

Autor und Verlag sind für alle Hinweise und Anregungen dankbar, die die Weiterentwicklung dieses Buches unterstützen. Schreiben Sie uns unter lektorat@europa-lehrmittel.de.
Wir wünschen Ihnen eine anregende Lektüre und nützliche Erkenntnisse mit diesem Buch!

Sommer 2016 Autor & Verlag

Inhaltsverzeichnis

1 Grundlagen

1.1 Wie arbeiten Sniffer?

Daten werden in Netzwerken als *Datenpakete* (*Packets*) versendet. Jedes dieser Pakete erhält zu Beginn seiner „Reise" eine Kennung, wie zum Beispiel die Adressen von Absender- und Zielrechner, und hat eine bestimmte maximale Länge, also eine maximale Anzahl von Bytes.

Eine Netzwerkkarte empfängt zwar grundsätzlich alle Datenpakete, die an ihrem Anschluss ankommen, filtert im Normalfall aber dann jene heraus, die für diesen Anschluss bestimmt sind. Alle anderen verwirft sie.

Damit Wireshark grundsätzlich *alle* Pakete, die im Netzwerk unterwegs sind, aufzeichnet, versetzt man die Karte in den sogenannten *Promiscuous Mode*. So registriert sie nicht nur Pakete, die explizit an ihre eigene MAC-Adresse adressiert sind, sondern auch *Multicasts*. Multicasts sind Gruppenadressen, z. B. alle Router, eine bestimmte Auswahl von Rechnern etc. oder *Broadcasts*, also Sendungen, die für *alle* Karten eines Netzes bestimmt sind. Wireshark nimmt diese Daten auf und gibt sie anschließend an das Betriebssystem weiter.

Die empfangenen Pakete lassen sich nun abspeichern und untersuchen.

1.2 Rechtliche Grundlagen

Sniffer wie Wireshark dürfen Sie zur Fehlersuche und zum Studium einsetzen. Natürlich bieten diese Programme auch die Möglichkeit, sich fremde Daten, Passwörter usw. zu beschaffen. Das sind allerdings Straftaten, sofern man dazu nicht ausdrücklich berechtigt ist bzw. von einem Befugten den klaren (schriftlichen) Auftrag erhalten hat.

Das Grundgesetz ist hier eindeutig:

> Das Briefgeheimnis sowie das Post- und Fernmeldegeheimnis sind unverletzlich.
>
> — *Grundgesetz – Artikel 10*

Hinzu kommen einschlägige Gesetze zur Telekommunikation oder auch der sog. „Hackerparagraph" aus dem Strafgesetzbuch:

> (1) Wer eine Straftat nach § 202a oder § 202b vorbereitet, indem er
>
> 1. Passwörter oder sonstige Sicherungscodes, die den Zugang zu Daten (§ 202a Abs. 2) ermöglichen, oder
> 2. Computerprogramme, deren Zweck die Begehung einer solchen Tat ist, herstellt, sich oder einem anderen verschafft, verkauft, einem anderen überlässt, verbreitet oder sonst zugänglich macht, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft. Strafgesetzbuch (StGB) – § 202c
>
> (2) § 149 Abs. 2 und 3 gilt entsprechend.
>
> — *Strafgesetzbuch (StGB) – § 202c*

Im Grunde verhält es sich wie mit jedem Werkzeug: Man darf es verwenden, sofern man keine Straftaten damit begeht.

1.3 Ein Schnellstart

Bevor die theoretischen Grundlagen der Netzwerktechnik betrachtet werden, wird hier ein Schnellstart in die Tiefen von Wireshark vorgenommen, aus dem erkennbar wird, wie einfach sich umfangreiche und auch hochsensible Daten im Netzwerk „erschnüffeln" lassen.

Beenden Sie alle offenen Programme, insbesondere E-Mail-Programme und Clients für soziale Netzwerke, Chats usw.

Öffnen Sie einen Browser und gehen Sie zu einem gewöhnlichen Webshop oder einem Forum. Die meisten dieser Shops legen wenig Wert auf Abhörsicherheit. Achten Sie darauf, dass es sich um eine unverschlüsselte Übertragung (ohne https) handelt.

Starten Sie Wireshark, indem Sie die Schnittstelle auswählen, auf der Sie kommunizieren. Im Normalfall ist dies die erste Ethernet-Schnittstelle (`eth0`).

Melden Sie sich im Browser bei dem gewählten Webshop oder Forum mit einem Fantasienamen und einem Fantasiepasswort an. Natürlich schlägt die Anmeldung fehl, aber das kann ignoriert werden.

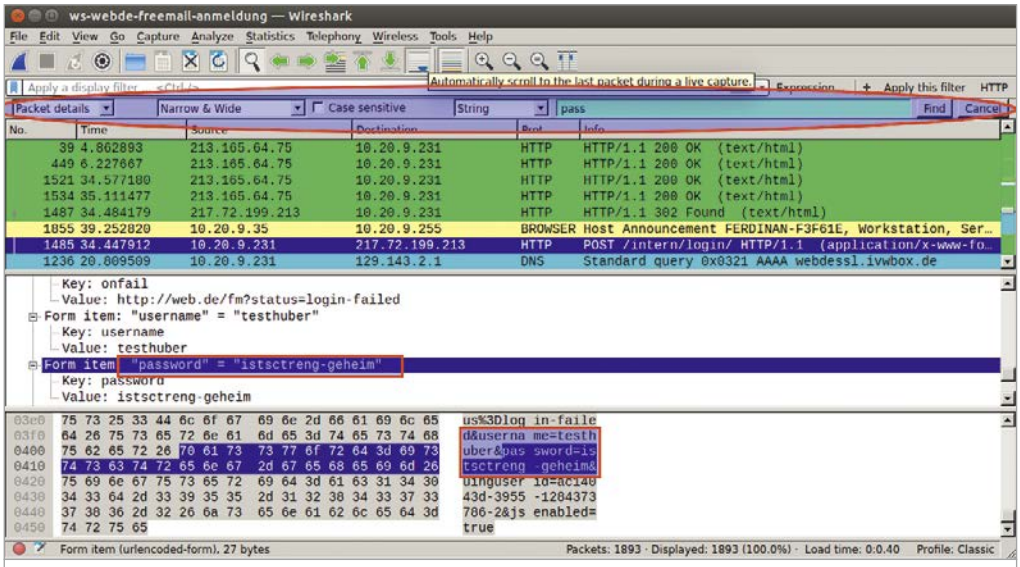

Bild 1.1: So schnell findet man ein unverschlüsseltes Passwort

Stoppen Sie dann die Wireshark-Aufzeichnung und sehen Sie sich die Menge der aufgezeichneten Pakete an. Es dürften sich schon einige angesammelt haben. Gehen Sie nun auf die Paketsuche, indem Sie Ctrl F eingeben (oder über das Menü *Edit▸Find Packet*).

Die Zeile „Paket finden" wird oberhalb der Paketliste eingeblendet. Hier suchen Sie in den Paket-Details nach dem String *pass* – und finden so die Pakete, die diese Zeichenkette beinhalten.

Im Normalfall finden Sie es sofort, und das Paket, das Ihr Passwort enthält, wird angezeigt.

Sollten Sie Ihr Passwort nicht gleich finden, kann das daran liegen, dass die Seite HTTPS verwendet und die Eingaben verschlüsselt überträgt.

Jetzt verstehen Sie auch die rechtlichen Hinweise im vorangegangenen Abschnitt, denn mit nur wenigen Einstellungen und der entsprechenden Berechtigung haben Sie auf diese Weise Zugriff auf *alle* Eingaben *aller* Nutzer in Ihrem Netzwerk.

Bei älteren Wireshark-Versionen (vor 2.0) öffnet sich zum Suchen mit Ctrl F ein separates Suchfenster. Die Eingaben sind identisch bei allen Versionen.

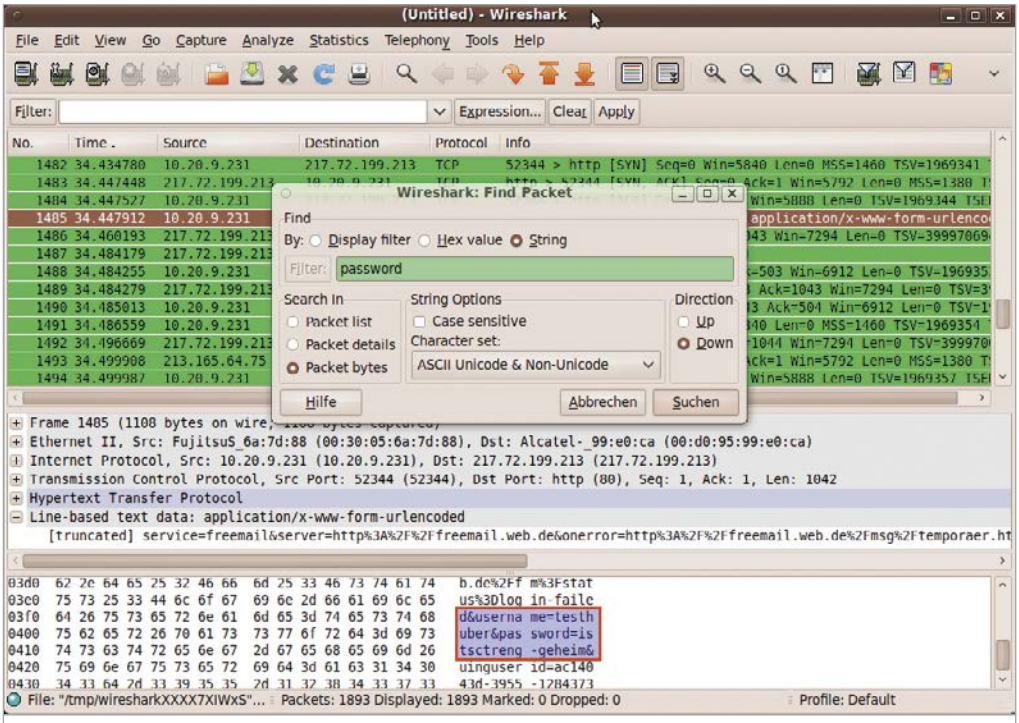

Bild 1.2: Paket finden mit einer älteren Wireshark-Version

1.4 Die notwendige Theorie

Ein klein wenig Theorie muss bekannt sein, bevor man mit Wireshark richtig loslegen kann. Sie wird hier auf ein absolutes Minimum beschränkt und kann auch als Wiederholung und Auffrischung von bereits Gelerntem dienen.

1.4.1 Referenzmodelle

Modelle dienen der Vereinfachung und veranschaulichen komplexere Sachverhalte – das gilt auch in der Datenkommunikation. Hier bedient man sich üblicherweise eines Schichtenmodells: Die sehr unterschiedlichen Aufgaben, die beim Senden und Empfangen von Daten anfallen, werden auf verschiedenen Ebenen (oder eben Schichten) erledigt. Jeder Schicht ist ein Programmmodul zugewiesen, das immer nur mit der Schicht darüber und der Schicht darunter kommuniziert, also in der Kommunikation niemals Schichten überspringt. Dieses Verfahren sorgt auch für hohe Flexibilität: Sind Änderungen an der Datenkommunikation notwendig, muss nur das entsprechende Programmmodul verändert oder ausgetauscht werden.

In jedem dieser Schichtenmodelle (es gibt verschiedene), befindet sich ganz oben die *Anwendung*. Ganz unten, sozusagen an der Basis, liegt die *physikalische Übertragung*, also Leitungen, Funk, Ströme, Spannungen usw.

Datenkommunikation im Netzwerk läuft stets nach demselben Schema ab: Daten werden, ausgehend von einer Anwendung auf einem Rechner, über mehrere Schichten (*Layer*) „nach unten" bis zur Basis, der Schicht der Bit-Übertragung, durchgereicht.

Jede Schicht hat ihre spezielle Funktion und erweitert die zu übertragenden (Inhalts-)Daten um einen eigenen „Protokollkopf" (*Protocol Header*). Darüber hinaus nimmt sie gegebenenfalls auch Änderungen an den Daten selbst vor, bevor sie diese zuletzt einkapselt und weiterreicht (*Encapsulation*).

Der Empfänger (also üblicherweise ein anderer Rechner im Netzwerk) nimmt die Daten als Folge von Nullen und Einsen entgegen und entfernt der Reihe nach alle Header, bis die Originaldaten des Senders übrigbleiben.

In LANs (*Local Area Networks*) ist heute überwiegend die Protokoll-Suite TCP/IP auf Ethernet im Einsatz. Ältere Netzwerktechnologien wie ARCNET, Token-Ring, VG-Anylan usw. sind weitgehend verschwunden. In den Anfängen der LAN-Technik herrschte ein Durcheinander an firmenspezifischen Lösungen. Man sah sich genötigt, durch entsprechende Normen Struktur und Ordnung zu schaffen, sodass auch Geräte unterschiedlicher Hersteller miteinander kommunizieren konnten. Die *International Organization of Standardization* (ISO) veröffentlichte dazu ein Schichtenmodell zur Datenkommunikation, das *Open Systems Interconnect Model* oder kurz: OSI-Modell.

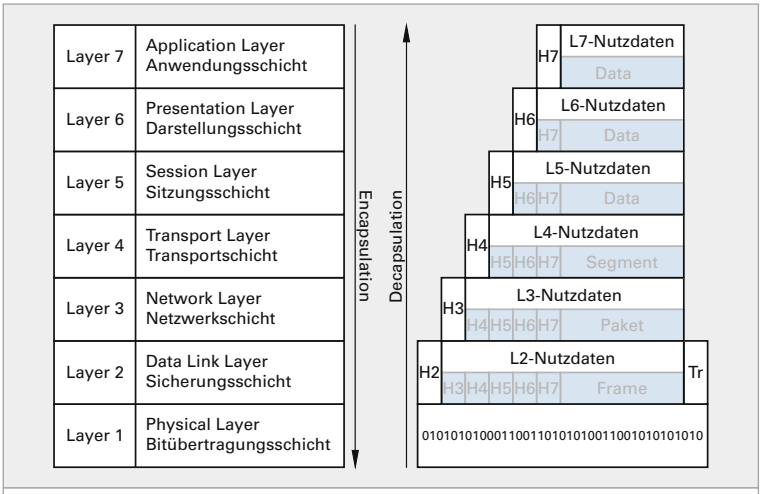

Bild 1.3: Protokollverschachtelung (Encapsulation)

Unabhängig davon wurde auf Veranlassung des US-amerikanischen Verteidigungsministeriums (Department of Defense, DoD) an einigen Universitäten das *DoD-Modell* entwickelt, das heute als TCP/IP-Modell bekannt ist.

TCP/IP steht für die beiden beteiligten Protokolle: *Transmission Control Protocol* und *Internet Protocol*.

Auf jeder Schicht dieser beiden Modelle (ISO/OSI und TCP/IP) gibt es eine Reihe von Protokollen mit ganz spezifischen Aufgaben, die im Folgenden näher betrachtet wird.

Das TCP/IP-Modell, das auch in Weitverkehrsnetzen häufig eingesetzt wird, kennt nur vier Schichten, das OSI-Modell hingegen sieben.

Bild 1.4 stellt beide Modelle einander gegenüber. Sie sehen, dass TCP/IP-Layer 1 dasselbe leistet wie die OSI-Layer 1 und 2 zusammen.

OSI-Layer 3 und 4 haben jeweils ihr Pendant im TCP/IP-Modell.

Die OSI-Layer 5, 6 und 7 wiederum entsprechen beim TCP/IP-Modell der Anwendungsschicht.

Um Unterschiede und Gemeinsamkeiten deutlich zu machen, wird im Folgenden die Funktionsweise anhand des OSI-Modells betrachtet. Tatsächlich nutzt keine Protokollsammlung alle 7 Schichten, wie im OSI-Modell definiert. Es wird hier das OSI-Modell beschrieben, weil es die Schichten und deren Funktionen gut fassbar macht.

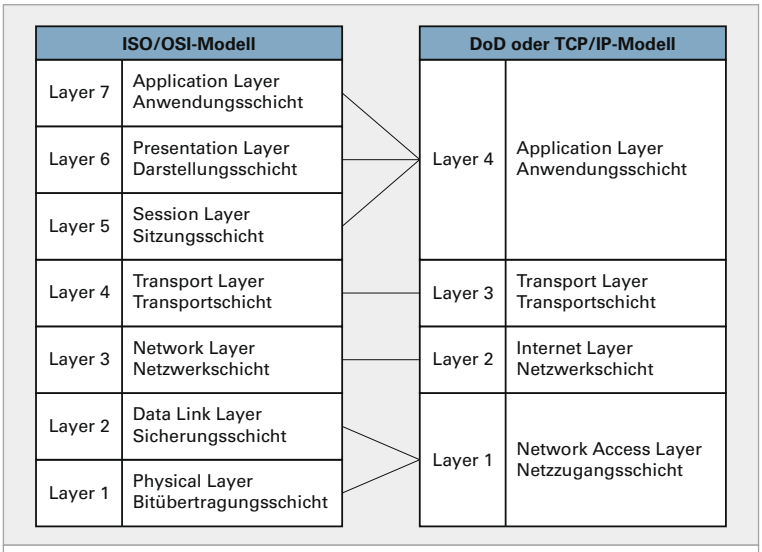

Bild 1.4: Gegenüberstellung von ISO/OSI-Modell und TCP/IP-Modell

Es wird „oben" (Anwendung) begonnen und nach „unten" (Übertragung) fortgesetzt.

***Upper Layers/Application Layer* (OSI-Layer 5 bis 7)**
Auf den Schichten oberhalb der Transportschicht befinden sich die Anwendungen mit bekannten Protokollen wie HTTP (Webseiten), FTP (Dateiübertragung) oder SMTP (E-Mail). Hier werden Daten erzeugt und nach unten zum Versenden weitergereicht. Ein Klick auf einen Link erzeugt beispielsweise eine Anfrage, die als HTTP-Request an die darunterliegende Transportschicht gegeben wird.

***Transport Layer* (OSI-Layer 4)**
Er bereitet die „von oben" kommenden Daten für den Transport vor, indem er die Anwendungen auf Absender- und Zielrechner adressiert. Diese Anwendungsadressen sind die *Ports*. Jede Netzwerkanwendung hat eine auf dem jeweiligen Rechner einmalige Port-Nummer. So kommuniziert immer ein Port des Clients mit einem Port des Servers, beispielsweise Browser und Webserver. Auf dieser Schicht findet die Überprüfung des Transportkanals statt, indem eine Kommunikation der beteiligten Stationen aufgebaut und nach der Datenübertragung wieder abgebaut wird. Gesendete Datenpakete werden quittiert. Bleiben Quittungen aus, erfolgt eine erneute Übertragung. Große Datenmengen werden auf dieser Schicht in kleinere Teile (*Segmente*) mit maximal 64 kByte aufgeteilt. Layer-4-Pakete werden dann an die Schicht 3 weitergereicht.

***Network Layer* (OSI-Layer 3)**
Auf der Netzwerkschicht werden die von Schicht 4 kommenden Daten für den Weg durch die Netzwerke vorbereitet, und zwar mithilfe der Netzwerkadressen (meist sind dies IP-Adressen). IP-Adressen sind zweigeteilt und adressieren ein Netzwerk, wie auch einen darin befindlichen Rechner. Die Grenze ist variabel. Man nennt die auf Layer 3 vorbereiteten Einheiten *Packets* oder *Pakete*. Zu große Segmente werden hier in noch kleinere Pakete aufgebrochen („fragmentiert"). Die maximale

Größe der Pakete ist abhängig von den darunter liegenden Schichten. Bei Ethernet sind dies meist 1500 Bytes an Nutzdaten.

***Data Link Layer* (OSI-Layer 2)**
Auf der *Datensicherungsschicht* werden die von Schicht 3 kommenden Daten für das darunter liegende Übertragungsmedium (Kupferleitung, Glasfaserleitung, Funk usw.) von Layer 1 vorbereitet. Hier werden ein Layer-2-Header vorangestellt und eine Prüfsumme an die Daten angehängt. Layer-3-Daten werden in einen Rahmen, bestehend aus *Header* und *Trailer* (hier die Prüfsumme), gebettet. Man nennt diese Datenpakete auch Layer 2 *Frames*. Auf dieser Schicht werden auch die Netzwerkkarten über die hardwarenahen MAC-Adressen (*Media Access Control*) adressiert. Mithilfe der Prüfsumme sind Übertragungsfehler zu erkennen, allerdings nicht zu beheben.

***Physical Layer* (OSI-Layer 1)**
Auf der *Bitübertragungsschicht* schließlich werden die von Schicht 2 kommenden Daten für die Übertragung über das Übertragungsmedium vorbereitet. Hier sind z. B. Ströme, Spannungen, Leitungen, Stecker, Lichtsignale, Funkfrequenzen usw. definiert.

1.4.2 Adressen im Netzwerk

Für die korrekte Übertragung ist die eindeutige Bestimmung von Start- und Zielpunkt unumgänglich. Im Netzwerk werden auf drei Ebenen Adressen verwendet:

TCP- oder UDP-Ports (Layer 4)
adressieren die Anwendungen (z. B. Port `80` für HTTP)

IP-Adressen (Layer 3)
adressieren einen bestimmten Host in einem bestimmten Netz

MAC-Adressen (Layer 2)
adressieren eine bestimmte Netzwerkkarte

Ports und IP-Adressen sind logische, MAC-Adressen sind physikalische Adressen.

1.4.3 Netzwerkgeräte

Im Folgenden werden die wichtigsten Netzwerkgeräte und ihre Rolle nach dem ISO/OSI-Schichtenmodell genauer beschrieben.

Layer 1
Die einfachsten Geräte im Netzwerk sind die *Repeater*, also „Signalauffrischer". Sie verstärken ein schwaches Netzwerksignal und liefern am Ausgang wieder den vollen Signalpegel. Mehrere Repeater in einem Gehäuse ergeben einen *Multiport Repeater*, auch *Hub* genannt. Hubs und Repeater arbeiten auf dem Physical Layer, der untersten Schicht der Modelle.

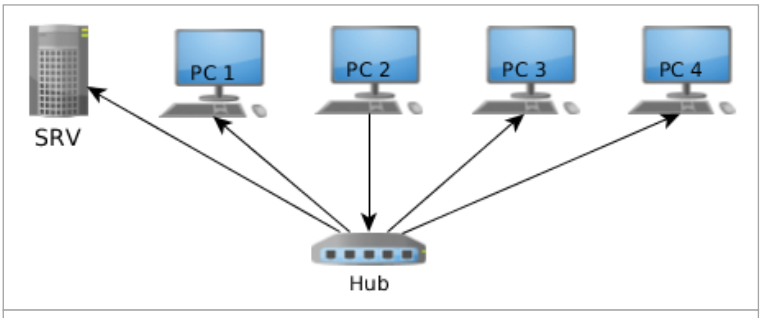

Bild 1.5: PC 2 schickt Daten über den Hub an den Server

Ein Signal an einen Hub schickt dieser auf allen anderen Anschlüssen verstärkt weiter (Bild 1.5) – nicht nur an den Empfänger!

Hubs und Repeater werden allerdings seit Jahren nicht mehr produziert; in älteren Installationen sind sie fast vollständig verschwunden und wurden durch Switches ersetzt.

Layer 2
Auf Layer 2 arbeiten *Bridges*, die Netzwerk-Brücken. Sie verfügen über Intelligenz und filtern den Datenverkehr: Sie lassen Pakete passieren, die an den jeweils anderen Anschluss gerichtet sind, andere Pakete leiten sie nicht weiter. Mehrere Bridges in einem Gehäuse bilden eine *Multiport Bridge*, auch als *Switch* bezeichnet.

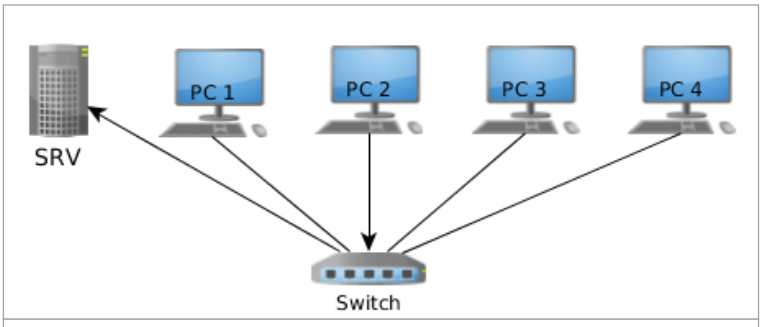

Bild 1.6: PC 2 schickt Daten über den Switch an den Server

Ein Switch „lernt", an welchem seiner Ports welche Netzwerkkarten angeschlossen sind. In einer Tabelle listet er zu jedem Anschluss die MAC-Adressen der daran angeschlossenen Geräte. Wenn er ein Paket empfängt, entscheidet er anhand dieser Tabelle, an welchen Port er das Paket weiterleitet. Somit landen die Datenpakete nur an den Stationen, die sie bekommen sollen. Alle anderen Stationen im Netzwerk bekommen von diesem Datentransfer nichts mit.

Befinden sich nur Switches in einem Netzwerk und ist an jedem Switchport nur ein einziger Rechner angeschlossen, spricht man von einem „voll geswitchten" oder auch „mikrosegmentierten Netz". Dies ist heute üblich.

An einem Rechner, auf dem ein Sniffer läuft, sind folglich nur die Daten zu empfangen, die entweder für diesen oder für alle Rechner bestimmt sind (*Broadcast*).

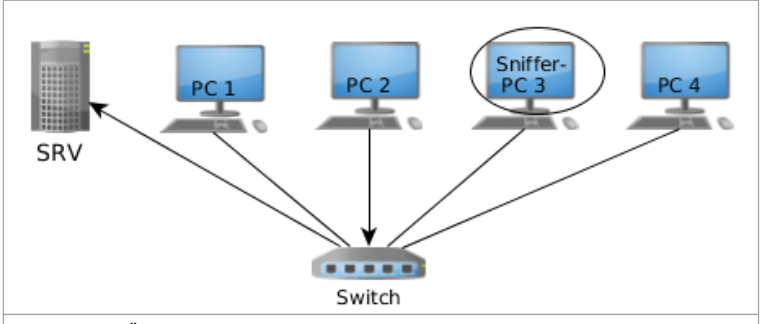

Bild 1.7: Ein Überwachungs-PC am Switch

Bild 1.7 zeigt ein typisches Szenario für einen Rechner im Netzwerk, auf dem Wireshark läuft. Auch wenn damit „nur" der eigene Netzwerkverkehr und der Broadcast-Verkehr aufgezeichnet werden kann, lassen sich auf diese Weise schon wesentliche Erkenntnisse über das Netzwerk sammeln.

Layer 3
Auf Layer 3 des OSI-Modells arbeiten die *Router*. Sie verbinden Netzwerke miteinander und werten dazu die Netzwerkadressen der Schicht 3 aus. Häufig kommt dabei TCP/IP zum Einsatz – die Netzwerkadressen sind die IP-Adressen. Andere Netzwerke, beispielsweise Novells IPX/SPX, nutzen IPX-Adressen.

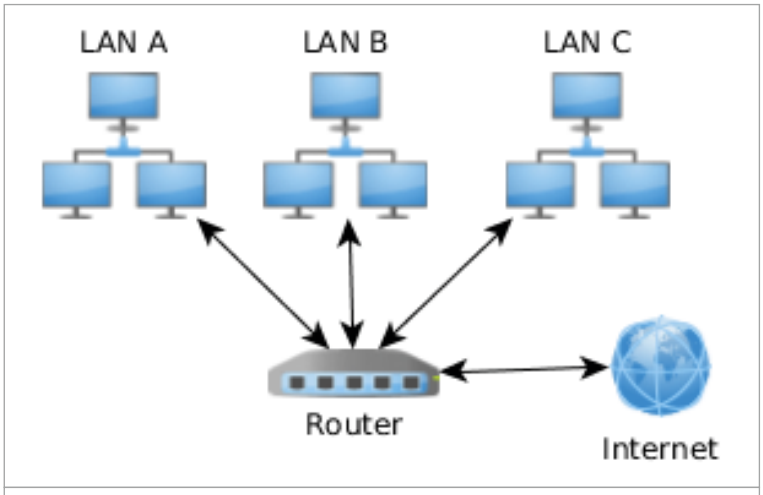

Bild 1.8: Ein Router verbindet mehrere Netzwerke

Router filtern den Datenverkehr und transportieren nur Datenpakete weiter, die für ein anderes Netzwerk bestimmt sind. Sniffing über Router hinweg ist daher sehr schwierig.

1.4.4 Verkabelung

Die Verkabelung eines Netzwerks erfolgt üblicherweise nach der Europäischen Norm EN 50173-1. Die Leitungsführung ist meist sternförmig. Dies gilt im Heimbereich ebenso wie in großen Konzernen und Behörden. In Firmen spricht man von einer „strukturierten, diensteneutralen Verkabelung" oder einer „universellen Gebäudeverkabelung" (UGV).

Die heute vorherrschende Netzwerktopologie ist die erweiterte Sterntopologie (*Extended Star*), die ein Firmengelände oder einen Campus in drei Verkabelungsbereiche einteilt.

Im ersten Verkabelungsbereich (*Primärverkabelung* oder *Core Layer*) werden, ausgehend von einem Standortverteiler (SV), sternförmig alle Gebäude miteinander verkabelt.

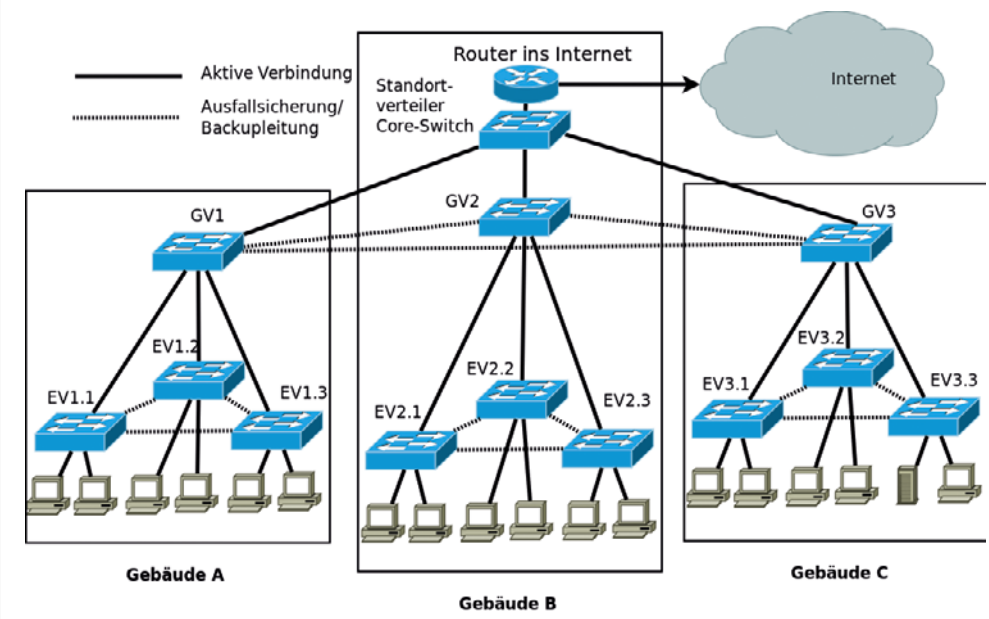

Bild 1.9: Strukturierte Verkabelung

Die *Sekundärverkabelung* (oder *Distribution Layer*), also der zweite Verkabelungsbereich, verbindet innerhalb eines Gebäudes oder Gebäudeteiles alle Etagen oder Abteilungen miteinander. Ausgangspunkt ist jeweils ein *Gebäudeverteiler* (GV), der jede Etage anfährt.

Auf jeder Etage wiederum sind die Netzwerkdosen sternförmig mit dem jeweiligen *Etagenverteiler* (EV) verbunden (*Tertiärverkabelung* oder *Access Layer*).

In den Verteilern befinden sich heute ausschließlich Switches, in großen Netzen zur Aufteilung in kleinere Subnetze, darüber hinaus Router. Um die Ausfallsicherheit von Netzen zu erhöhen, werden Querverbindungen hergestellt: So wird aus dem Sternnetz ein Maschennetz. Durch diese Schleifen im Netz entstehen Broadcast-Stürme. Die Switches deaktivieren einzelne Ports, sodass alle Schleifen aufgetrennt werden. Nach diesem Auftrennen der Schleifen hat das Netz wieder eine eindeutige Sternstruktur.

Bild 1.9 zeigt die Verkabelung dreier Gebäude. Die Gebäudeverteiler sind untereinander über Sicherungsleitungen (gestrichelt) miteinander verbunden. Ebenso sind die Etagenverteiler in den einzelnen Gebäuden mehrfach untereinander verbunden.

INFO: Es kann immer nur der Datenverkehr aufgezeichnet werden, der beim Sniffer ankommt!

Für das Mitschneiden von Daten ist es wichtig, dass der Sniffer möglichst nah an dem Rechner platziert wird, dessen Datenverkehr untersucht werden soll. In Kapitel 9 wird auf die Technik und das Platzieren der Geräte eingegangen, sofern Sie Wireshark nicht direkt auf Ihrem zu überwachenden Rechner installiert haben.

2 Bedienung

2.1 Startbildschirm und Hauptfenster

Nach dem Aufrufen des Programms sieht man folgenden Startbildschirm.

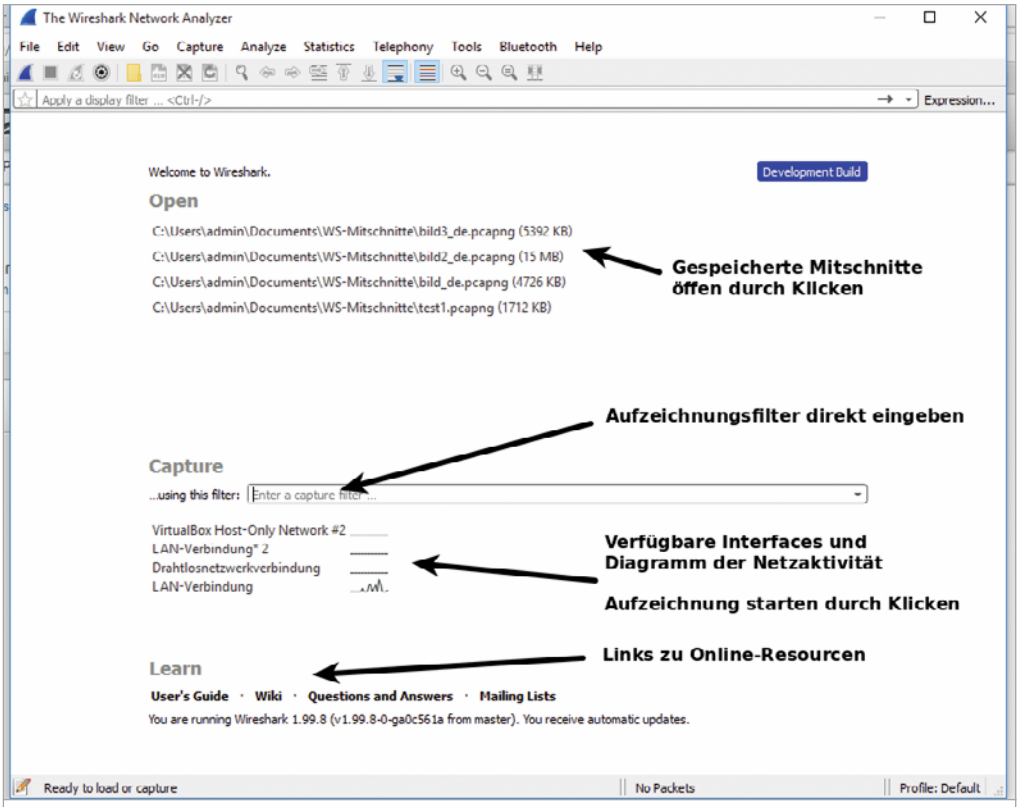

Bild 2.1: Der Startbildschirm von Wireshark 2.0

Nach dem Öffnen eines gespeichertem Mitschnitts oder nach dem Start einer Aufzeichnung bekommt man das dreigeteilte Hauptfenster zu sehen.

Das Wireshark-Hauptfenster (Bild 2.2) zeigt ganz oben das Befehlsmenü, darunter eine konfigurierbare Werkzeugleiste. Der Rest des Hauptfensters ist in drei Bereiche unterteilt – am unteren Rand des Fensters befindet sich die *Statuszeile*.

Im oberen Bereich des Hauptfensters sieht man die *Paketliste*: Alle empfangenen Pakete sind durchnummeriert (Nummernspalte) und mit Zeitstempeln versehen (Zeit/Time-Spalte). Ziel- und Quelladresse der Pakete, der transportierte Protokolltyp und die wichtigsten Inhalte sind in einer Zeile zusammengefasst. Eine kurze Vorschau auf den Inhalt eines jeden Paketes findet man in der Spalte Info.

Die Liste der mitgeschnittenen/angezeigten Pakete ist im Normalfall chronologisch geordnet, d. h. fortlaufend nummeriert (erste Spalte, in der Reihenfolge ihres Eintreffens). Durch Klick in das entsprechende Überschriftenfeld der jeweiligen Spalte lässt sich die Sortierung ändern. Ein kleiner Pfeil nach oben oder unten zeigt an, nach welchem Feld und in welcher Reihenfolge sortiert wurde.

Eine nützliche Neuerung ist das Anzeigen von zusammengehörigen Paketen durch eine Klammer ganz links in der Nummernspalte. Das erste und das letzte zu einen Konversation gehörende Paket sind durch einen horizontalen Pfeil markiert. Zwischen den Pfeilen zieht sich ein vertikaler Strich. Ist dieser Strich durchgezogen, dann gehört das daneben stehende Paket zu dieser Konversation. Ist er gestrichelt, dann gehört dieses Paket nicht zur ausgewählten Verbindung.

Ein Beispiel: Ein Klick in das *Time*-Feld ändert die Reihenfolge der Pakete nach der Zeit. Die Pakete werden dadurch rückwärts sortiert, die jüngsten Pakete stehen oben. Durch erneutes Klicken in dieses Feld wird die Sortierreihenfolge wieder umgekehrt.

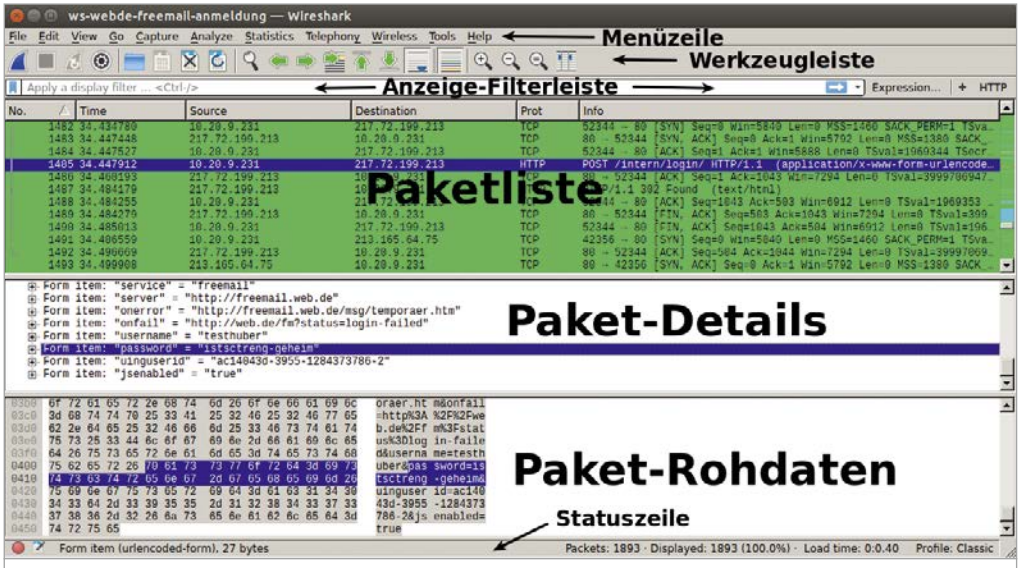

Bild 2.2 Der Aufbau des Hauptfensters

Ein Klick in das *Source*-Feld sortiert die Pakete nach der Quell-Adresse. Das Ergebnis zeigt Bild 2.3.

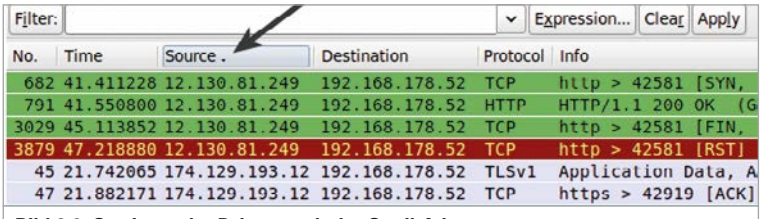

Bild 2.3: Sortieren der Pakete nach der Quell-Adresse