WILHELM BÜCHNER HOCHSCHULE
Mobile University of Technology

Studienheft

SRN05

Sichere Netzwerkkommunikation

SRN05

Sichere Netzwerkkommunikation

Prof. Dr. Wolfgang Hommel

Werden Personenbezeichnungen aus Gründen der besseren Lesbarkeit nur in der männlichen oder weiblichen Form verwendet, so schließt dies das jeweils andere Geschlecht mit ein.

Falls wir in unseren Studienheften auf Seiten im Internet verweisen, haben wir diese nach sorgfältigen Erwägungen ausgewählt. Auf die zukünftige Gestaltung und den Inhalt der Seiten haben wir jedoch keinen Einfluss. Wir distanzieren uns daher ausdrücklich von diesen Seiten, soweit darin rechtswidrige, insbesondere jugendgefährdende oder verfassungsfeindliche Inhalte zutage treten sollten.

Sichere Netzwerkkommunikation

Inhaltsverzeichnis

Vorwort

Erinnern Sie sich noch an die Zeit, als Breitband-Internet-Flatrates für Privathaushalte noch unüblich waren und mit Handies nur telefoniert werden konnte? Inzwischen ist die Vernetzung von Geräten und Anwendungen zum Datenaustausch global völlig unverzichtbar geworden. Ohne sie wäre die Zusammenarbeit in und zwischen Unternehmen mittlerweile ebenso undenkbar wie der persönliche Informationsaustausch mit Bekannten und die bequeme Nutzung von Fitness-Trackern, Kameras und Smart-Home-Komponenten im Zusammenspiel mit dem eigenen PC, Notebook und Tablet zu Hause. Immer mehr Daten werden global verteilt, auf Servern im Internet gespeichert und so flexibel zugänglich gemacht – ein fehlender oder gestörter Internetzugang wird von vielen heutzutage sogar als ähnlich einschränkend empfunden wie ein Stromausfall.

Standardisierte Kommunikationsprotokolle definieren die Formate und Abläufe, die für einen erfolgreichen Datenaustausch zwischen vernetzten Systemen eingehalten werden müssen. Dank moderner Programmierschnittstellen sind sie einfach anzuwenden; sie stammen aber aus einer Ära, in der die heutige Bedeutung der Netzwerkkommunikation und der IT-Sicherheit nur ansatzweise erahnt werden konnte. Ohne das Nachrüsten von dedizierten Sicherheitsmaßnahmen sind Kommunikationsvorgänge deshalb trivial angreifbar. Wie Sie schnell sehen werden, sind naheliegende Lösungsansätze wie die verschlüsselte Datenübertragung bei global verteilten Systemen aber nicht ganz einfach umzusetzen. Für sich allein genommen wären sie sogar fast völlig wirkungslos. Vielmehr ist eine systematische Implementierung und Anwendung von Kommunikationsprotokollen unter diversen Sicherheitsaspekten erforderlich.

In diesem Studienheft setzen Sie sich zunächst damit auseinander, welche grundlegenden Ziele sichere Netzwerkkommunikation verfolgt und wie diese durch Angriffe beeinträchtigt werden können. In Kapitel 2 rekapitulieren Sie diejenigen kryptografischen Verfahren, ohne die global skalierende sichere Kommunikationsprotokolle heute technisch unmöglich wären. In Kapitel 3 und Kapitel 4 lernen Sie die Funktionsweise und praktische Anwendung ausgewählter Protokolle unterhalb und auf der Anwendungsebene des ISO/OSI-Schichtenmodells kennen, die gleichermaßen als Bausteine bei der Entwicklung eigener verteilter Systeme und für den Betrieb vernetzter Anwendungen und IT-Infrastrukturen relevant sind. müssen. Neben der Absicherung einzelner Verbindungen und Anwendungen ist es insbesondere für den IT-Service-Betrieb größerer Organisationen wichtig, die Sicherheit der Netzwerkkommunikation in der Breite überprüfen und verwalten zu können. Wie sich sichere Kommunikationsprotokolle darauf auswirken, vertiefen Sie in Kapitel 5. Abschließend verschaffen Sie sich in Kapitel 6 eine Übersicht über Ansätze zur verschleierten Kommunikation sowohl im Internet als auch in organisationsinternen Netzen und lernen, ihre Vorteile und Grenzen realistisch einzuschätzen.

Beim Bearbeiten dieses Studienhefts wünschen wir Ihnen viel Spaß und Erfolg.

Ihre Studienleitung

1 Grundlagen sicherer Netzwerkkommunikation

Nach Bearbeitung dieses Kapitels können Sie die Eigenschaften und Ziele, die eine sichere Netzwerkkommunikation charakterisieren, beschreiben und für beliebige Angriffe beurteilen, welche Auswirkungen sie auf die definierten Ziele haben. Sie beherrschen die Grundbegriffe der Informationssicherheit im Kontext der Netzwerkkommunikation und können diese präzise voneinander abgrenzen. Sie wissen, wie Abläufe in Kommunikationsprotokollen bei der Betrachtung von Angriffen modelliert werden können und wie Angreifermodelle beschrieben werden, um angemessene Gegenmaßnahmen planen zu können. Sie kennen die wesentlichen Eckpunkte der OSI Security Architecture und beherrschen eine Strukturierungsmethode, um die Vielzahl heute verfügbarer Sicherheitsmaßnahmen systematisch zu kategorisieren.

1.1 Ziele der sicheren Netzwerkkommunikation

Wenn Sie einen Rechner Ihrer Wahl, z. B. einen PC, ein Notebook oder ein Tablet, vor sich auf dem Tisch liegen haben, können Sie einige grundlegende Eigenschaften durch einfache Messungen erfassen und quantifizieren: Welche Abmessungen und welches Gewicht hat er? Wie warm und wie laut wird er im Dauerbetrieb, z. B. beim Abspielen eines Films? Auch softwareseitige Eigenschaften lassen sich recht einfach ermitteln: Wie viele Anwendungen sind installiert? Wie viel Speicherplatz ist aktuell belegt bzw. noch frei?

Wie aber würden Sie die Informationssicherheit des Geräts beurteilen und mit anderen Systemen vergleichen? Ist ein Notebook, auf dem als Betriebssystem Microsoft Windows installiert ist, sicherer oder unsicherer als ein Server, auf dem Linux läuft? Was ändert sich an Ihrer Einschätzung, wenn Sie erfahren, dass auf einem der beiden Geräte seit zwei Jahren keine Software-Updates mehr installiert wurden?

Dieses kleine Gedankenspiel zeigt ein fundamentales Problem, vor dem die Informationssicherheit als Disziplin immer noch steht: Die Sicherheit von Systemen kann nicht direkt gemessen werden – anders als u. a. für Gewichte und Temperaturen gibt es noch gar keine standardisierte Maßeinheit dafür. Auch Ansätze für Sicherheitskennzahlen (engl. *security metrics*), wie sie im IT-Sicherheitsmanagement verwendet werden, drücken immer nur punktuelle Teilaspekte in Zahlen aus und sind bislang nur sehr eingeschränkt system- und organisationsübergreifend nutzbar.

Aus diesem Grund wird Informationssicherheit üblicherweise durch einander ergänzende Eigenschaften oder Zielsetzungen charakterisiert, also *qualitativ* statt *quantitativ* beschrieben. Diese Eigenschaften sind aber je nach Anwendungsfall nicht immer gleich wichtig. Ihre gewünschte Ausprägung hängt vom jeweils individuellen Schutzbedarf der verarbeiteten Daten ab. Sicher haben Sie die drei grundlegenden Ziele *Vertraulichkeit*, *Integrität* und *Verfügbarkeit* im Laufe des Studiums bereits kennengelernt; wir wollen im Folgenden rekapitulieren, was sie im Kontext der Netzwerkkommunikation konkret bedeuten.

1.1.1 Grundziele: Vertraulichkeit, Integrität und Verfügbarkeit

Allen heute in der Praxis verbreitet eingesetzten Rechnernetzen, z. B. also organisationsinternen kabelgebundenen lokalen Netzen, WLAN-Netzen und dem Internet, liegt das Prinzip der Paketvermittlung zugrunde: Beliebig lange Nachrichten, die sogenannten Nutzdaten, werden in relativ kleine einzelne Datenpakete zerlegt; diese werden mit Metadaten wie der Absender- bzw. Empfängeradresse versehen und über ein Übertragungsmedium, z. B. eine Glasfaserleitung oder ein Kupferkabel, losgeschickt. Falls der designierte Empfänger der Nachricht nicht direkt an dieses Übertragungsmedium angeschlossen ist, kommen sogenannte Transitsysteme, i. d. R. aktive Netzkomponenten wie Router, zum Einsatz. Sie haben die Aufgabe, eingehende Datenpakete auf ein anhand der Empfängeradresse gewähltes anderes Übertragungsmedium weiterzureichen, bis es den Empfänger erreicht oder feststeht, dass eine Fehlersituation vorliegt. Die Zustellung eines Datenpakets über das Internet kann deshalb oft über rund ein Dutzend solcher Zwischenstationen führen.

Jede Datenübertragung über Rechnernetze kann mit technischen Mitteln abgehört werden: Der Datenverkehr auf Glasfaser- und Kupferkabeln kann mit TAPs (Test Access Points) kopiert werden. Zusätzliche lokale Empfänger können WLAN-Signale aufgreifen. Letztlich erlauben sogenannte SPAN- oder Mirror-Ports an Netzkomponenten das Kopieren der diesen Zwischenstationen eigentlich nur zur Weiterleitung übergebenen Datenpakete. Wir müssen also immer damit rechnen, dass die über ein Rechnernetz übertragenen Daten nahezu beliebig von Dritten mitgelesen werden. Dies gilt erst recht für Store-and-Forward-Protokolle wie den Versand von E-Mails, bei denen die Daten zusätzlich explizit auf beteiligten Servern zwischengespeichert werden.

Beispiel 1.1:

Abb. 1.1 zeigt dieses Problem anhand einer E-Mail, die Alice an Bob schickt: Jeder, der Zugriff auf eines der beteiligten Übertragungsmedien, eine der beteiligten Netzkomponenten oder einen der beiden Mailserver hat, kann den E-Mail-Inhalt mitlesen.

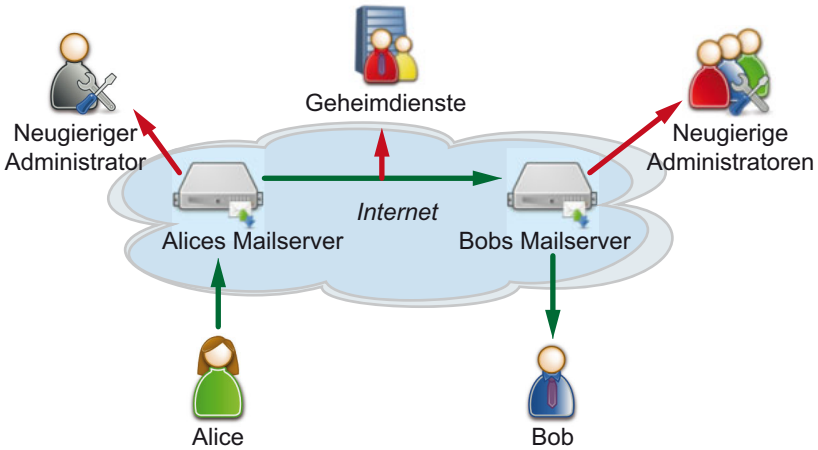

Abb. 1.1: Verletzung der Vertraulichkeit durch Abhören am Beispiel E-Mail

Eine naheliegende Eigenschaft *sicherer* Netzwerkkommunikation ist deshalb, dass nur der gewünschte Empfänger die übermittelten Nutzdaten verwenden kann.

Definition 1.1: Vertraulichkeit

Vertraulichkeit (engl. *confidentiality*) ist gewährleistet, wenn geschützte Daten nur von Berechtigten genutzt werden können.

Die Vertraulichkeit wird meist durch Datenverschlüsselung erreicht: Ein Dritter kann die verschlüsselt übertragenen Daten zwar abhören, aber nicht sinnvoll weiterverarbeiten, wenn er sie nicht korrekt entschlüsseln kann.

Übung 1.1:

Überlegen Sie sich für verschiedene Webseiten und -anwendungen sowie andere internetbasierte Dienste, die Sie regelmäßig nutzen (z.B. Nachrichten, Verkehrsmeldungen, Online-Banking, Instant Messaging, soziale Netzwerke...), ob und in welchem Grad Sie als Benutzer Vertraulichkeit bei der Datenübertragung erwarten.

Die Verschlüsselung einer Nachricht ändert nichts daran, dass Dritte weiterhin beobachten können, wer mit wem kommuniziert. Auch weitere Metadaten, beispielsweise die Zeitpunkte und der Umfang der einzelnen Kommunikationsvorgänge, bleiben erkennbar und lassen gewisse Rückschlüsse zu. Es können also weiterhin sogenannte Verkehrsflussanalysen durchgeführt werden.

Verschlüsselung schützt nur die Vertraulichkeit der übertragenen *Nutzdaten*.

Die *Metadaten* der Kommunikation bleiben erhalten!

Verschlüsselung beugt also zumindest dem unerwünschten Mitlesen der Nachrichteninhalte durch Dritte vor. Ein Angreifer, der z.B. eine Glasfaserleitung oder einen Router unter seiner Kontrolle hat, könnte aber auch aktiv in die Kommunikation eingreifen und einzelne Datenpakete entweder manipulieren oder gar nicht zum Empfänger durchlassen. Er könnte auch eigene Datenpakete mit gefälschten Absenderadressen auf die Leitung geben (engl. *spoofing*) oder Datenpakete, die er früher mitgelesen hat, erneut einspielen (engl. *replay attack*).

Beispiel 1.2:

Ein typischer Angriff ist in Abb. 1.2 dargestellt: Gelingt es einem Angreifer unbemerkt, das Datenfeld „Ziel-Kontonummer“ der Online-Überweisung auf einen von ihm gewählten Wert abzuändern, wird Alice sich wundern, wenn Bob sich irgendwann beschwert, dass das Geld nie bei ihm angekommen ist.

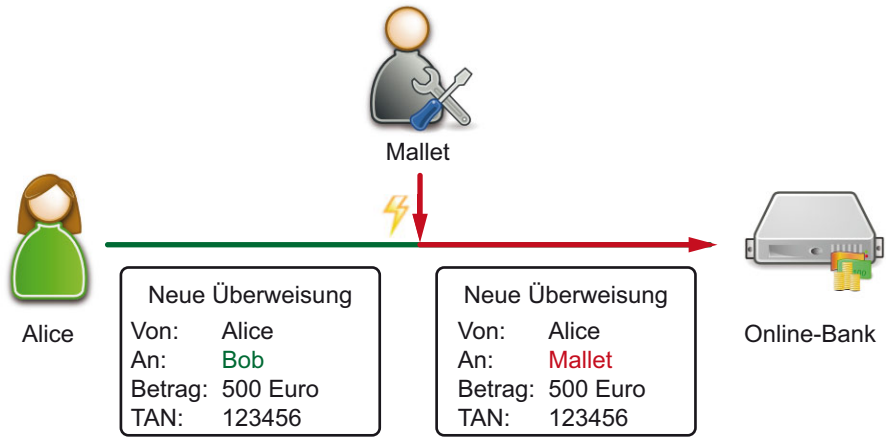

Abb. 1.2: Verletzung der Integrität einer Nachricht durch Datenmanipulation

Sichere Netzwerkkommunikation bedeutet entsprechend auch, dass derartige Manipulationen erkannt werden können.

Definition 1.2: Integrität

Integrität (engl. *integrity*) ist gewährleistet, wenn geschützte Daten nicht unautorisiert und unbemerkt modifiziert werden können.

Im Zusammenspiel mit der Vertraulichkeit der Kommunikation ist wichtig, zu verstehen, dass eine Verschlüsselung nicht automatisch vor einer Manipulation der Nutzdaten schützt: Selbst an verschlüsselten Daten kann der Angreifer „blind“ Veränderungen vornehmen, die der Empfänger nicht erkennen kann, falls aus der Entschlüsselung noch halbwegs plausible Daten resultieren.

Verschlüsselung allein bietet keinen Schutz vor Datenmanipulation!

Als typische Sicherheitsmaßnahme kommen deshalb sogenannte kryptografische Prüfsummen zum Einsatz. Wie Sie in Kapitel 2 sehen werden, haben diese zwar eine andere Zielsetzung als Verfahren zur Verschlüsselung, werden aber oftmals gemeinsam mit diesen eingesetzt.

Übung 1.2:

Auch in vielen anderen Bereichen werden Prüfsummen eingesetzt, z. B. zum Erkennen von Zifferndrehern bei Kontonummern oder zum Erkennen von technischen Übertragungsfehlern bei IP-Paketen. Überlegen Sie sich, warum ein einfaches Mitschicken von Prüfsummen zusammen mit den Nutzdaten-Paketen nicht ausreichend ist, wenn ein Angreifer die Nutzdaten manipulieren möchte.

Die dritte grundlegende Eigenschaft sicherer Netzwerkkommunikation ist die Verfügbarkeit der benötigten Dienste und Daten. Wahrscheinlich haben Sie sich schon einmal geärgert, dass ein Server oder Dienst im Internet wegen eines Hardwaredefekts, Wartungsarbeiten an der Software oder Fehlkonfiguration vorübergehend nicht nutzbar war.

Angreifer können aber auch versuchen, Dienste so anzugreifen, dass sie für ihre legitimen Anwender nicht mehr zugänglich sind.

Beispiel 1.3:

Ein auch in der Praxis häufig vorkommendes Beispiel sind Distributed-Denial-of-Service-(DDoS-)Angriffe wie in Abb. 1.3 dargestellt: Die Angreifer haben eine größere Anzahl von vernetzten Endgeräten unter ihrer Kontrolle, z.B. weil diese mit einer Malware infiziert oder durch eine Sicherheitslücke von außen kompromittiert wurden. Indem sie jede Menge Datenmüll an den Server schicken, überlasten sie entweder dessen Ressourcen oder den ganzen Internet-Uplink der betroffenen Organisation. Für die Bearbeitung der Zugriffe der anderen Anwender stehen dann keine Ressourcen mehr zur Verfügung.

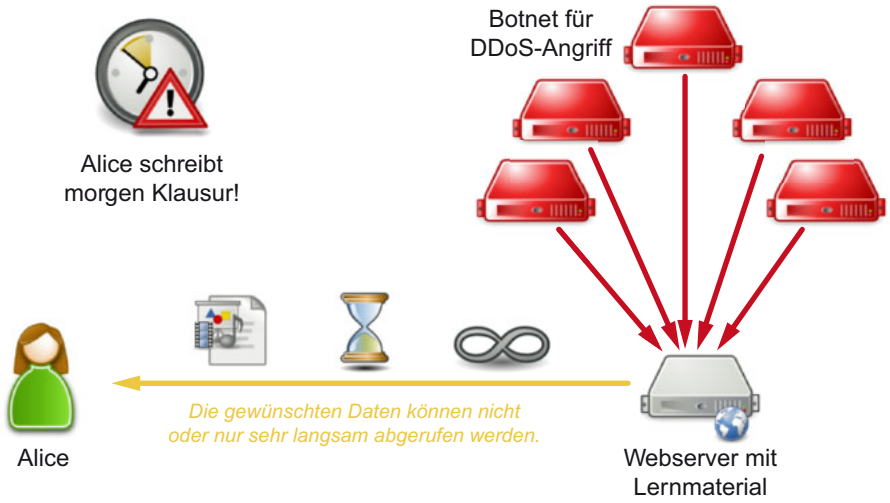

Abb. 1.3: Verletzung der Verfügbarkeit durch einen DDoS-Angriff

Die *sichere* Netzwerkkommunikation hat deshalb auch die Verfügbarkeit als Ziel:

Definition 1.3: Verfügbarkeit

Verfügbarkeit (engl. *availability*) ist gewährleistet, wenn autorisierte Subjekte störungsfrei ihre Berechtigungen wahrnehmen können.

Das **Gesamtziel** der Informationssicherheit wird gerne mit den Anfangsbuchstaben der englischen Bezeichnungen ihrer Teilziele abgekürzt:

CIA = Confidentiality + Integrity + Availability

Übung 1.3:

Überlegen Sie sich einige Beispiele für IT-Dienste, bei denen der Bedarf an *Vertraulichkeit*, *Integrität* und *Verfügbarkeit* unterschiedlich ausgeprägt ist.

1.1.2 Erweiterte Schutzziele

Letztlich wirken sich alle erfolgreichen Angriffe auf mindestens eines der Grundziele Vertraulichkeit, Integrität oder Verfügbarkeit aus. Um eine feiner granulierte Diskussion führen zu können, werden aber häufig auch aus den Grundzielen abgeleitete, sogenannte erweiterte Schutzziele betrachtet. Im Kontext sicherer Netzwerkkommunikation sind dabei die folgenden besonders relevant:

- Authentizität (engl. *authenticity*): Die Echtheit einer empfangenen Nachricht kann geprüft werden; insbesondere wird sichergestellt, dass der angegebene Absender mit dem tatsächlichen Absender der Nachricht übereinstimmt.
- Revisionsfähigkeit (engl. *auditability*): Alle relevanten Verarbeitungsvorgänge können während und nach ihrer Durchführung lückenlos nachvollzogen werden.
- Verbindlichkeit (engl. *non-repudiation*): Sowohl der Absender als auch der Empfänger einer Nachricht können im Nachhinein nicht unwiderlegbar abstreiten, dass sie die Nachricht verschickt bzw. entgegengenommen haben.

1.2 Beschreibung und Modellierung von Angriffen

Bei der Diskussion der Schutzziele haben wir den Begriff *Angriff* bereits intuitiv verwendet.

Wir betrachten ihn nun genauer und Sie lernen dann Notations- und Modellierungsmethoden kennen, auf die wir im weiteren Verlauf bei der Betrachtung konkreter Angriffe und Sicherheitsmaßnahmen immer wieder zurückkommen werden.

1.2.1 Einordnung von Angriffen

Der Ausgangspunkt für die Präzisierung des Begriffs „Angriff“ sind Schwachstellen, die man begrifflich präzise von Verwundbarkeiten abgrenzen muss.

Definition 1.4: Schwachstellen und Verwundbarkeiten

Eine Schwachstelle (engl. *weakness*) ist ein punktueller Fehler eines Systems, der noch nicht beseitigt wurde. Eine Verwundbarkeit (engl. *vulnerability*) ist eine solche Schwachstelle, die zur Verletzung mindestens eines der Grundziele der Informationssicherheit führen kann.

Eine Verwundbarkeit ist somit ein latenter Zustand, der mit einer Bedrohung (engl. *threat*) verbunden ist: Ihre Ausnutzung durch einen Angreifer würde zum Verfehlen eines oder mehrerer Schutzziele fehlen. Angriffe stellen deshalb konkrete Instanzen einer solchen Bedrohung dar:

Definition 1.5: Angriff

Ein Angriff (engl. *attack*) ist im engeren Sinn die versuchte Ausnutzung einer bekannten oder vermuteten Verwundbarkeit. Im weiteren Sinn werden auch alle Maßnahmen zur gezielten Vorbereitung dieses Versuchs zum Angriff gezählt.

Idealerweise würde man erfolgreiche Angriffe dadurch verhindern, dass von vornherein alle Schwachstellen vermieden oder beseitigt werden. Praktisch ist es aber unmöglich, komplexe Systeme mit begrenzten Ressourcen absolut fehlerfrei zu konzipieren und zu implementieren. Deshalb müssen Sicherheitsmaßnahmen ergriffen werden, die wir in Abschnitt 1.4 vertiefen.

1.2.2 Ablaufnotation und Angreifermodelle

In den Beispielen zu den Grundzielen ist Ihnen vielleicht schon aufgefallen, dass die Protagonisten immer die Namen Alice und Bob hatten. Das war kein Zufall, weil sich die Verwendung dieser Namen in der Literatur über Kryptografie und Informationssicherheit seit Langem eingebürgert hat. Wir orientieren uns daran und verwenden im Folgenden immer diese Namen, wenn damit die „Guten“ gemeint sind:

- Alice ist die Person, die den beschriebenen Ablauf anstößt, also der Initiator eines Kommunikationsprotokolls.
- Bob nimmt die Nachricht von Alice entgegen, verarbeitet sie entsprechend und antwortet ggf. darauf.
- Carol und Dave sind bei Bedarf weitere gutartige Teilnehmer.
- Trent ist ein vertrauenswürdiger Dritter (engl. *trusted third party*).

Demgegenüber gibt es auch typische Namen für die „Bösen“, also die Angreifer in einem Szenario:

- Eve ist ein passiver, d. h. die Netzkommunikation lediglich abhörender Angreifer (von engl. *eavesdropper*).
- Mallet und Mallory sind aktive Angreifer, die also beispielsweise Datenpakete während der Übertragung manipulieren können.

Wie Abb. 1.4 mit einem einfachen Beispiel zeigt, kann der Ablauf eines Angriffs u. a. als UML-Sequenzdiagramm modelliert werden: Eve hört die Kommunikation zwischen Alice und Bob ab. Wichtig ist, hierbei zu verstehen, dass Alice und die anderen Akteure nicht nur natürliche Personen repräsentieren können, sondern beliebige Beteiligte an einer Kommunikation. Im Beispiel könnte Alice also einen clientseitigen Webbrowser darstellen, wohingegen Bob für einen Webserver und Eve für einen Geheimdienst stehen. Die Idee dahinter ist, dass man den grundsätzlichen Ablauf einheitlich beschreiben kann, auch wenn er beliebig viele konkrete Ausprägungen in der Praxis haben kann.

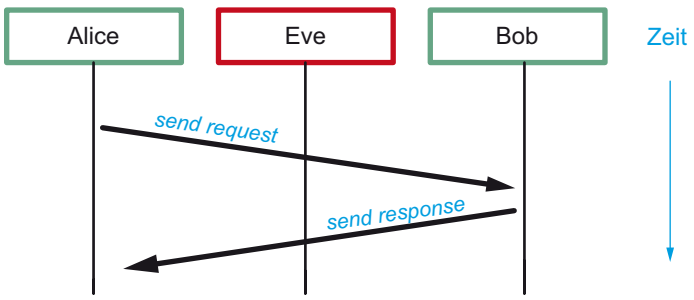

Abb. 1.4: Modellierung eines Angriffs auf ein Kommunikationsprotokoll in UML

Ein wichtiger Schritt bei der Analyse möglicher Angriffe und geeigneter Sicherheitsmaßnahmen ist die realistische Beschreibung von Angreifern. Zu einem solchen Angreifermodell gehören insbesondere Angaben

- zur Position des Angreifers: Handelt es sich um eigenes Personal, also Innentäter, oder Besucher bzw. Einbrecher? Oder wird der Angriff von außen über das Internet durchgeführt?
- zu Fähigkeiten des Angreifers: Hierzu zählen insbesondere sein Wissen und seine finanziellen Möglichkeiten. Typischerweise gibt es diesbezüglich Unterschiede z. B. zwischen experimentierfreudigen Jugendlichen, Fachleuten mit praktischer Erfahrung und erfahrenen Industriespionen oder Geheimdiensten.
- zur Motivation und Zielsetzung des Angreifers: Die früher verbreiteten Motivationen Spieltrieb, Geltungsbedürfnis und Vandalismus werden zunehmend durch überwiegend finanzielle Anreize abgelöst. Auch politische Ansichten, religiöser Fanatismus und vermeintlicher Patriotismus stehen hinter vielen Angriffen.
- zu spezifischen Charakteristika der Angriffe, mindestens also die Unterscheidung zwischen passiven und aktiven Angriffen.

Übung 1.4:

Informieren Sie sich im Internet über die *Operation Payback*, mit der Internet-Aktivisten 2010 für das Lahmlegen mehrerer Websites weltweit bekannt wurden. Beschreiben Sie anhand der o. g. Aspekte das zugehörige Angreifermodell.

1.3 OS Security Architecture

In den bisherigen Abschnitten haben wir uns mit grundlegenden Begriffen, insbesondere zur Präzisierung der Zielsetzung und Beschreibung von Angriffen, auseinandergesetzt. Was ist nun aber konkret zu unternehmen, um zu einer sicheren Netzwerkkommunikation zu gelangen? Einen ersten Anhaltspunkt dazu liefern Standards, die bewährte Good Practices strukturiert bündeln.

Aus dem Gebiet der Rechnernetze ist Ihnen sicher noch das ISO/OSI-Schichtenmodell vertraut, das die Implementierung von Netzwerkkommunikation auf sieben Schichten – von der physikalischen Schicht 1 bis zur Anwendungsschicht 7 – beschreibt. Seine Praxisbedeutung ist aufgrund fehlender Implementierungen und der Dominanz des Internet- bzw. TCP/IP-Modells bekanntlich vernachlässigbar, aber seine Systematik führt dazu, dass es insbesondere in der Ausbildung und Lehre weiterhin erfolgreich eingesetzt wird.

Mit der OSI Security Architecture verhält es sich ganz ähnlich. Sie ist ein international standardisiertes Referenzmodell, das auf die Sicherheitsanforderungen verteilter bzw. vernetzter Systeme ausgelegt ist; es geht also nicht auf die Sicherheit einzelner Maschinen oder Betriebssysteme, sondern ausschließlich der Netzwerkkommunikation ein. In seinem Kern betrachtet das Referenzmodell zum einen fünf Klassen sogenannter Sicherheitsdienste und zum anderen Sicherheitsmechanismen, die zur Umsetzung der Sicherheitsdienste herangezogen werden können.

Die Sicherheitsdienste der OSI Security Architecture umfassen:

a) Authentisierungsdienste, mit denen sowohl Kommunikationspartner als auch der Ursprung übertragener Daten zweifelsfrei identifiziert werden.

b) Zugriffskontrolldienste, die einen unautorisierten Zugriff auf Ressourcen unterbinden.

c) Vertraulichkeitsdienste, die eine Offenlegung von Kommunikationsinhalten und Verbindungsmetadaten gegenüber Dritten unterbinden.

d) Integritätsdienste, die vor der Modifikation, Einfügung, Löschung, Umordnung, Duplikation und Wiedereinspielung von Daten schützen.

e) Verbindlichkeitsdienste, die das Absenden bzw. das Empfangen von Daten belegen, sodass der jeweilige Vorgang nicht geleugnet werden kann.

Bei den Sicherheitsmechanismen unterscheidet die OSI Security Architecture zwischen *spezifischen* Mechanismen, die Sicherheitsdienste umzusetzen, und *durchgängigen* Mechanismen, die als Schnittstellen zum IT-Sicherheitsmanagement fungieren:

- Spezifische Sicherheitsmechanismen:

  a) Verschlüsselung zur Umsetzung der Vertraulichkeitsdienste. Entsprechende Verfahren rekapitulieren Sie in Kapitel 2.

  b) Digitale Signaturen werden für die Verbindlichkeitsdienste benötigt. Eine praktische Anwendung betrachten wir in Abschnitt 2.3.

  c) Zugriffskontrollmechanismen unterstützen die Zugriffskontrolldienste; sie bilden Verfahren wie Zugriffskontrolllisten ab, die u.a. von Dateisystemen und Betriebssystemen bekannt sind.

  d) Datenintegritätsmechanismen setzen die Integritätsdienste um; die zum Einsatz kommenden Hashverfahren wiederholen Sie in Kapitel 2.

  e) Authentifikationsmechanismen werden von den Authentisierungsdiensten eingesetzt; mit verschiedenen Varianten befassen wir uns in Abschnitt 2.2.

  f) Traffic Padding dient der Verschleierung von Verbindungsmetadaten. Moderne Ansätze dazu diskutieren wir in Kapitel 6.

  g) Wegewahlmechanismen beeinflussen Routingentscheidungen, also über welche Zwischenstationen ein Datenpaket vom Absender zum Empfänger übertragen wird. In der Praxis haben Anwender, Endgeräte und Anwendungen darauf i.d.R. keinen Einfluss und müssen sich mit den Einstellungen, die die Betreiber von Routern vornehmen, arrangieren. Mit zunehmender Verbreitung moderner Betriebsparadigmen wie Software-defined Networking (SDN) ist jedoch absehbar, dass entsprechende technische Schnittstellen dafür geschaffen werden.

  h) Notariatsmechanismen sichern Eigenschaften wie Integrität und Erzeugungsdatum von Datenpaketen zu; sie stellen im Wesentlichen eine Kombination aus Datenintegritäts- und Signaturmechanismen dar.

- Durchgängige Sicherheitsmechanismen:

  a) Vertrauenswürdige Funktionalität erbringt den Nachweis, dass ein Sicherheitsdienst nicht bösartig verändert wurde; zur Realisierung kommen z.B. Trusted Platform Modules (TPM-Chips) zum Einsatz.

b) Sicherheitsklassifikation ermöglicht die Kennzeichnung (engl. *labeling*) von Daten und Ressourcen, um beispielsweise Zugriffskontrollmechanismen umsetzen zu können.

c) Ereignisverwaltung dient der Protokollierung z. B. von fehlgeschlagenen oder erfolgreichen Login-Versuchen von Anwendern bei Diensten.

d) Auditing ist eng mit der Ereignisverwaltung verknüpft und prüft beispielsweise die aktuelle Konfiguration und protokollierte Ereignisse darauf, dass sie bestimmte Vorgaben erfüllt.

e) Recovery muss das System nach einem erkannten Sicherheitsproblem wieder in einen konsistenten Zustand überführen.

Der Standard legt zudem die Beziehungen zwischen den einzelnen Sicherheitsmechanismen und Diensten dar und gibt Empfehlungen, wie die Dienste auf den verschiedenen Schichten des ISO/OSI-Referenzmodells eingesetzt werden können; bei den in Kapitel 3 behandelten Protokollen werden wir darauf ebenfalls eingehen.

1.4 Kategorisierung von Sicherheitsmaßnahmen

In der Praxis kommt es immer wieder vor, dass Organisationen erfolgreich angegriffen werden, obwohl sie Sicherheitsmaßnahmen auf dem aktuellen Stand der Technik betreiben. Beispielsweise könnte ein Angreifer eine Sicherheitslücke ausnutzen, die erst vor Kurzem bekannt geworden ist und für die noch kein Security-Update verfügbar ist; oder ein Mitarbeiter wird Opfer einer gut gemachten Phishing-E-Mail, sodass dem Angreifer Zugangsdaten in die Hände fallen, mit denen er auf sensible Daten zugreifen kann. Wichtig sind deshalb zwei Erkenntnisse:

a) Präventive Sicherheitsmaßnahmen allein reichen nicht aus. Mindestens genauso wichtig sind Sicherheitsmaßnahmen, mit denen erfolgreiche Angriffe zeitnah erkannt und mit denen geeignet darauf reagiert werden kann.

b) Technische Sicherheitsmaßnahmen allein reichen nicht aus. Es gibt in der Praxis keinen perfekten technischen Schutz vor Angriffen. Der Faktor Mensch ist in der Informationssicherheit mindestens genauso wichtig.

Für jedes Sicherheitsproblem, vor dem man sich schützen möchte, muss deshalb eine Kombination aus Sicherheitsmaßnahmen ergriffen werden, durch die der aus den obigen Überlegungen resultierende Problemraum möglichst vollständig abgedeckt wird.

Beispiel 1.4:

Abb. 1.5 zeigt dies anhand des Beispiels, dass der PC eines Mitarbeiters beim Surfen im Web mit Schadsoftware infiziert werden könnte.

- Die technischen Aspekte werden durch eine aktuelle Antivirus-Software abgedeckt:
  - Prävention: Ein auf dem PC laufender Virenscanner schlägt beim Download einer bereits bekannten Schadsoftware Alarm und verhindert deren Ausführung.

  - Detektion: Eine ganz neue Schadsoftware ist beim Download möglicherweise noch nicht bekannt. Werden nach einem Antivirus-Signaturupdate aber die lokalen Datenträger regelmäßig erneut überprüft, kann eine bereits heruntergeladene Schadsoftware erkannt werden.
  - Reaktion: Schadsoftwareverseuchte Dateien werden gelöscht oder in einen Quarantänebereich verschoben, in dem sie der Benutzer nicht ausführen kann.

- Organisatorische Maßnahmen umfassen häufig Schulungen und die Festlegung von Abläufen im Ernstfall:
  - Prävention: Die gesamte Belegschaft kann geschult werden, bei der Internetnutzung am Arbeitsplatz vorsichtig zu agieren und keine Dateien aus unbekannten Quellen herunterzuladen.
  - Detektion: Analog dazu können Hinweise gegeben werden, bei welchen Auffälligkeiten ihrer PCs die Mitarbeiter/innen von einer möglichen Schadsoftware-Infektion ausgehen sollten.
  - Reaktion: Für den Fall, dass ein infizierter Rechner vermutet wird, kann z. B. festgelegt werden, dass sich die betroffene Person schnellstmöglich an den internen Service Desk oder einen Ansprechpartner in der IT-Abteilung wenden soll.

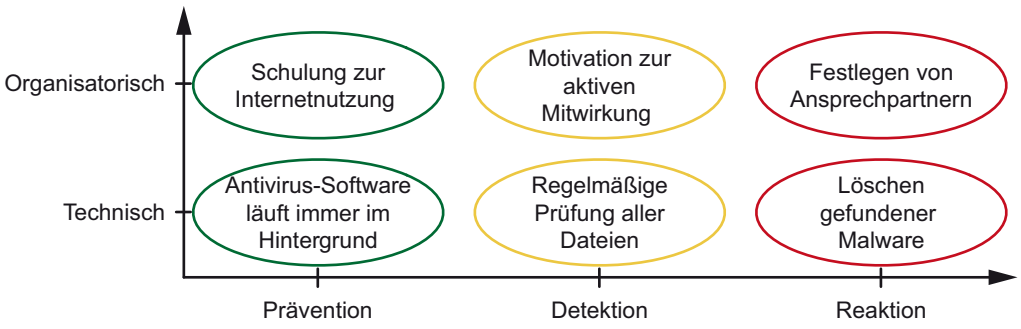

Abb. 1.5: Kategorisierung von Sicherheitsmaßnahmen am Beispiel Malware-Schutz

Natürlich könnte man sich für jeden der Teilbereiche auch viele andere Sicherheitsmaßnahmen überlegen; wesentlich ist, keinen der Aspekte zu vergessen oder zu vernachlässigen.

Übung 1.5:

In welche Bereiche würden Sie folgende typische Sicherheitsmaßnahmen einordnen?

- Planung eines Prozesses zum Wiederherstellen von Serverkonfigurationen aus Backups
- Firewall
- Automatisierte Auswertung von Webserver-Logfiles

Zusammenfassung

In diesem Kapitel haben wir zunächst die Grundziele der Informationssicherheit – Vertraulichkeit, Integrität und Verfügbarkeit – mit ihrem Bezug zur sicheren Netzwerkkommunikation kennengelernt. Dabei hat sich gezeigt, dass jedes Ziel durch Angriffe gefährdet werden kann, bei denen sogenannte Verwundbarkeiten – also sicherheitsrelevante Schwachstellen – ausgenutzt werden. Angriffe auf Kommunikationsvorgänge können einheitlich mit Protagonisten wie Alice, Bob und Mallet beschrieben werden, wobei die Fähigkeiten, die Mallet als böser Angreifer hat, in einem Angreifermodell dargelegt werden können.

Sie haben gesehen, dass Sicherheitsmaßnahmen erforderlich sind, um mit Angriffen umzugehen. Die OSI Security Architecture hat uns als Referenzmodell erste Anhaltspunkte geliefert, welche Maßnahmen zur Absicherung der Netzwerkkommunikation im Regelfall benötigt werden. Bei der Auswahl von Sicherheitsmaßnahmen können wir uns gut am Lebenszyklus erfolgreicher Angriffe orientieren und müssen darauf achten, sowohl präventive als auch detektierende und reagierende Maßnahmen vorzusehen. Um den Faktor Mensch in der IT-Sicherheit nicht zu vernachlässigen, sind neben technischen immer auch organisatorische Sicherheitsmaßnahmen umzusetzen.

Aufgaben zur Selbstüberprüfung

1.1 Erstellen Sie ein Angreifermodell zu dem in Abb. 1.1 gezeigten Angriff auf die Vertraulichkeit von E-Mails.

1.2 Überlegen Sie sich den Schutzbedarf, also die begründete Ausprägung der Anforderungen Vertraulichkeit, Integrität und Verfügbarkeit z.B. in den drei Stufen *niedrig*, *mittel* oder *hoch* für

- den Zugriff auf das Personalverwaltungssystem eines kleinen mittelständischen Unternehmens,
- einen Webserver, auf dem Hobbyastronomen Fotos des Sternenhimmels veröffentlichen können, und
- einen Webserver, über den aktuelle Börsenkurse abgerufen werden können!

1.3 Ordnen Sie *Verschlüsselung* in die Matrix aus
{*präventiven, detektierenden, reagierenden*} × {*technischen, organisatorischen*}
Sicherheitsmaßnahmen ein und begründen Sie, warum sie als alleinige Sicherheitsmaßnahme nicht ausreichend ist.

2 Kryptografische Grundlagen für sichere Netzwerkkommunikation

Wenn Sie dieses Kapitel bearbeitet haben, haben Sie Ihr Wissen über kryptografische Verfahren, insbesondere Chiffren und Hashfunktionen, aufgefrischt. Sie können kryptografische Ansätze wie Verschlüsselung, Prüfsummenbildung und Schlüsselmanagement voneinander abgrenzen und kennen wichtige Verfahren aus jedem dieser Bereiche. Sie können essentielle Eigenschaften wie Perfect Forward Secrecy beschreiben und kennen erste Beispiele dafür, wie verschiedene kryptografische Ansätze miteinander kombiniert werden können, um mehrere Teilziele der Informationssicherheit zu erreichen.

2.1 Rekapitulation der Grundlagen der Kryptografie

Für das Verständnis sicherer Kommunikationsprotokolle benötigen Sie Grundlagen aus der Kryptografie, die Sie im Lauf Ihres Studiums bereits kennengelernt haben. Bitte rekapitulieren Sie vor dem Weiterlesen unbedingt die folgenden Begriffe und Themen:

- Abgrenzung der Grundbegriffe Kryptologie, Kryptografie, Kryptoanalyse und Steganografie.
- Prinzipielle Funktionsweise symmetrischer Chiffren. Begriffe und Verfahren wie Substitution, Permutation, Feistelchiffre, S-Box, Konfusion, Diffusion und Lawineneffekt sowie das Prinzip von Kerckhoffs müssen vertraut sein.
- Funktionsweise bzw. genauerer Ablauf von DES, TripleDES und AES.
- Asymmetrische Kryptografie, insbesondere RSA und Elliptische-Kurven-Kryptografie.
- Vorgehensweise bei hybrider Verschlüsselung.
- Grundlagen kryptografischer Prüfsummen, insbesondere Kollisionsresistenz.
- Funktionsweise konkreter kryptografischer Prüfsummenverfahren, sowohl Merkles Meta-Verfahren als auch Sponge-Funktionen (MD5, SHA-1, SHA-2, SHA-3).

2.2 Verwaltung von Schlüsselmaterial und Public Key Infrastructures

Die meisten der in den nachfolgenden Kapiteln betrachteten Protokolle setzen bekannte Chiffren wie AES bzw. RSA und kryptografische Prüfsummenfunktionen aus der SHA-Reihe ein. Auch wenn verschiedene Alternativen angeboten werden, gestaltet sich die Auswahl einfach: Im Zweifelsfall können Sie mit einem standardisierten Verfahren, zu dem aktuell noch keine Schwachstellen bekannt sind, nichts falsch machen. Schwieriger ist der Umgang mit den verwendeten Schlüsseln, insbesondere wenn es viele Kommunikationspartner gibt. Sehen wir uns dazu einige Varianten an.

2.2.1 Out-of-Band Key Management (Pre-Shared Keys)

Bei einem symmetrischen Verschlüsselungs- oder MAC-Verfahren müssen Absender Alice und Empfänger Bob denselben Schlüssel verwenden; der Schlüssel stellt also ein geteiltes Geheimnis (engl. *shared secret*) dar. Da sie ihn i. d. R. vorab vereinbart haben müssen, spricht man auch von einem Pre-Shared Key (PSK).

Wenn wir davon ausgehen, dass ein Angreifer alle über ein Medium übertragenen Nachrichten abhört, ist es offensichtlich keine gute Idee, den Schlüssel selbst darüber als Nachricht zu verschicken. Wir benötigen also eine Möglichkeit, den Schlüssel anderweitig – auf englisch *out of band* – auszutauschen. Je nach Angreifermodell kann dies beispielsweise über Telefon, Briefpost oder persönliche Treffen erfolgen. Wie aus klassischen Agentenfilmen oder -büchern bekannt ist, mussten Spione früher beispielsweise Code-Bücher mitführen, um aus der Ferne verschlüsselt kommunizieren zu können.

Da Schlüssel regelmäßig gewechselt werden sollten und z. B. ein persönliches Treffen zum Schlüsselaustausch für die meisten Anwendungen völlig unpraktikabel wäre – denken Sie an populäre Webseiten und Milliarden von Zugriffen darauf – sind Pre-Shared Keys fast nur in kleinen Endanwenderszenarien im Einsatz: Wenn Sie z. B. zu Hause einen WLAN-Accesspoint betreiben, ist der Zugriff wahrscheinlich über Verfahren wie WPA2-PSK geschützt: Der Schlüssel muss dann im Accesspoint und jedem autorisierten Endgerät eingetragen werden.

2.2.2 Diffie-Hellman Key Exchange

Das nach seinen Erfindern Whitfield Diffie und Martin Hellman benannte Schlüsselaustauschverfahren aus dem Jahr 1976 hat den praktischen Einsatz von Kryptografie revolutioniert; ohne es wären heutige sichere Kommunikationsprotokolle fast undenkbar. Das Diffie-Hellman-Verfahren erlaubt es zwei Kommunikationspartnern, einen geheimen Schlüssel in Form einer Zahl zu berechnen, obwohl alle zwischen ihnen ausgetauschten Nachrichten vom Angreifer abgehört werden dürfen. Es macht also den Out-of-Band-Schüsselaustausch überflüssig!

In seiner ursprünglichen Fassung läuft der Schlüsselaustausch wie folgt ab:

- Alice und Bob einigen sich auf eine Primzahl p und einen Generator g; dies bedeutet, dass mit $g^n \mod p$ alle Zahlen von 1 bis $p-1$ erzeugt werden können. p und g werden als DH-Gruppe bezeichnet und dürfen veröffentlicht werden.
- Alice wählt zufällig ein x mit $1 \leq x \leq p-1$, das sie geheimhält.
- Bob wählt analog dazu zufällig ein y mit $1 \leq y \leq p-1$, das er geheimhält.
- Alice schickt folgende Nachricht an Bob: $A = g^x \mod p$.
- Bob schickt folgende Nachricht an Alice: $B = g^y \mod p$.
- Beide verwenden dann folgende Zahl als Schlüssel: $A^y = (g^x)^y = g^{xy} = (g^y)^x = B^x$.

Übung 2.1:

Probieren Sie das Verfahren aus für $p = 23$, $g = 5$, $x = 6$ und $y = 15$.

Dem Diskreter-Logarithmus-Problem ist zu verdanken, dass ein Angreifer den Wert des Schlüssels nicht berechnen kann, selbst wenn er die hinreichend groß gewählten Werte von p, g, A und B kennt. In einer an elliptische Kurven angepassten Variante, die als Elliptic Curve Diffie-Hellman (ECDH) bezeichnet wird, ist das Verfahren heute überall im Einsatz, wo z.B. das Transport-Layer-Security-Verfahren (TLS) eingesetzt wird, das wir in Abschnitt 3.5 im Detail kennenlernen werden.

2.2.3 X.509v3-Zertifikate und Public Key Infrastructures

Stellen wir uns vor, dass Alice und Bob noch nie *sicher* miteinander kommuniziert haben. Sie wissen dann ohne Weiteres auch nicht, ob sie voneinander gegenseitig die richtigen Public Keys vorliegen haben. Nehmen wir aber an, dass beide mit Trent einen gemeinsamen Bekannten haben, dem sie vertrauen. Zudem hat Trent den richtigen Key von Alice vorliegen und Bob hat Trents Public Key aus zuverlässiger Quelle. Trent könnte dann eine Nachricht an Bob schicken, in der Alices Public Key enthalten ist; Trent verschlüsselt diese Nachricht mit seinem Private Key, sodass Bob weiß, dass sie wirklich von ihm kommt. Da Bob Trent vertraut, glaubt er ihm auch, dass der enthaltene Key von Alice der richtige ist.

In der Realität übernehmen Certificate Authorities (CAs) die Rolle von Trent: Sie stellen für Personen und Systeme sogenannte Zertifikate nach dem ITU-T-Standard X.509v3 aus. Solche Zertifikate enthalten im Wesentlichen

- den Public Key des Zertifikatsnehmers Alice,
- Metadaten wie z.B. nähere Angaben zu Alice, eine Seriennummer und einen Gültigkeitszeitraum und
- eine digitale Unterschrift der CA, die dadurch erstellt wird, dass die CA eine kryptografische Prüfsumme der übrigen Inhalte berechnet und diese mit ihrem Private Key verschlüsselt im Zertifikat abspeichert.

Zusätzlich stellt die CA eine Certificate Revocation List (CRL) und eine Abfragemöglichkeit über das Online Certificate Status Protocol (OCSP) zur Verfügung, um zu klären, ob ein Zertifikat bereits vor Ablauf seines Gültigkeitszeitraums zurückgerufen wurde, z.B. weil der Private Key von Alice kompromittiert wurde.

Es kann mehrere CAs geben, die typischerweise in einer hierarchischen Struktur zueinander angeordnet sind und sich zum Teil gegenseitig zertifizieren (Cross-Zertifizierung). Zusammen bilden sie eine sogenannte Public Key Infrastruktur (PKI). Eine PKI kann jeder für sich betreiben, beispielsweise mit einer CA an jedem Firmenstandort. Eine besondere Bedeutung kommt der *Global PKI* im Internet zu: Betriebssysteme und Webbrowser werden heutzutage mit denPublic Keys von mehreren Hundert CAs ausgeliefert, um beispielsweise bei HTTPS-Verbindungen überprüfen zu können, ob die Gegenseite ein gültiges Zertifikat vorweisen kann; dies betrachten wir in Abschnitt 4.3 genauer.

Übung 2.2:

Ein großes Problem der Global PKI ist, dass einzelne CAs ihre Vertrauensstellung missbrauchen können oder selbst gehackt werden. Recherchieren Sie nach Vorfällen wie dem DigiNotar-Hack 2011. Welches Problem ergibt sich für die Endanwender?

2.2.4 Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy (PFS) ist eine wichtige Eigenschaft von Schlüsselaustauschverfahren. Bislang haben wir angenommen, dass ein Angreifer nur die Datenübertragung abhören oder manipulieren kann. Nun gehen wir aber davon aus, dass es ihm irgendwann gelingt, den Rechner von Alice oder Bob zu kompromittieren. In dieser Situation erlangt er offensichtlich Kenntnis der aktuell verwendeten und aller zukünftig vereinbarten Schlüssel.

Noch schlimmer, als dies sowieso schon ist, wäre aber, wenn der Angreifer damit auch sämtliche Nachrichten entschlüsseln könnte, die er vor der Kompromittierung mitgehört und aufgezeichnet hat. Die Eigenschaft PFS ist genau dann erfüllt, wenn ihm diese nachträgliche Entschlüsselung nicht gelingen kann. PFS setzt also voraus, dass die verwendeten Schlüssel regelmäßig gewechselt werden und dass diese später nicht mehr rekonstruierbar sind, z. B. weil sie Zufallswerte verwendet haben.

Übung 2.3:

Vergleichen Sie folgende Ansätze im Kontext Perfect Forward Secrecy:

- Alice und Bob verwenden das Diffie-Hellman-Verfahren.
- Alice legt einen zufälligen Schlüssel fest und schickt ihn mit Bobs Public Key verschlüsselt an Bob.

2.3 Anwendungsbeispiel: Signieren und Verschlüsseln von E-Mails

Neben der Absicherung einzelner TCP/IP-Verbindungen mit TLS, auf die wir in Abschnitt 3.5 eingehen, ist das Signieren und Verschlüsseln von E-Mails eines der wichtigsten Anwendungsbeispiele für die in diesem Kapitel behandelten Verfahren.

2.3.1 Allgemeiner Ablauf

Alice möchte Bob eine E-Mail schicken. Dabei sollen sowohl die Vertraulichkeit als auch die Authentizität der E-Mail sichergestellt werden. Es kann also niemand außer Bob die E-Mail lesen und zugleich kann er sich sicher sein, dass sie wirklich von Alice stammt. Alice und Bob haben sich Public-/Private-Key-Schlüsselpaare erstellt und sich geeignet zur Verfügung gestellt; auf diesen Teilaspekt gehen wir noch im nächsten Abschnitt ein.

E-Mails haben zwei hier relevante Eigenschaften: Zum einen sind sie durch ihre Länge und z. B. Dateianhänge viel zu groß, um asymmetrisch verschlüsselt werden zu können. Zum anderen ist die Kommunikation asynchron; man könnte also nicht ohne größere Verzögerungen z. B. einen Diffie-Hellman-Schlüsselaustausch implementieren.

Die Vertraulichkeit wird deshalb mittels hybrider Verschlüsselung sichergestellt:

- Alice erzeugt einen zufälligen Schlüssel für ein symmetrisches Verschlüsselungsverfahren. Damit verschlüsselt sie den E-Mail-Inhalt.
- Zusätzlich verschlüsselt sie den zufällig gewählten Schlüssel mit Bobs Public Key und schickt den resultierenden Geheimtext mit der verschlüsselten E-Mail mit.

Auch die Authentizität wird sichergestellt, indem asymmetrische Verschlüsselung in Kombination mit einem anderen Verfahren eingesetzt wird:

- Alice berechnet eine kryptografische Prüfsumme der (verschlüsselten) E-Mail.
- Die Prüfsumme verschlüsselt sie mit ihrem Private Key.
- Den resultierenden Geheimtext schickt sie mit der (verschlüsselten) E-Mail mit.

Auch bei anderen Vorgängen stellt sich häufig die Frage, ob die Prüfsumme über den Klartext oder den Geheimtext gebildet werden sollte. Hier haben kryptoanalytische Ansätze wie der Angriff von Vaudenay und die Praxis gezeigt, dass ein Empfänger immer erst die Prüfsumme verifizieren können sollte, bevor er eine Nachricht entschlüsseln muss. Empfehlenswert ist auf Absenderseite also nur der Ansatz „Encrypt, then hash“.

Bob geht nach dem Empfangen der E-Mail wie folgt vor:

- Wenn er feststellt, dass die E-Mail signiert ist, entschlüsselt er die von Alice mitgeschickte Prüfsumme mithilfe ihres Public Keys. Dann berechnet er die Prüfsumme über die (verschlüsselte) E-Mail selbst und vergleicht die beiden Prüfsummen. Wenn sie identisch sind, ist die E-Mail authentisch.
- Das von Alice zufällig gewählte Passwort kann Bob mit seinem Private Key entschlüsseln. Damit kann er dann den eigentlichen E-Mail-Inhalt mit der symmetrischen Chiffre entschlüsseln.

Abb. 2.1 fasst diesen Ablauf zusammen.

Größere Mengen an Nutzdaten werden praktisch immer *symmetrisch* verschlüsselt, auch wenn die Benutzer wie im Fall von E-Mail-Clients nur mit ihrem *asymmetrischen* Schlüsselpaar in Berührung kommen. Der symmetrische Schlüssel wird vom Absenderzufällig generiert und dem Empfänger asymmetrisch verschlüsselt mitgeliefert, sodass nur dieser ihn entschlüsseln kann.

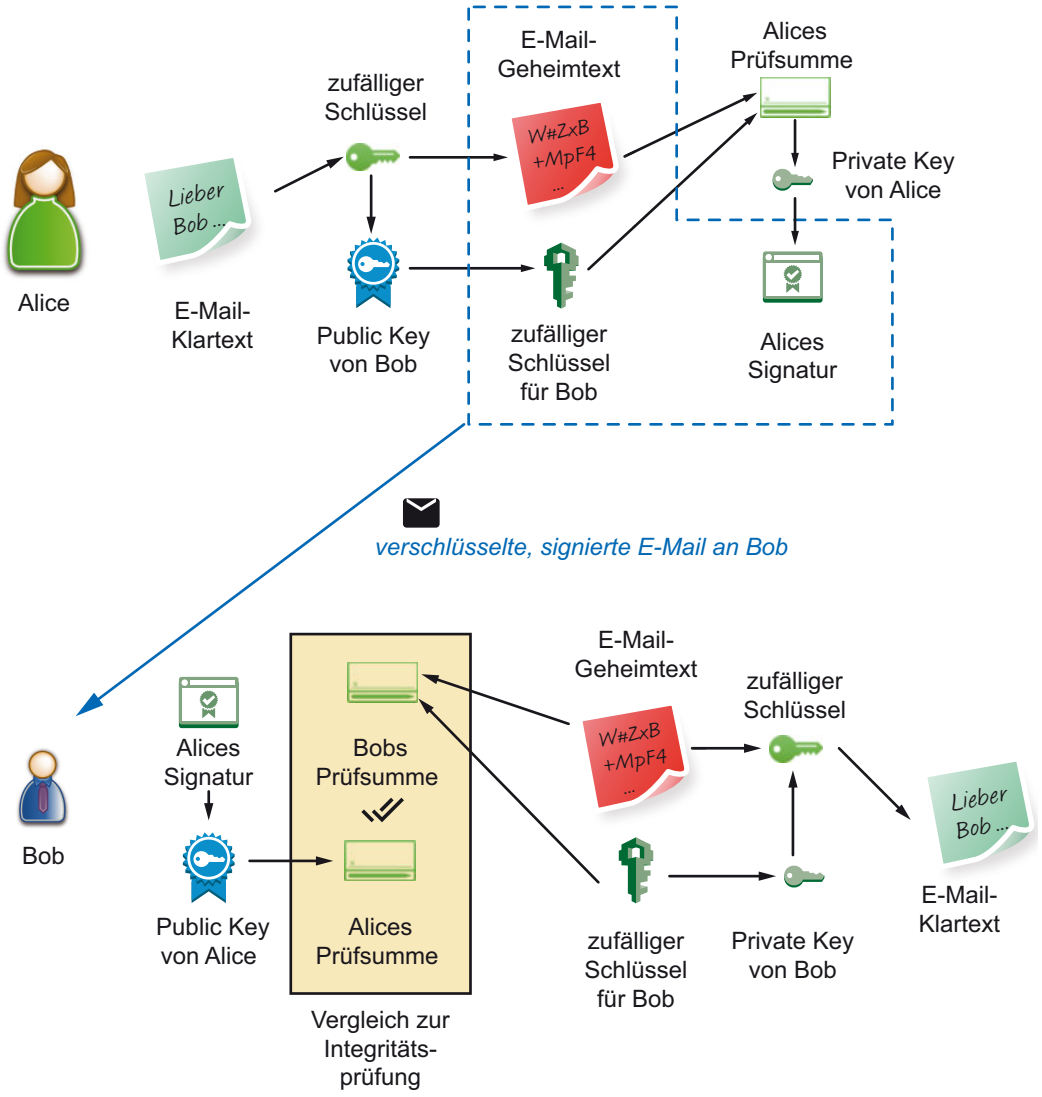

Abb. 2.1: Ablauf von E-Mail-Verschlüsselung und -Signatur

2.3.2 Einsatz von S/MIME vs. PGP/MIME

Damit Alice und Bob gegenseitig zuverlässig an ihre Public Keys gelangen, werden in der Praxis zwei unterschiedliche Verfahren eingesetzt:

a) S/MIME ist ein Datenformat für E-Mails, für das X.509v3-Zertifikate benötigt werden. Bob wird dem Zertifikat von Alice also vertrauen, wenn es von einer CA signiert ist, der Bob vertraut.

b) PGP/MIME verwendet hingegen Verschlüsselung und Signatur auf Basis von Pretty Good Privacy (PGP), wie sie von GnuPG oder OpenPGP implementiert werden. PGP setzt nicht auf das Signieren von Zertifikaten durch hierarchische CAs, sondern verwendet einen als *Web of Trust* bezeichneten Ansatz, bei dem sich die Endanwender ihre Public Keys gegenseitig bestätigen können.

Da beide Datenformate technisch nicht zueinander kompatibel sind, unterstützen viele E-Mail-Clients beide Verfahren und die Anwender müssen sich um entsprechende Schlüsselpaare bzw. Zertifikate aus beiden Welten kümmern.

Übung 2.4:

Mit wie vielen Personen kommunizieren Sie verschlüsselt per E-Mail? Worauf führen Sie den geringen Verbreitungsgrad von E-Mail-Verschlüsselung zurück?

Zusammenfassung

In diesem Kapitel haben Sie zunächst die wichtigsten Grundlagen der Kryptografie wiederholt. Als wichtige Aufgabe beim Einsatz von Kryptosystemen haben Sie die Schlüsselverwaltung identifiziert; ohne den Diffie-Hellman-Schlüsselaustausch und den Einsatz von X.509v3-Zertifikaten der Global PKI wäre sichere Kommunikation im Internet heute nicht möglich. Schließlich haben Sie anhand des Verschlüsselns und Signierens von E-Mails gesehen, wie eine konkrete Anwendung aussehen kann, in die alle Bestandteile einfließen.

Aufgaben zur Selbstüberprüfung

2.1 Überlegen Sie, ob beim Verschlüsseln von E-Mails mit S/MIME die Eigenschaft Perfect Forward Secrecy erreicht wird. Begründen Sie Ihre Antwort.

2.2 Welche Vorbehalte gibt es gegenüber der Global PKI?

3 Sichere Protokolle unterhalb der Anwendungsschicht

Wenn Sie dieses Kapitel bearbeitet haben, können Sie die Wirksamkeit von Sicherheitsmaßnahmen auf den verschiedenen Schichten des ISO/OSI-Referenzmodells voneinander abgrenzen. Sie wissen, wie Netzzugangssteuerung für Endgeräte wie Clients und Server in lokalen Netzen implementiert werden kann. Sie kennen die verschiedenen Funktionen und Betriebsmodi des IPsec-Protokolls und verstehen, wie seine Konfiguration mit dem Internet-Key-Exchange-Protokoll automatisiert werden kann. Sie wissen, wie die Sicherung von Verbindungen mit Transport Layer Security (TLS) abläuft und worauf beim praktischen Einsatz zu achten ist. Sie kennen typische Einsatzgebiete von Virtual Private Networks (VPNs) und wissen, wie sie mit IPsec oder TLS realisiert werden können.

3.1 Einordnung in die Schichten des ISO/OSI-Modells

In Abschnitt 1.3 haben Sie die OSI Security Architecture kennengelernt. Sie setzt die in Kapitel 2 behandelten kryptografischen Verfahren als Sicherheitsmechanismen ein, um die Sicherheitsdienste umzusetzen, mit denen wir die in Abschnitt 1.1 diskutierten Sicherheitsziele erreichen können.

Bevor wir in diesem Kapitel nun ausgewählte Protokolle für die sichere Netzwerkkommunikation vertiefen, müssen wir uns mit der Frage auseinandersetzen, welchen Teil einer Netzwerkkommunikation diese Protokolle überhaupt schützen sollen und können. Wie Siewissen, nimmt das ISO/OSI-Referenzmodell drei Schnitte vor: Der *Dienstschnitt* sorgt innerhalb eines Systems für die bekannten sieben Schichten – jede Schicht bietet der darüberliegenden bestimmte Dienste an. Zwischen mehreren Systemen, z. B. einem Client, einem Router und einem Server, liegt jeweils ein *Systemschnitt* vor. Als *Protokollschnitt* bezeichnen wir schließlich, dass eine Netzwerkkommunikation, wenn man sie abstrahiert betrachtet, zwischen jeweils zwei Systemen immer auf der gleichen Schicht abläuft.

Daraus ergeben sich folgende Beobachtungen:

- Wenn wir eine Sicherheitsmaßnahme auf einer niedrigen Schicht umsetzen, profitieren die darüberliegenden Schichten davon. Die Reichweite der Maßnahmen auf niedrigen Schichten ist aber begrenzt, z. B. bei Schicht 1 auf eine direkte Kabel- oder Funkverbindung.
- Je höher wir im Schichtenmodell gehen, desto mehr verschiedene Protokolle und Implementierungen gibt es. Dieselben Sicherheitsmaßnahmen müssen dann entsprechend oft umgesetzt werden.

Beispiel 3.1:

Nehmen wir an, wir könnten zwischen einem PC und dem Switch, an dem er angeschlossen ist, eine garantiert abhör- und manipulationssichere Verkabelung schaffen. Dann wäre die Netzkommunikation in diesem lokalen Bereich abgesichert; diese Sicherheit endet aber am Switch. Wenn der PC über das Internet mit anderen Systemen kommuniziert, bringt dieser Schutz insgesamt nur wenig.

In Abschnitt 2.3 haben wir uns mit E-Mail-Verschlüsselung befasst. E-Mails sind eine Schicht-7-Anwendung. Entsprechend muss Unterstützung für E-Mail-Verschlüsselung in jedem E-Mail-Programm implementiert werden und neben E-Mails gibt es beliebig viele weitere Anwendungen und Protokolle auf Schicht 7, die ebenfalls Sicherheitsmaßnahmen implementieren müssen.

Eine gewisse Redundanz an Sicherheitsmaßnahmen ist dabei durchaus gewünscht und wird als *Defense in depth* bezeichnet: Falls eine Maßnahme gebrochen wird, sind noch andere vorhanden. Im Regelfall wird man aber eher versuchen, Sicherheitsmaßnahmen auf den verschiedenen Schichten geschickt miteinander zu kombinieren. Wir arbeiten uns deshalb im Folgenden von unten nach oben durch das ISO/OSI-Schichtenmodell:

- Network Access Control (NAC, Abschnitt 3.2) regelt den Zugang von Endgeräten zum lokalen Netz und ist Schicht 2 zuzuordnen.
- IPsec (Abschnitt 3.3) dient der Verschlüsselung und Integritätssicherung von IP-Paketen und arbeitet wie Virtual Private Networks (VPN, Abschnitt 3.4) auf Schicht 3.
- Transport Layer Security (TLS, Abschnitt 3.5) wird zur Absicherung von TCP/IP-Verbindungen eingesetzt; es kann den Schichten 5–6 zugeordnet werden.
- Auf Anwendungsschicht betrachten wir in Kapitel 4 dann sowohl Protokolle, die auf TLS aufsetzen, als auch solche wie DNSSEC, die zusätzliche Sicherheitseigenschaften mit sich bringen oder einen eigenen Lösungsweg verfolgen.

Als *Ende-zu-Ende*-Sicherheitsmaßnahmen verstehen wir dabei solche, die übertragene Daten über alle Zwischenstationen hinweg absichern.

Beispiel 3.2:

Ende-zu-Ende-Verschlüsselung bedeutet, dass nur der Absender und der Empfänger mit dem Klartext arbeiten. Alle Transitsysteme, z. B. Router für IP-Pakete, bekommen als transportierte Daten nur die Geheimtexte zu sehen.

3.2 Network Access Control (NAC)

Network Access Control (NAC) wird eingesetzt, um nur ausgewählten Geräten bzw. Benutzern Zugriff auf ein lokales Netz zu geben. Beispielsweise soll in vielen Firmen vermieden werden, dass Beschäftigte ihre privaten Endgeräte mit an den Arbeitsplatz bringen und dort nutzen, z. B. weil für diese kein ausreichender Schutz vor Schadsoftware sichergestellt werden kann und sie dann das lokale Firmennetz gefährden würden. Vom WLAN-Zugang in Cafés oder Hotels kennen Sie vielleicht die sogenannten Landing Pages: Webseiten, die als Erstes im Browser angezeigt werden, nachdem man sich mit dem Netz verbunden hat; dort muss man sich entweder authentifizieren oder zumindest die Nutzungsbedingungen anerkennen.

3.2.1 MAC-basierte NAC

Die einfachste Form von NAC erfolgt auf Basis der Ethernet- bzw. MAC-Adressen der angeschlossenen Endgeräte: Der Switch bzw. Accesspoint, mit dem ein Endgerät verbunden wird, leitet nur Datenpakete von und an freigeschaltete MAC-Adressen weiter.

Als *MAC Filter* kennen Sie dies vielleicht von Ihrem Consumer-WLAN-Accesspoint zu Hause. Bei administrierbaren Switches kann typischerweise für jeden Port einzeln eingestellt werden, welche MAC-Adressen dort zugelassen sind.

MAC-basierte NAC ist mit Pflegeaufwand verbunden, weil die MAC-Adresse jedes zugelassenen Endgeräts in die Konfiguration der jeweiligen Netzkomponente einzutragen ist. Von Angreifern kann sie leider sehr einfach umgangen werden: Die MAC-Adresse wird in die Ethernet-Datenpakete geschrieben und kann genauso einfach gefälscht werden wie alle anderen Angaben; u. a. bei Windows-, Linux- und Mac OS-Geräten kann man die MAC-Adresse jedes Netzwerk-Interfaces mit Bordmitteln ändern. Der Angreifer muss also nur die MAC-Adresse eines zugelassenen Gerätes in Erfahrung bringen.

Beispiel 3.3:

Auf vielen Geräten sind ab Werk kleine Aufkleber angebracht, auf denen die MAC-Adresse steht. Auch durch Umstecken des Netzwerkkabels eines zugelassenen Geräts, z. B. statt in die Wanddose ins Notebook des Angreifers, kann die MAC-Adresse einfach ausgelesen werden.

Übung 3.1:

Finden Sie heraus, wie Sie die MAC-Adresse eines Windows 10-Rechners ändern können.

3.2.2 NAC mit IEEE 802.1X

Der Standard IEEE 802.1X kann für NAC auf Geräte- oder Benutzerbasis eingesetzt werden; er ist also flexibler als rein gerätebasierte MAC-Filter und kann nicht durch MAC-Spoofing umgangen werden. Die Umsetzung ist allerdings etwas aufwendiger, da i. d. R. ein zentraler Authentifizierungsserver benötigt wird. 802.1X unterscheidet wie in Abb. 3.1 dargestellt folgende Rollen für beteiligte Systeme:

- Als *Supplicant* (Bittsteller) wird ein 802.1X-fähiges Endgerät bezeichnet, das Zugriff auf das Netz bekommen möchte.
- Der Switch oder WLAN-Accesspoint, mit dem sich der Supplicant verbindet, heißt *Authenticator*.
- Über den Authenticator wird eine Verbindung zwischen dem Supplicant und dem *Authentifizierungsserver* aufgebaut. Der Supplicant muss diesem nachweisen, dass er zugriffsberechtigt ist.

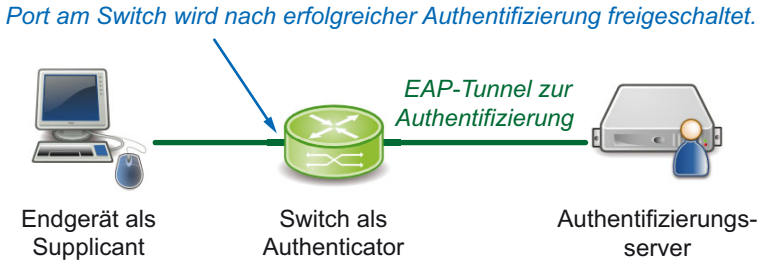

Abb. 3.1: Rollen bei 802.1X-basierter Network Access Control

IEEE 802.1X spezifiziert keine eigenen Authentifizierungsprotokolle, sondern nutzt üblicherweise das Extensible Authentication Protocol (EAP) für die Kommunikation zwischen Supplicant und Authentifizierungsserver.Letzterer kann entsprechend frei darüber entscheiden, wie er die Zugriffsberechtigung des Supplicants prüft; in der Praxis kommen oft folgende Varianten zum Einsatz:

- *Benutzerbasierte* Authentifizierung: Überprüfung von Benutzername und Passwort, ggf. auch Mehrfaktor-Authentifizierung, z. B. hardwarebasiert mit RSA SecurId-Token oder softwarebasiert mit dem Google Authenticator.
- *Gerätebasierte* Authentifizierung: Überprüfung eines Client-Zertifikats oder Kommunikation mit einer clientseitig installierten NAC-Software, die z.B. auch meldet, ob auf dem Endgerät aktuelle Patches und ein Virenscanner installiert sind.

Der Authentifizierungsserver teilt dem Authenticator dann mit, ob das Endgerät zugelassen werden soll oder nicht. NAC dient also nur dazu, in einem lokalen Netz zu regeln, welche Geräte angeschlossen werden dürfen; zur Absicherung der Kommunikation der zugelassenen Geräte sind die nachfolgenden Maßnahmen erforderlich.

Übung 3.2:

Informieren Sie sich im Internet über NAC-Produkte, u.a. das Open-Source-Produkt openNAC. Welche Komponenten werden für eine passwortbasierte Benutzerauthentifizierung üblicherweise im Zusammenspiel mit 802.1X verwendet?

3.2.3 Zugang zu WLAN-Netzen mit WPA2

Beim Anschluss von Endgeräten über WLAN fällt der mit Glasfaser- oder Kupferverkabelung implizit verbundene Schutz – eine reguläre Nutzung oder ein Angriff erfordern unmittelbare physische Nähe – weg: Die elektromagnetischen Wellen können von jedem anderen WLAN-fähigen Endgerät in der Umgebung empfangen oder gesendet werden. Aus diesem Grund kommt sowohl NAC als auch einer grundlegenden Vertraulichkeits- und Integritätssicherung im WLAN eine besondere Bedeutung zu.

Das Verfahren WiFi Protected Access 2 (WPA2) hat seine Vorgänger WPA und Wired Equivalent Privacy (WEP) bereits 2004 abgelöst und ist in IEEE 802.11i standardisiert. Für NAC können entweder Pre-Shared Keys oder das oben behandelte 802.1X eingesetzt werden; aufgrund der typischen Anwendungsszenarien „Consumer-Bereich“ und „Unternehmensumfeld“ werden die beiden Varianten auch als WPA2 Personal und WPA2 Enterprise bezeichnet.

WPA2 verwendet im Kern AES mit 128 Bit langen Schlüsseln als Chiffre; dafür wird das in IEEE 802.11i festgelegte Verfahren CCMP eingesetzt, das Verschlüsselung mit Integritätssicherung kombiniert. Die verwendeten Schlüssel werden aus einem sogenannten Master Key abgeleitet und sind kurzlebig, werden also regelmäßig und recht häufig erneuert. Zudem werden für beide Kommunikationsrichtungen – vom Endgerät zum Accesspoint und umgekehrt – unterschiedliche Schlüssel verwendet.

Bei Verwendung von 802.1X für WLAN-NAC kommen dabei individuelle Master Keys pro Endgerät zum Einsatz, die der Authentifizierungsserver vorgibt. Anders sieht es mit Pre-Shared Keys aus: Ein Angreifer, der den Pre-Shared Key kennt, kann den Verbindungsaufbau eines neuen WLAN-Endgeräts mitsamt den dabei abgeleiteten Schlüsseln

abhören. Zudem wird bei WPA2-PSK keine Perfect Forward Secrecy erreicht: Wer nachträglich Kenntnis des Pre-Shared Keys und damit des Master Keys erlangt, kann früher aufgezeichnete WLAN-Datenpakete entschlüsseln. Die Integration eines Diffie-Hellman-Schlüsselaustausches in das Verfahren würde zwar dieses Teilproblem lösen, wäre aber immer noch anfällig für Man-in-the-Middle-Angriffe durch *Rogue Accesspoints*. Diese würden sich nur durch den Einsatz z. B. von X.509v3-Zertifikaten vermeiden lassen; der Gesamtaufwand wäre dann allerdings schon sehr nahe an der Variante mit 802.1X.

Beispiel 3.4:

Als Nutzer öffentlicher WLANs, z. B. in Hotels und Gaststätten, sollte man sich deshalb bewusst sein, dass ein WPA2-PSK-gesichertes Netz nicht besser vor Lauschangriffen durch andere Nutzer schützt als ein gänzlich ungesichertes WLAN.

3.3 IPsec

IPsec ist eine von der Internet Engineering Task Force (IETF) standardisierte Sicherheitserweiterung für das Internet-Protokoll (IP). Sie wurde mit IPv6 eingeführt, kann aber auch für das nach wie vor verbreitete IPv4 genutzt werden. Dabei sollen die drei klassischen Sicherheitsprobleme von IP-Kommunikationsverbindungen gelöst werden:

a) Authentisierung: IP-Pakete können mit beliebig gefälschten Absenderadressen verschickt werden (IP Spoofing).

b) Vertraulichkeit: IP-Pakete können einfach mitgehört werden, es sind Man-in-the-Middle-Angriffe und Verkehrsflussanalysen möglich.

c) Integrität: Angreifer können Inhalte von IP-Paketen manipulieren oder mitgehörte Datenpakete neu einspielen (Replay).

IPsec führt zu diesem Zweck zwei neue Protokolle ein: Authentication Header (AH) und Encapsulating Security Payload (ESP); zudem verwendet es das Internet-Key-Exchange (IKE)-Protokoll zur Verwaltung der für die Kryptoverfahren benötigten Schlüssel. Diese Bestandteile betrachten wir im Folgenden genauer.

3.3.1 Betriebsmodi und Funktionsweise

Wie Abb. 3.2 zeigt, kann IPsec in zwei Varianten, die als Betriebsmodi bezeichnet werden, verwendet werden:

a) Im *Transport Mode* kommunizieren zwei Endgeräte über IPsec miteinander; IPsec wird also als Ende-zu-Ende-Sicherheitsmaßnahme eingesetzt.

b) Im *Tunnel Mode* kommuniziert jedes Endgerät in seinem lokalen Netz über eine herkömmliche IP-Verbindung mit einem sogenannten *Security Gateway*. Der Gateway des einen Kommunikationspartners kommuniziert dann IPsec-gesichert mit dem Gateway des anderen Kommunikationspartners. Er nimmt also sozusagen das IP-Paket des Absenders entgegen, verpackt es in ein IPsec-Paket und schickt dieses zum Gateway auf der Empfängerseite. Dieser packt das ursprüngliche IP-Paket wieder aus und stellt es auf herkömmliche Weise dem Empfänger zu.

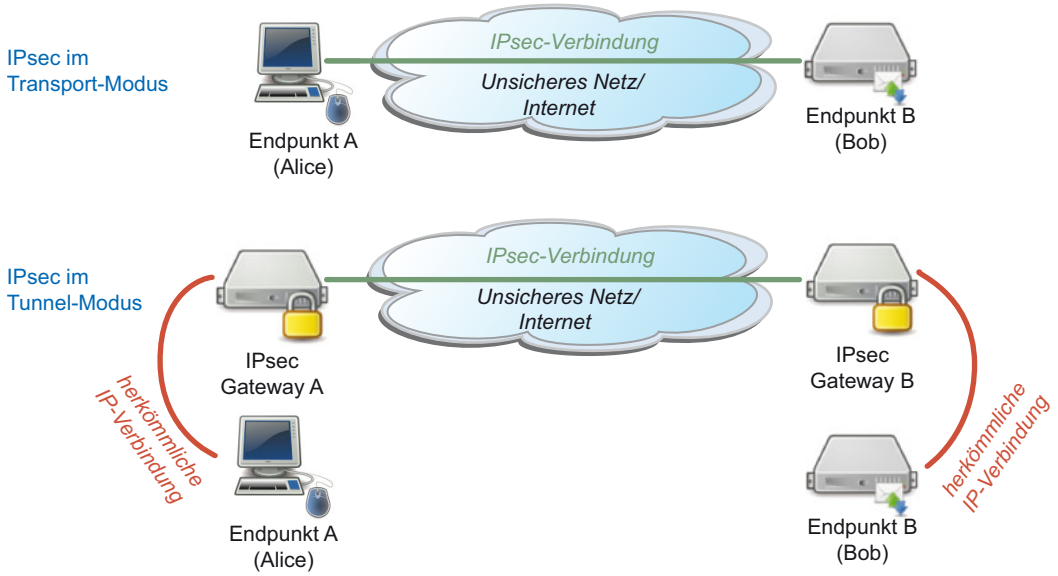

Abb. 3.2: IPsec: Transport Mode und Tunnel Mode

Der Tunnel Mode kann also insbesondere dann eingesetzt werden, wenn die Endgeräte selbst nicht IPsec-fähig sind; allerdings wird dabei keine vollständige Ende-zu-Ende-Sicherheit erreicht, da IPsec nur zwischen den beiden Gateways eingesetzt wird. Typisches Einsatzgebiet sind Firmen-Standortkopplungen über das Internet: Die Daten sollen gesichert über ein unsicheres Netz wie das Internet transportiert werden, ohne den IPsec-Konfigurationsaufwand für alle Endgeräte an allen Standorten betreiben zu müssen.

Der Einsatz von IPsec Security Gateways erreicht keine Ende-zu-Ende-Sicherheit!

Übung 3.3:

Finden Sie heraus, welche der von Ihnen verwendeten Geräte IPsec-fähig sind.

Um zu verstehen, wie die IPsec-Protokolle funktionieren, müssen wir kurz rekapitulieren, wie die IP-basierte Nachrichtenübertragung funktioniert: Das IP-Protokoll arbeitet auf Schicht 3 des ISO/OSI-Modells und führt u. a. das Konzept von IP-Adressen für Absender und Empfänger ein. Auf Schicht 4 muss man sich dann für ein Protokoll wie TCP oder UDP entscheiden. Wie in Abb. 3.3 links dargestellt ist, besteht ein übertragenes TCP/IP-Datenpaket deshalb aus drei Teilen: IP-Header, TCP-Header und Nutzdaten; die Nutzdaten werden auch als TCP-Payload bezeichnet. Damit der Empfänger eines IP-Pakets weiß, dass es sich um ein TCP/IP-Paket handelt, wird in das IP-Header-Feld *Protocol* der Wert für TCP eingetragen.

Links: Herkömmliche TCP/IPv4-Paket

Rechts: TCP/IPv4-Paket mit IPsec AH im Transport Mode.
Die Prüfsumme wird über die markierten Datenfelder gebildet.

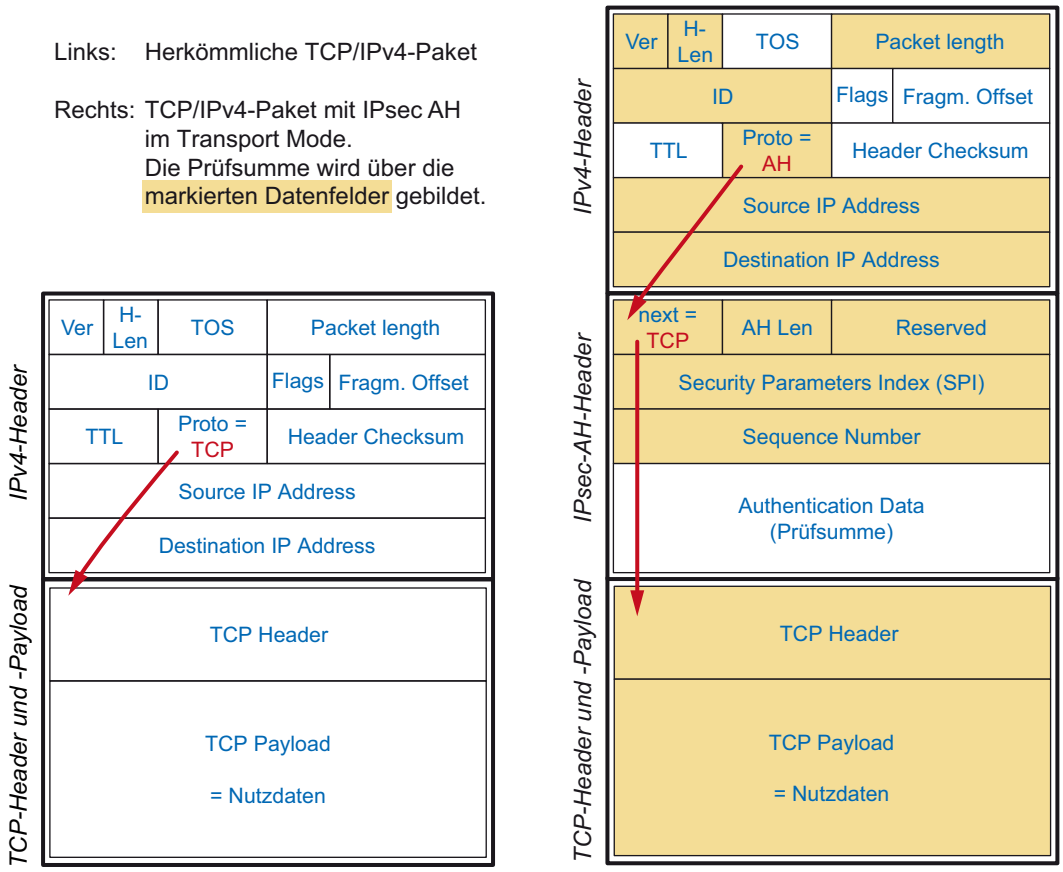

Abb. 3.3: TCP/IP-Datenpaket im herkömmlichen Fall und mit IPsec-AH-Header im Transport-Modus

An dieser Stelle setzt nun IPsec an: Zwischen den IP-Header und den TCP-Header wird ein AH- und/oder ein ESP-Header eingeschoben. Abb. 3.3 zeigt rechts den einfachsten Fall, bei dem nur der AH-Header im Transport-Modus eingesetzt wird: Das IP-Header-Feld *Protocol* signalisiert, dass im gesamten Datenpaket als Nächstes ein IPsec-AH-Header folgt. Im AH-Header ist dann vermerkt, dass wiederum anschließend ein TCP-Header samt TCP-Payload folgt.

Der IPsec-AH-Header umfasst im Wesentlichen eine Sequenznummer, anhand derer Replay-Angriffe erkannt werden können, und eine kryptografische Prüfsumme. Diese Prüfsumme wird über alle (TCP-)Nutzdaten des Datenpakets und diejenigen IP-Header-Felder gebildet, die sich beim regulären Transport des Datenpakets nicht ändern dürfen. Ausgenommen ist also z. B. das IP-Header-Feld Time-to-Live (TTL), das von jedem Router auf dem Transportweg um den Wert 1 heruntergezählt wird, um Fehlersituationen wie nicht erreichbare Zielsysteme erkennen zu können. Damit ein Angreifer die Prüfsumme nicht einfach genauso wie die Nutzdaten manipulieren kann, fließt i. d. R. zusätzlich ein *Shared Secret* ein, auf das sich Absender und Empfänger geeinigt haben. Welches Shared Secret dafür zu verwenden ist, wird durch das AH-Header-Feld Security Parameters Index (SPI) festgelegt, auf das wir in Abschnitt 3.3.2 genauer eingehen.

Durch den Einsatz von IPsec AH kann also insbesondere sichergestellt werden, dass der Empfänger die Integrität des gesamten empfangenen Datenpakets einschließlich der Absender-IP-Adresse überprüfen kann. Beim Einsatz von IPsec im Tunnel-Modus sieht das resultierende Datenpaket wie in Abb. 3.4 dargestellt etwas anders aus: Da hier nur die

beiden Security Gateways miteinander IPsec-basiert kommunizieren, wird das ursprüngliche TCP/IP-Paket schlichtweg in Gänze in ein neu erstelltes AH/IP-Datenpaket verpackt. Sein gesamter Inhalt fließt dabei in die Prüfsummenberechnung mit ein. Der empfangende Security Gateway verifiziert dann die Korrektheit der Prüfsumme und schickt das enthaltene TCP/IP-Paket weiter zum eigentlichen Empfänger.

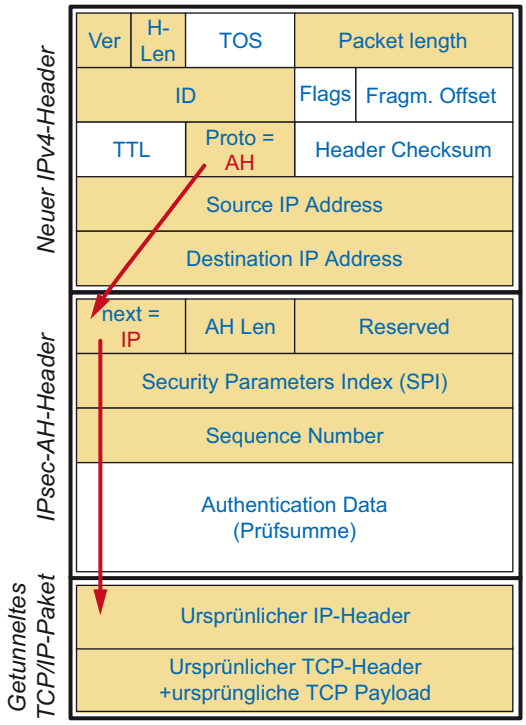

Abb. 3.4: IPsec Authentication Header im Tunnel-Modus

IPsec ESP kommt zum Einsatz, wenn die Nutzdaten verschlüsselt werden sollen. Auch hier spezifiziert das Header-Feld SPI u. a. den zu verwendenden Schlüssel. Im Transport-Modus wird der TCP-Header mitsamt den Nutzdaten wie in Abb. 3.5 links gezeigt verschlüsselt übertragen; im Tunnel-Modus ver- bzw. entschlüsseln die beteiligten Security Gateways das gesamte ursprüngliche TCP/IP-Paket (siehe Abb. 3.5 rechts). Da die verschlüsselten Daten je nach eingesetzter Chiffre als Länge das Vielfache einer Blockgröße wie 128 Bit aufweisen müssen, kommt ggf. Padding zum Einsatz; dieses wird als Gegenstück zum ESP-Header auch als ESP-Trailer bezeichnet.

Übung 3.4:

Warum stellt IPsec ESP im Tunnel-Modus bei der Kopplung von Firmenstandorten über das Internet einen guten Schutz vor Verkehrsflussanalysen dar?

Optional kann auch ESP eine Prüfsumme zur Integritätssicherung verwendet; diese bezieht allerdings keine Felder des IP-Headers mit ein. Um die Authentizität der Absender-IP-Adresse sicherzustellen, muss also immer IPsec AH eingesetzt werden.

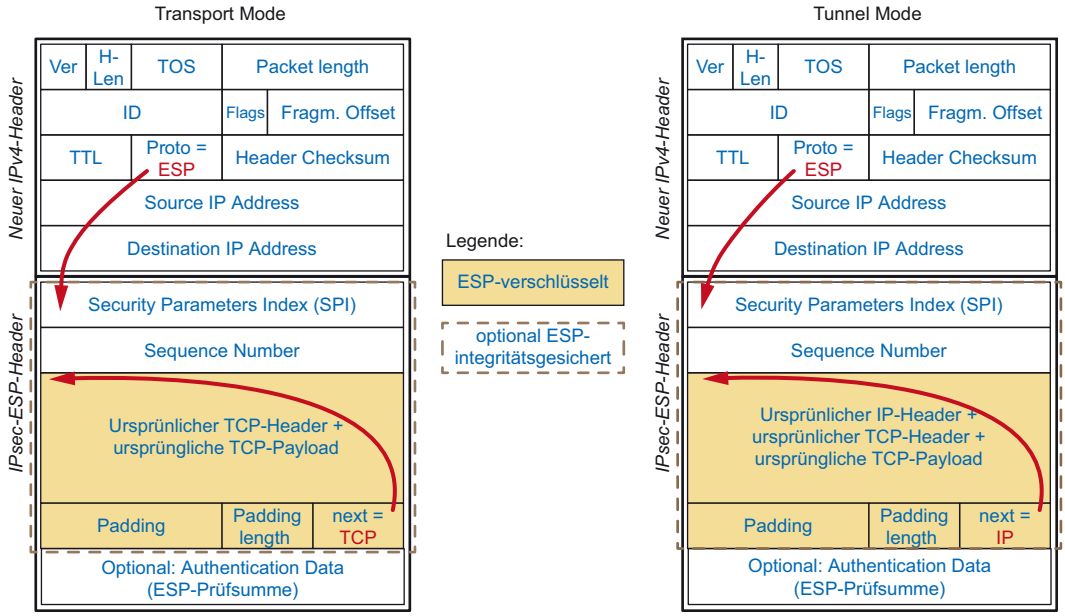

Abb. 3.5: IPsec Encapsulating Security Payload im Transport- und Tunnel-Modus

Die beiden IPsec-Header AH und ESP können und müssen miteinander kombiniert werden, wenn sowohl Integritäts- als auch Vertraulichkeitssicherung erreicht werden sollen. Abb. 3.6 zeigt den schematischen Aufbau eines entsprechenden Datenpakets.

Abb. 3.6: Schematischer Paketaufbau bei Kombination von IPsec AH und ESP

3.3.2 Schlüsselmanagement mittels Internet Key Exchange

Im oberen Abschnitt haben Sie gesehen, dass für die Prüfsummenberechnung und Verschlüsselung der Security Parameters Index (SPI) eine wichtige Rolle spielt. Wenn wir uns das Thema „IPsec-Schlüsselmaterial“ gleich näher ansehen, werden wir feststellen, dass es bei einer größeren Anzahl an Kommunikationspartnern schnell unübersichtlich wird. Fest mit IPsec verbunden ist deshalb das Internet-Key-Exchange(IKE)-Protokoll, das die Aushandlung und regelmäßige Aktualisierung aller dieser Konfigurationsangaben vollständig automatisieren kann.

Die Konfiguration einer IPsec-Verbindung wird als Security Association (SA) bezeichnet; eine SA wird durch eine Kombination von drei Merkmalen eindeutig identifiziert:

a) Security Parameters Index wie im AH- oder ESP-Header angegeben

b) Ziel-IP-Adresse des Empfängers

c) Angabe des verwendeten Protokolls (AH oder ESP)

Für eine bidirektionale Verbindung zwischen zwei Endpunkten werden also immer mindestens zwei SAs benötigt – je eine für die Hin- und für die Rückrichtung. Jedes IPsec-fähige Gerät speichert seine SAs lokal in einer sogenannten Security Policy DatabaseindexSecurity Policy Database (SPD).

In einer SA werden im Wesentlichen festgehalten:

- IPsec-Protokoll-Modus: Transport oder Tunnel
- verwendete Verschlüsselungs- und Prüfsummenalgorithmen mit den dazugehörenden Schlüsseln oder Zertifikaten
- Angabe zur Lebensdauer der SA, z. B. zum Anstoßen der Erneuerung von Schlüsseln
- Sequenznummern zum Erkennen von Duplikaten/Replay Attacks

Das UDP/IP-basierte Protokoll IKEv2 hat zum Ziel, diese SAs automatisiert zwischen zwei IPsec-fähigen Geräten auszuhandeln; dabei müssen sich die Geräte gegenseitig zuverlässig authentisieren können, da die ausgehandelten Schlüssel sonst wertlos für die Authentisierung des Absenders auf Basis von IPsec AH wären. Eine IKE-Verbindung läuft deshalb in zwei Phasen ab, von denen die erste wie in Abb. 3.7 dargestellt aus zwei Schritten besteht:

a) Alice als Initiator der IKE-Verbindung legt Bob eine Liste von ihr unterstützter Verschlüsselungs- und Prüfsummenverfahren vor und schickt ihren Teil eines Diffie-Hellman-Schlüsselaustausches mit. Bob antwortet mit seiner Auswahl der Verfahren und seinem Teil des DH-Schlüsselaustausches. Damit steht nun eine SA für die IKE-Verbindung an sich fest.

b) Alice und Bob überprüfen gegenseitig ihre Authentizität anhand von X.509v3-Zertifikaten oder einem Shared Secret und handeln die Parameter für die erste IPsec-SA aus.

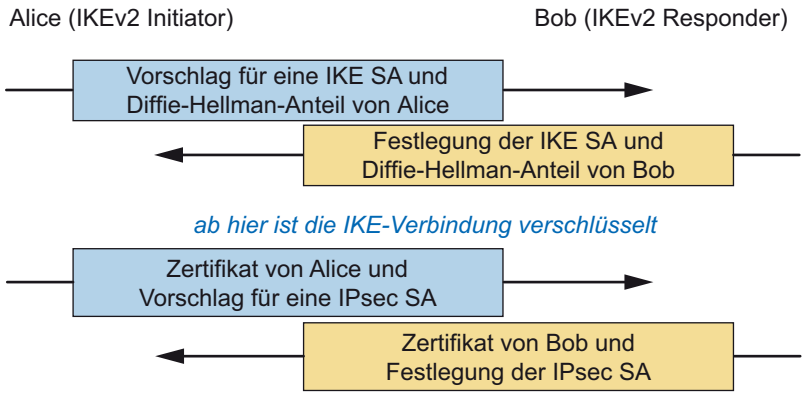

Abb. 3.7: Protokoll IKEv2: Ablauf der ersten Phase

Anschließend können über die gesicherte IKE-Verbindung beliebig viele weitere IPsec-SAs ausgehandelt bzw. erneuert werden. Bei der Erzeugung neuer SAs ist wiederum die priorisierte Schnittmenge der unterstützten Verfahren ausschlaggebend. Zur Vereinbarung von Schlüsseln, z. B. für symmetrische Kryptoverfahren, kommt wiederum das Diffie-Hellman-Verfahren zum Einsatz: Die mit ihm festgelegte geheime Zahl dient dabei als Eingabe für einen Pseudozufallszahlengenerator, mit dem Alice und Bob die Schlüssel in den benötigten Längen erzeugen können.

Wenn also X.509v3-Zertifikate aus der Global PKI verwendet werden, können beliebige IPsec-fähige Endgeräte miteinander SAs und damit auch Schlüssel vereinbaren, die für die Absicherung der IPsec-Kommunikation eingesetzt werden, ohne dass ein manueller Konfigurationsaufwand anfällt.

Das IKE-Protokoll dient der automatischen IPsec-Konfiguration. Neben der initialen Auswahl von Verfahren werden insbesondere Schlüssel für Chiffren ausgehandelt, die auch regelmäßig automatisch erneuert werden können.

Übung 3.5:

Suchen Sie im Internet nach der aktuellen Spezifikation zulässiger Kryptoalgorithmen für IKE. Welche Verschlüsselungs- und Hashverfahren dürfen demnach verwendet werden?

3.4 Virtual Private Networks

Ein Virtual Private Network (VPN) dient allgemein dazu, auf Basis eines vorhandenen physischen Rechnernetzes ein zusätzliches logisches, also „virtuelles“ Rechnernetz einzurichten. Die an das virtuelle Netz angeschlossenen Systeme können dann so miteinander kommunizieren, als wären sie direkt über dessen Verkabelung – die aber real nicht existiert – verbunden. Auf Schicht 2 des ISO/OSI-Referenzmodells sind mit VLANs nach IEEE 802.1Q ähnliche Ansätze bekannt, um die an einen einzigen physischen Switch angeschlossenen Geräte in logisch getrennte Netze einzuordnen. VPNs arbeiten aber auf Schicht 3, also auf Ebene der IP-Pakete; sie verbinden typischerweise Netze und Endgeräte über das Internet und stellen mittels Verschlüsselung und kryptografischen Prüfsummen die Vertraulichkeit und Integrität der über das VPN übertragenen Daten sicher.

VPNs dienen wie in Abb. 3.8 gezeigt üblicherweise einem der folgenden Anwendungsfälle:

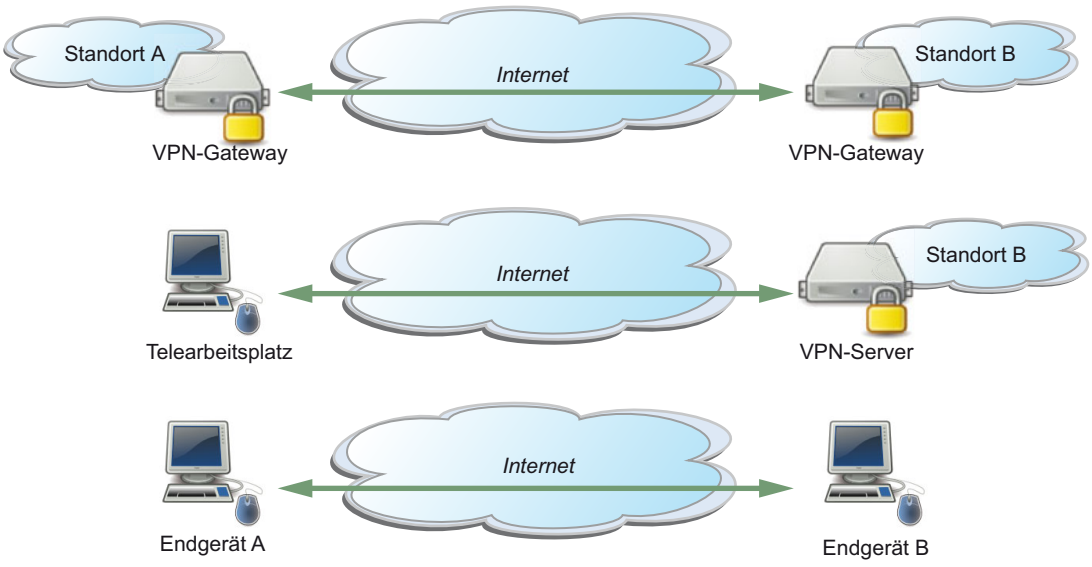

Abb. 3.8: Einsatzvarianten für Virtual Private Networks

- Kopplung ganzer Standorte: Organisationen, die sich räumlich über mehrere Standorte erstrecken, möchten zwischen diesen Standorten oftmals Daten sicher austauschen. Häufig werden IT-Dienste wie E-Mail-Server und Intranet-Webserver nur an einem zentralen, größeren Standort betrieben und sollen von den Außenstandorten so genutzt werden können, als wären sie ins dortige LAN eingebunden.
- Fernzugriff für Einzelarbeitsplätze: Für Telearbeitsplätze, Außendienstmitarbeiter und andere Geschäftsreisende, die z.B. auf die Nutzung eines nicht vertrauenerweckenden WLAN-Zugangs im Ausland angewiesen sind, bietet ein VPN die Möglichkeit, sich über einen sicheren Kanal von außen mit dem Netz des Arbeitgebers zu verbinden. Darüber können dann Dienste wie E-Mail und Intranet genauso genutzt werden, als ob man in der Firmenzentrale direkt mit dem LAN verbunden wäre.
- Kopplung von Einzelgeräten: Mehrere Systeme können über das Internet zu einem virtuellen LAN zusammengeschlossen werden. Befreundete Personen können sich so z.B. gegenseitig Zugriff auf Dateien geben, ohne diesen Dienst von außen aus dem gesamten Internet erreichbar machen zumüssen. Bei mehr als zwei Systemen kann ein zentraler VPN-Server zum Einsatz kommen, technisch ist aber eine beliebige Vermaschung auch ohne zentrale Instanz möglich.

In Abschnitt 6.1 werden wir sehen, dass VPNs auch mit mäßigem Erfolg eingesetzt werden können, um die eigene Kommunikation zu verschleiern. Für die Nutzung von VPNs ist relevant, welche Daten über das VPN übertragen werden: Im Fall der Standortkopplung werden über das VPN üblicherweise nur diejenigen Daten geschickt, deren Ziel-IP-Adressen zum jeweils anderen Standort gehören. Alle anderen Daten werden wie im Fall ohne VPN direkt über das Internet übertragen. Beim Fernzugriff für Einzelarbeitsplätze kann es aber gewünscht sein, dass der gesamte Datenverkehr eines angebundenen Endgeräts über das VPN läuft. Im Fall des Auslandsreisenden, dessen Hotel-WLAN als zu unsicher gilt, würde also auch die gesamte Kommunikation mit beliebigen Internetdiensten zunächst über einen VPN-Tunnel und das Firmennetz laufen. Wenn das Hotel-WLAN von einem Angreifer überwacht wird, sieht dieser nur die Kommunikation mit dem VPN-Server und nicht die Einzelverbindungen zu Servern im Internet.

3.4.1 VPNs mit IPsec

Mit IPsec im Tunnel Mode haben Sie in Abschnitt 3.3.1 schon einen Standard für das Einrichten von VPNs kennengelernt. Viele Router- und Firewall-Produkte können als IPsec Security Gateways konfiguriert werden.

Wenn Security Gateways auf beiden Seiten einer Kommunikationsbeziehung eingesetzt werden, liegt eine klassische Standortkopplung vor. Diese Variante stellt den derzeit häufigsten IPsec-VPN-Anwendungsfall in der Praxis dar. Auch dieanderen Varianten können mit IPsec umgesetzt werden. Die IPsec-Konfiguration von Endgeräten mit Betriebssystem-Bordmitteln war bis vor wenigen Jahren aber insbesondere für wenig IT-affine Benutzer noch recht mühsam und fehleranfällig. Deshalb kommt oftdedizierte VPN-Client-Software zum Einsatz, die entweder die IPsec-Konfiguration vereinfacht oder wie im nächsten Abschnitt betrachtet ein anderes VPN-Protokoll umsetzt.

3.4.2 VPNs auf TLS-Basis

Viele kommerzielle VPN-Produkte und auch Open-Source-Lösungen wie OpenVPN verwenden das TLS-Protokoll, mit dem wir uns im nachfolgenden Abschnitt im Detail auseinandersetzen. Die Grundidee ist aber dieselbe wie bei IPsec: Ein VPN-Client, typischerweise ein Endgerät wie ein Telearbeitsplatz, baut eine verschlüsselte und integritätsgesicherte Verbindung zu einem VPN-Server auf; dieser VPN-Server ist vergleichbar mit einem empfängerseitigen IPsec Security Gateway im Tunnel Mode.

Jedes IP-Paket, das über das VPN verschickt werden sollen, wird vom VPN-Client dann in ein TLS-gesichertes Datenpaket verpackt und zum VPN-Server geschickt. Dieser extrahiert das ursprüngliche IP-Paket nach Prüfsummenverifikation und Entschlüsselung und schickt es dann weiter zum eigentlichen Empfänger. Die Rückrichtung funktioniert dazu analog: Der VPN-Server verpackt das IP-Paket in ein TLS-gesichertes Datenpaket, das zum VPN-Client geschickt wird. Dieser packt es aus und verarbeitet es lokal.

Übung 3.6:

Einige Zeit lang waren VPNs zum Umgehen von Ländersperren bei Video-Streaming-Diensten populär. Wie funktioniert dies und welche Gegenmaßnahmen wurden ergriffen?

3.5 Transport Layer Security

Das Protokoll Transport Layer Security (TLS) setzen Sie sicherlich jeden Tag ein, wenn Sie Internetdienste verwenden. Es dient der Absicherung von TCP/IP-Verbindungen und übernimmt insbesondere die Sicherheitsdienste Server-Authentizitätsprüfung, Vertraulichkeit durch Verschlüsselung und Integritätssicherung durch kryptografische Prüfsummen. Es kommt bei jedem HTTPS-Zugriff auf einen Webserver und vielen anderen Protokollen, die wir in Kapitel 4 betrachten, zum Einsatz.

An vielen Stellen findet sich als Synonym zu TLS noch der Begriff Secure Socket Layer (SSL): SSL war der ursprünglich 1995 von der Firma Netscape entwickelte Vorgänger von TLS, sollte aber aufgrund diverser Design- und Implementierungsfehler längst nicht mehr verwendet werden. TLS ist ein von der Internet Engineering Task Force (IETF) gepflegter Standard, dessen Version 1.3 im Jahr 2017 erschienen ist.

Auch wenn im TCP/IP-Schichtenmodell die drei Schichten 5–7 des ISO/OSI-Referenzmodells zu einer einzigen Schicht zusammenfallen, lässt sich gut zeigen, wie TLS in das OSI-Schichtenmodell eingeordnet werden kann: Da TLS auf TCP/IP-Verbindungen aufsetzt, ist es wie in Abb. 3.9 dargestellt zunächst oberhalb von Schicht 4 angesiedelt. TLS bietet ein integriertes Session Management, d. h., über eine TLS-Verbindung zwischen Client und Server können mehrere unabhängige Kommunikationsvorgänge ablaufen; TLS nimmt also Aufgaben der OSI-Schicht 5 (Session Layer) wahr. Zudem kümmert sich TLS um die Verschlüsselung von Daten, beeinflusst also deren Darstellung; somit kann es auch der OSI-Schicht 6 (Presentation Layer) zugeordnet werden. Schließlich setzen Anwendungsprotokolle der OSI-Schicht 7 wie HTTPS darauf auf.

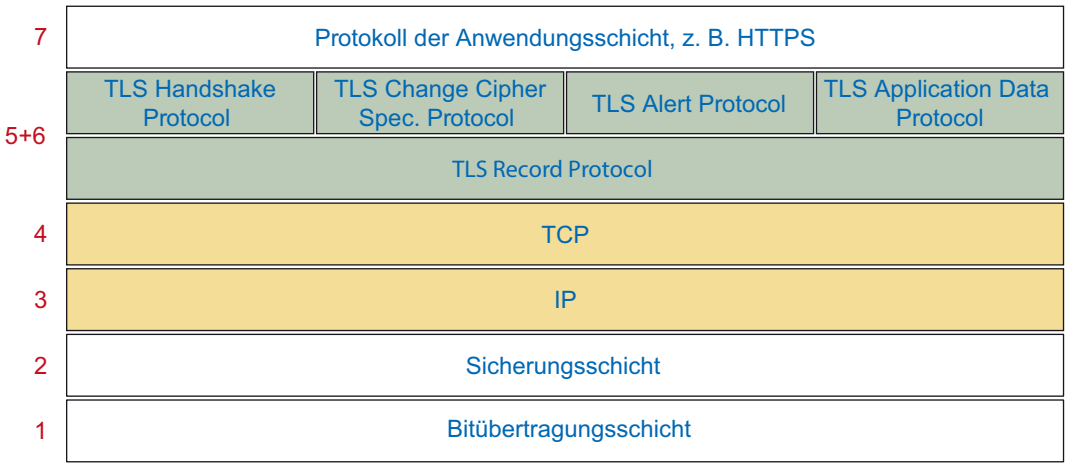

Abb. 3.9: Einordnung von Transport Layer Security ins ISO/OSI-Schichtenmodell

3.5.1 TLS Record Protocol

TLS besteht eigentlich aus fünf einzelnen Protokollen, die wie in Abb. 3.9 dargestellt in zwei logischen Schichten angeordnet werden können.

Das TLS Record Protocol bildet dabei die Basis. Es setzt in Form von TCP/IP-Paketen auf OSI-Schicht 4 auf und bietet zum einen die Datenverschlüsselung mit symmetrischen Chiffren und die Integritäts- und Authentizitätsprüfung auf Basis von Message Authentication Codes an. Die dafür erforderlichen Verfahren und Schlüssel werden im Rahmen des TLS Handshake Protocol vereinbart. Optional können die zu übertragenden Daten vorher noch komprimiert werden; aus Performancegründen wird darauf serverseitig aber oft verzichtet.

3.5.2 TLS Handshake Protocol

Das TLS Handshake Protocol wird zu Beginn einer TLS-gesicherten Verbindung ausgeführt und benötigt, um die Kryptoverfahren und Schlüssel für den Nutzdatenaustausch über das TLS Record Protocol festzulegen. Wir können es bezüglich seiner Zielsetzung also grob mit dem Aushandeln von IPsec Security Associations durch das Internet Key Exchange Protocol vergleichen. Die in Abb. 3.10 gezeigten einzelnen Schritte eines TLS Handshake können wir uns mit dem in Abschnitt 2.2 Gelernten gut herleiten:

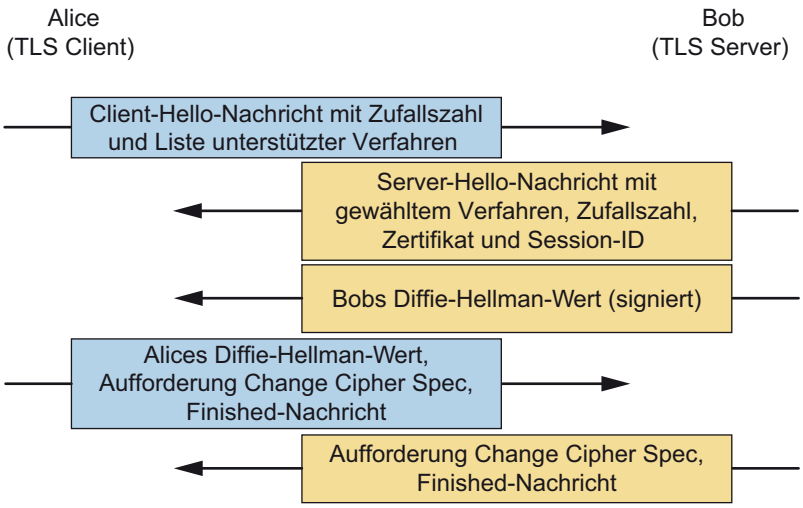

Abb. 3.10: Ablauf des TLS Handshake mit Diffie-Hellman-Schlüsselaustausch

- Der Client schickt im Rahmen der eröffnenden *Hello*-Nachricht eine Liste der von ihm unterstützten kryptografischen Verfahren und eine Zufallszahl zum Erkennen von Replay-Angriffen.
- Der Server antwortet ebenfalls mit einer *Hello*-Nachricht, in der die von ihm ausgewählten Kryptoverfahren genannt, das Server-X.509v3-Zertifikat mitgeschickt und eine TLS Session-Id zugewiesen werden.
- Ebenso schickt der Server seinen Teil des Diffie-Hellman-Schlüsselaustausches; der Server weist nach, im Besitz des richtigen Private Key zu sein, indem er diese Nachricht signiert.
- Nach Prüfung des Zertifikats schickt der Client seinen Diffie-Hellman-Teil; mit derselben Nachricht wird angekündigt, auf eine verschlüsselte Verbindung umzustellen, und das erfolgreiche Ende des Handshake signalisiert.
- Der Server kündigt ebenfalls an, auf Verschlüsselung umzustellen, und gibt den erfolgreichen Handshake-Abschluss bekannt.

Alternativ zum Diffie-Hellman-Schlüsselaustausch kann der Client auch einen zufällig gewählten Schlüssel mit dem Public Key des Servers verschlüsselt an diesen übermitteln. Zusammen mit dem bekannten Drei-Wege-Handshake beim Aufbau der TCP/IP-Verbindung müssen aber insgesamt recht viele Nachrichten zwischen Client und Server ausgetauscht werden, bis die eigentlichen Nutzdaten übertragen werden können. Die Weiterentwicklung von TLS zielt deshalb auch darauf ab, diese Latenzen beim Verbindungsaufbau zu reduzieren, indem mehrere aufeinanderfolgende Nachrichten mit demselben Datenpaket verschickt werden.

Wesentlich für die Sicherheit des TLS-Verfahrens der Verwendung aktueller Verschlüsselungs- und Prüfsummenverfahren, dass der Client das Serverzertifikat umfassend überprüft:

Passt das Zertifikat überhaupt zu dem Server, mit dem Kontakt aufgenommen wurde? Wurde es von einer vertrauenswürdigen Certificate Authority ausgestellt? Ist es noch gültig und wurde noch nicht zurückgerufen?

Wenn diese Prüfungen nicht durchgeführt werden, wird die Verbindung zwar verschlüsselt und integritätsgesichert, läuft aber möglicherweise über einen Man-in-the-Middle-Angreifer, der die Nachrichten entschlüsseln kann!

3.5.3 TLS-Protokolle

Neben dem Record Protocol und dem Handshake Protocol kommen die folgenden weiteren TLS-Protokolle zum Einsatz:

- Das TLS Change Cipher Spec Protocol umfasst nur *eine* Nachricht. Sie signalisiert, dass die weitere Kommunikation über die mit dem TLS Handshake Protocol vereinbarten Verfahren gesichert werden soll.
- Das TLS Alert Protocol dient der Mitteilung von Warnungen und Fehlerzuständen, die möglicherweise auf Angriffe zurückzuführen sind, z. B. ein fehlerhaft empfangener Message Authentication Code. Bei *Fehlern* wird anschließend die Verbindung abgebrochen; *Warnungen* können sich z. B. auf abgelaufene Zertifikate beziehen, führen aber nicht zwingend zum Abbruch.
- Das TLS Application Data Protocol dient schließlich der gesicherten Übertragung der Nutzdaten mit symmetrischen Chiffren und kryptografischen Prüfsummen.

Sollen für eine bestehende Verbindung die Kryptoverfahren bzw. die dafür verwendeten Schlüssel neu ausgehandelt werden, so können beide Seiten eine TLS Renegotiation anstoßen; dabei wird im Wesentlichen der TLS Handshake neu durchgeführt.

Zusammenfassung

In diesem Kapitel haben Sie Protokolle zur Absicherung der Netzwerkkommunikation auf den Schichten 2–6 des ISO/OSI-Schichtenmodells kennengelernt. Auf Schicht 2 haben Sie gesehen, dass die Netzzugangskontrolle auf Basis von MAC-Adressen einfach umgangen werden kann, mit IEEE 802.1X aber bessere, allerdings auch komplexere Verfahren zur Verfügung stehen. Mit IPsec haben Sie den Standard zur Absicherung der Vermittlungsschicht vertieft; es besteht aus Authentication Header zur Integritätssicherung und Encapsulating Security Payload zur Verschlüsselung der Nutzdaten und kann Ende-zu-Ende bzw. über Gateways eingesetzt werden, was als Transport bzw. Tunnel Mode bezeichnet wird.

Als derzeit wichtigstes Protokoll für die Praxis haben Sie Transport Layer Security betrachtet. TLS setzt auf TCP/IP auf und wird von vielen Protokollen der Anwendungsschicht wie HTTPS verwendet, ist selbst also den Schichten 5 und 6 zuzuordnen. Zudem haben Sie die Anwendungsfälle von Virtual Private Networks betrachtet; dabei hat sich gezeigt, dass VPNs zwar logisch auf Schicht 3 arbeiten, aber nicht nur mit IPsec, sondern auch mit TLS-basierten Tunneln realisiert werden können.

Aufgaben zur Selbstüberprüfung

3.1 Können IPsec ESP und TLS miteinander kombiniert werden? Nennen Sie jeweils ein Beispiel, wann dies sinnvoll bzw. redundant ist.

3.2 Welches Problem ergibt sich bei IPsec AH im Transport Mode bei der Verwendung privater IP-Adressen wie `192.168.0.*/24` bei der Kommunikation über das Internet mittels Network Adress Translation (NAT)?

3.3 Wie funktioniert die gegenseitige Authentifizierung der Kommunikationspartner beim Internet Key Exchange Protocol?

3.4 Wann wird bei einem TLS Handshake die Eigenschaft Perfect Forward Secrecy erreicht?

3.5 Welche zwei wichtigen Varianten bezüglich Art und Umfang übertragener Daten gibt es bei VPNs?

4 Sichere Protokolle auf der Anwendungsschicht

Nach Bearbeitung dieses Kapitels wissen Sie, welche Sicherheitsmechanismen wichtige Vertreter der Kommunikationsprotokolle auf der Anwendungsschicht einsetzen. Sie kennen die typische Vorgehensweise, um herkömmliche Kommunikationsprotokolle auf Basis von TLS abzusichern; Sie wissen aber auch, bei welchen Protokollen aus guten Gründen davon abgewichen wird. Sie kennen für verschiedene Anwendungsbereiche wie E-Mail-Versand, Instant Messaging und Internet-Telefonie sichere Kommunikationsprotokolle und die mit ihnen verbundenen verteilten Systemarchitekturen. Zudem verstehen Sie die Funktionsweise von DNSSEC und wie es zur Verbesserung der Sicherheit anderer Kommunikationsprotokolle eingesetzt werden kann.

Mit IPsec im Transport Mode und TLS haben wir zwei standardisierte, nahezu universell verwendbare Ansätze kennengelernt, um die Netzwerkkommunikation zwischen zwei Endgeräten abzusichern.[1] Sie fragen sich deshalb vielleicht, warum man zusätzlich noch sichere Kommunikationsprotokolle auf Schicht 7, also der Anwendungsschicht, benötigt und nicht z. B. einfach flächendeckend IPsec einsetzt. Der Grund dafür liegt hauptsächlich in der Historie: IPsec gibt es zwar schon lange, durch die Kombination aus Komplexität und fehlenden benutzerfreundlichen Konfigurationswerkzeugen hat sich aber noch keine signifikante Verbreitung im Bereich von Endanwendern ergeben. SSL bzw. TLS hat sich mit dem Aufkommen von eCommerce in den 1990er-Jahren entwickelt, also mit einem Fokus auf das HTTP-Protokoll, das damit zu HTTPS wird (siehe Abschnitt 4.3). Für andere Zwecke wurden aber auch damals schon sichere Kommunikationsprotokolle wie das im nächsten Abschnitt behandelte SSH entwickelt; diese sind bis heute im Einsatz, weil sie sich bewährt haben und durch eine Umstellung auf TLS keine Rückwärtskompatibilität möglich wäre.

Wir betrachten im Folgenden deshalb wichtige Vertreter für drei Arten von Schicht-7-Protokollen: Zum einen solche, die kein TLS verwenden, sondern vergleichbare Mechanismen selbst spezifiziert haben. Ferner Protokolle, die bestimmte Sicherheitseigenschaften erreichen können, bevor überhaupt eine TLS- oder IPsec-Verbindung aufgebaut wird. Schließlich werden Sie auch einige Protokolle kennenlernen, die auf TLS aufbauen und damit „sichere“ Versionen ihrer klassischen, ungeschützten Varianten darstellen.

4.1 Secure Shell

Secure Shell, kurz SSH, ist ein Protokoll, um sich über das Netz auf anderen Maschinen einloggen zu können. Es wurde 1995 als Alternative zu den damals in der UNIX-Welt verbreiteten Protokollen rlogin und Telnet entwickelt, die alle Daten unverschlüsselt übertragen haben. SSH ist in Version 2 heutzutage der Standard für die Remote-Administration von Linux- und UNIX-Servern und auch für den entfernten Kommandozeilenzugang zu Windows-Servern verfügbar.

1. TLS funktioniert nur mit TCP/IP. Datagram TLS (DTLS) funktioniert auch mit UDP, spielt in der Praxis aber keine so große Rolle.

Die SSH-Architektur besteht wie in Abb. 4.1 dargestellt ähnlich zu TLS aus mehreren einzelnen Protokollen, die im Laufe einer Verbindung zum Einsatz kommen können:

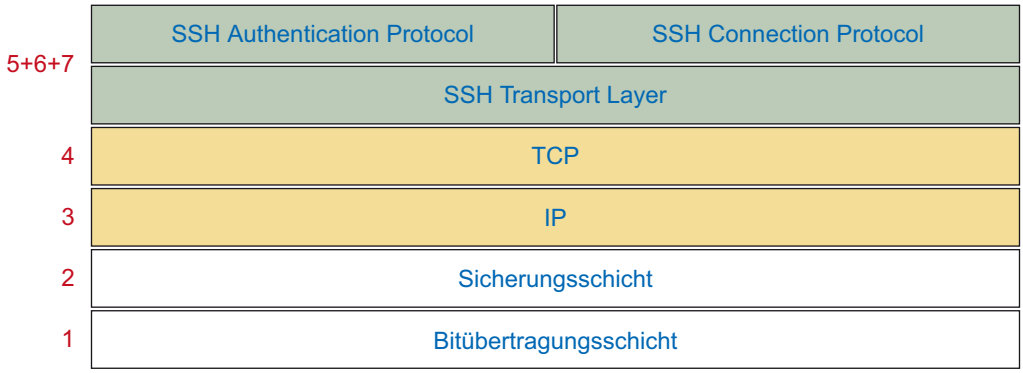

Abb. 4.1: Einordnung von Secure Shell ins ISO/OSI-Schichtenmodell

- Der SSH Transport Layer bildet die Basis. Er sorgt für die Authentisierung des Servers gegenüber dem Client und bietet Datenkompression, Verschlüsselung und Prüfsummen zur Sicherstellung von Vertraulichkeit und Integrität. Auf Basis des Diffie-Hellman-Verfahrens (siehe Abschnitt 2.2.2) wird die Vereinbarung von Sitzungsschlüsseln mit Perfect Forward Secrecy und Schlüsselerneuerung, z.B. einmal pro Stunde oder nach 1 GB übertragenem Datenvolumen, unterstützt.
- Der SSH Authentication Layer dient der Authentisierung des Clients gegenüber dem Server. Neben einer klassischen Benutzerauthentifizierung über Username und Passwort werden auch SSH-Schlüsselpaare unterstützt: Der Benutzer hinterlegt dazu seinen SSH Public Key auf dem Server und weist beim Login durch Verschlüsseln mit seinem SSH Private Key nach, dass er diesen besitzt. Im Zusammenspiel mit dem Betriebssystem werden auch weitere Authentifizierungsverfahren unterstützt, beispielsweise Kerberos oder Mehrfaktor-Authentifizierung z. B. mit dem Google Authenticator.
- Der SSH Connection Layer bietet Multiplexing an, d.h., es können mehrere logische Kanäle über eine SSH-Verbindung genutzt werden.

Auf dem SSH Transport Layer kommen wie erwartet Chiffren wie AES und Hashfunktionen wie SHA zum Einsatz. Ähnlich zu IPsec Security Associations könnten für die beiden Kommunikationsrichtungen unterschiedliche Kryptoverfahren zum Einsatz kommen, auch wenn dies in der Praxis nur selten gewählt wird.

Das Multiplexing auf dem SSH Connection Layer bietet einige interessante Möglichkeiten, um SSH für mehr als nur den interaktiven Kommandozeilenzugang zu nutzen:

- Über eine SSH-Verbindung können einzelne TCP/IP-Verbindungen sowie das Protokoll X11 für grafische Benutzeroberflächen unter Linux/UNIX getunnelt werden. Implementierungen wie OpenSSL unterstützen auch das SOCKS-Protokoll, mit dem ein dynamisches Forwarding von IP-Verbindungen möglich ist – zu einem beliebigen Internet-Dienst wird die Verbindung also nicht direkt vom Client aus, sondern von der Maschine mit dem SSH-Server aus aufgebaut. In Abschnitt 6.1 gehen wir darauf genauer ein. Mit dieser Funktion kann SSH also als eine Art Mini-VPN genutzt werden, wobei allerdings ohne weitere Maßnahmen nur einzelne Maschinen angeschlossen, aber nicht ganze Netze miteinander gekoppelt werden können.

- Über das SSH File Transfer Protocol (SFTP) und auf SSH aufsetzende Dateisysteme wie SSHFS können SSH-Server als File-Server genutzt werden.

Übung 4.1:

Informieren Sie sich über den „passwortfreien“ Login auf Linux-Servern mittels SSH-Keys. Warum sollte ein SSH Private Key nur verschlüsselt gespeichert werden?

4.2 DNSSEC

Das Domain Name System (DNS) ist ein elementarer Bestandteil des Internets. Sie verwenden es beispielsweise jedes Mal, wenn Sie den Namen eines Webservers im Browser eingeben oder auf einen Link klicken. Bekannt ist DNS insbesondere dafür, „merkbare“ Rechnernamen wie `www.wb-fernstudium.de` auf IP-Adressen abzubilden, zu denen dann auf Schicht 3 des OSI-Referenzmodells Verbindungen aufgebaut werden.

Das DNS ist aber eigentlich eine global verteilte Datenbank, die in Zonen wie Domains und Subdomains strukturiert ist und fast beliebige Einträge aufnehmen kann. Einzelne DNS-Einträge werden dabei als Resource Records (RR) bezeichnet. Über das DNS-Protokoll können diese Einträge abgerufen werden. Dadurch, dass der Datenbestand global verteilt ist, muss man sich aber häufig erst „durchfragen“, bis man bei einem DNS-Server angelangt ist, der eine autoritative Auskunft erteilen kann.

Beispiel 4.1:

Clients stellen ihre DNS-Anfragen wie in Abb. 4.2 gezeigt an einen als Resolver bezeichneten, in der Netztopologie nahe gelegenen DNS-Server, der beispielsweise beim Verbindungsaufbau eines DSL-Anschlusses oder in einem Firmennetz über DHCP zugewiesen wird.

Wenn dieser Resolver z.B. nach der IP-Adresse für `www.wb-fernstudium.de` gefragt wird, kann es sein, dass er die Antwort darauf noch nicht kennt. Er muss dann erst bei den DNS-Root-Servern nachfragen, welche DNS-Server für die Top-Level-Domain `de` zuständig sind. Einen davon kann er dann fragen, welcher DNS-Server für die Domain `wb-fernstudium.de` verantwortlich ist. Von diesem erfragt er schließlich die IP-Adresse für die Maschine mit dem Namen `www` in dieser Domain.

Jeder DNS-Eintrag ist mit einer Angabe in Sekunden verknüpft, wie lange die Antwort in einem Cache gespeichert werden darf, bevor eine neue Anfrage an den verantwortlichen DNS-Server gestellt werden soll. Wenn ein weiterer Client beim selben Resolver nachdem gleichen Eintrag fragt, kann die Anfrage also schneller beantwortet werden.

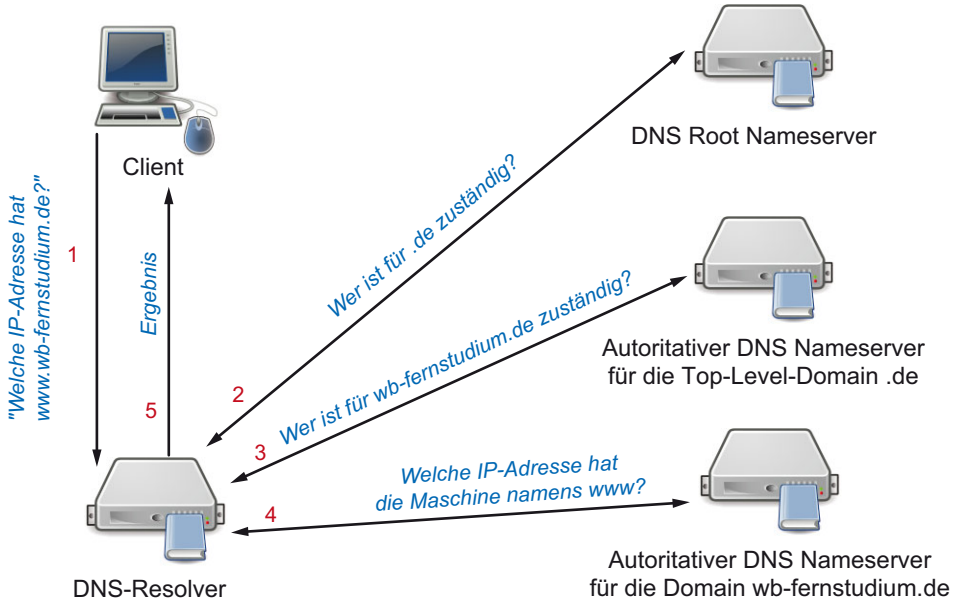

Abb. 4.2: Rekursive Bearbeitung einer DNS-Anfrage durch einen Resolver

Das DNS-Protokoll hat leider das große Problem, dass es UDP-basiert arbeitet und keine sichere Authentisierung von Antwortnachrichten vornimmt. Nehmen wir also an, dass es einem Angreifer gelingt, auf eine DNS-Anfrage ein dazu passendes DNS-Antwortpaket zu schicken, das vor dem richtigen Antwortpaket eintrifft. Der anfragende DNS-Client, also z.B. ein Resolver, nimmt dessen Inhalt dann für bare Münze, speichert ihn in seinem Cache und liefert ihn an seine Clients aus. Der Angreifer hat die Anfrage nach der IP-Adresse eines bestimmten Webservers aber natürlich mit der IP-Adresse eines von ihm kontrollierten Webservers beantwortet. Bei einem solchen DNS-Spoofing-Angriff baut der clientseitige Webbrowser nachfolgend also eine Verbindung zum falschen Webserver auf, ohne dass es dem Benutzer auffällt – zumindest wenn keine weiteren Komponenten wie Serverzertifikate bei HTTPS-Verbindungen genutzt werden.

DNS Spoofing ist aus Angreifersicht erfolgreich, wenn ein Endgerät eine nicht authentische DNS-Antwort weiterverarbeitet. DNS Cache Poisoning tritt ein, wenn ein DNS-Resolver die gefälschten Angaben in seinem Cache speichert und an beliebig viele Clients ausliefert.

DNSSEC ist eine Erweiterung des DNS, die es insbesondere ermöglicht, DNS-Einträge kryptografisch zu signieren. Der Empfänger eines DNS-Antwortpakets kann deshalb mithilfe des Public Keys überprüfen, ob die empfangenen Inhalte tatsächlich von der abgefragten Zone erstellt wurden. Der Public Key wird ebenfalls über DNS zugänglich gemacht und ist, vergleichbar mit der Rolle von Certificate Authorities bei X.509v3-Zertifikaten, seinerseits vom Server der übergeordneten Zone, also z.B. der Top-Level-Domain, signiert.

Durch die kryptografische Signatur stellt DNSSEC somit die Integrität und Authentizität von DNS-Einträgen sicher. DNSSEC umfasst allerdings keine verschlüsselte Datenübertragung. Ein mithörender Angreifer kann also immer noch sehen, nach welchen DNS-Einträgen sich ein Client erkundigt, also durchaus wesentliche Metadaten in Erfahrung bringen.

DNSSEC kann als Mehrwert für andere Dienste aber auch verwendet werden, um weitere Informationen signiert und damit authentisierbar über DNS zugänglich zu machen. Dieses als DNS-based Authentication of Named Entities (DANE) bezeichnete Verfahren kann beispielsweise genutzt werden, um Public Keys z. B. von E-Mail- oder Web-Servern bzw. von E-Mail-Benutzern bereitzustellen. Da solche Einträge nur jemand vornehmen und modifizieren kann, der Schreibzugriff auf die entsprechende DNS-Zone hat, hat DANE das Potenzial, von Certificate Authorities der Global PKI ausgestellte X.509v3-Zertifikate überflüssig zu machen. Bislang wird DNSSEC aber nur von einem Bruchteil aller Domains genutzt, sodass es noch viele Jahre dauern wird, bis die neuen Möglichkeiten flächendeckend eingesetzt werden.

4.3 HTTPS

Wie Sie in Abschnitt 3.5 schon gesehen haben, ist das HyperText Transfer Protocol (HTTP) *das* Kommunikationsprotokoll, für das TLS bzw. sein Vorgänger SSL ursprünglich entwickelt wurde; insofern ist es ein naheliegendes Musterbeispiel für TLS nutzende Protokolle. Das Namensschema, HTTPS durch das Anhängen des Buchstaben `S` als secure-Variante von HTTP zu kennzeichnen, wurde auch von anderen Protokollen übernommen.

HTTPS wird aber nicht nur interaktiv – mit einem Benutzer, der einen Webbrowser verwendet – eingesetzt, sondern dient auch als nahezu universell einsetzbares Protokoll für Client-Server-basierte Programmierschnittstellen. Bei jeder Art von HTTPS-Anwendung ist es essenziell, dass mindestens der Client per Authentisierung prüfen kann, ob er mit dem richtigen Server verbunden ist. Deshalb lohnt sich ein etwas tieferer Blick hinter die Kulissen von HTTPS.

4.3.1 Einsatz von X.509v3-Zertifikaten bei HTTPS-Servern

Bei einer herkömmlichen HTTP-Verbindung über TCP/IP, üblicherweise auf Port 80, schickt der Client eine als HTTP Request bezeichnete Anfrage an den Server, die beispielsweise wie folgt aussehen kann:

```
GET /index.html HTTP/1.1
Host: www.wb-fernstudium.de
```

Der Server antwortet darauf mit einer HTTP Response, die entweder die gewünschte HTML-Datei `index.html` oder eine Fehlermeldung enthält. In der Anfrage ist mit der `Host`-Zeile angegeben, von welchem Webserver die gewünschte Datei bezogen werden soll. Diese Angabe ist erforderlich, weil unter derselben IP-Adresse auf demselben Port mehrere, sogenannte *virtuelle* Webserver betrieben werden können – denken Sie beispielsweise an große Webhosting-Dienstleister, die für Hunderttausende von Kunden jeweils nur relativ wenig angefragte Webserver ressourcenschonend betreiben möchten. Die Adressierung über die IP-Adresse auf Schicht 3 reicht also nicht aus, um den gewünschten Webserver eindeutig zu identifizieren.

Im Zusammenspiel mit TLS steht ein solcher Webserver nun vor folgendem Problem: Der TLS Handshake muss abgeschlossen werden, bevor der HTTP Request mit der `Host`-Zeile vom Client zum Server übertragen wird. Andererseits muss der Server im Rahmen des TLS Handshake das X.509v3-Zertifikat zum Client schicken, damit dieser es überprüfen kann. Aber welches Zertifikat soll ausgeliefert werden, wenn unter derselben IP-Adresse nun Dutzende von virtuellen Webservern betrieben werden?

Dieses Problem zeigt eine recht grundlegende Schwierigkeit beim Design von Kommunikationsprotokollen und Maßnahmen zu ihrer Absicherung: Erst in der Praxis, wenn schon diverse Implementierungen auf dem Markt sind und eingesetzt werden, stellt man fest, dass etwas Wichtiges vergessen wurde. Im konkreten Fall hat man sich damit beholfen, eine TLS-Erweiterung namens Server Name Indication (SNI) einzuführen; damit kann der Client bereits beim TLS-Verbindungsaufbau in der Client-Hello-Nachricht (vgl. Abschnitt 3.5.2) den gewünschten Servernamen mitteilen und bekommt im Rahmen des TLS Handshake das passende Zertifikat geschickt.

4.3.2 Benutzerauthentifizierung durch Webserver

Bei Anwendungen wie Online-Banking hat nicht nur der Benutzer ein Interesse daran, anhand des Server-Zertifikats prüfen zu können, ob er mit dem richtigen Webserver verbunden ist. Auch der Webserver möchte den Benutzer authentifizieren. Im Rahmen des TLS Handshake könnte der Webserver ein Client-Zertifikat anfordern; allerdings sind Client-X.509v3-Zertifikate wenig verbreitet, sodass diese Option meist ausscheidet.

Die HTTP-Spezifikation sieht die sogenannte HTTP Basic Authentication vor. Dabei wird vom Client im HTTP Request eine Zeile der Form

```
Authorization: Basic aWNoOmdlaGVpbQ==
```

eingefügt. Der letzte Bestandteil dieser Zeile enthält einen Benutzernamen und mit einem Doppelpunkt davon getrennt das zugehörige Passwort in der sogenannten Base64-Codierung; dabei kommen also nur 64 statt der von Byte-Werten gewohnten 256 verschiedenenZeichen zum Einsatz. Dennoch handelt es sich nicht um eine Verschlüsselung, sondern nur einen etwas anders dargestellten Klartext. Ohne HTTPS würde also das Passwort im Klartext vom Client zum Server geschickt werden.

Ähnlich verhält es sich, wenn statt HTTP Basic Authentication ein HTML-Formular eingesetzt wird, in das der Anwender seinen Benutzernamen und sein Passwort eintragen soll: Bei reinem HTTP würden die eingegebenen Werte im Klartext zum Server geschickt und könnten somit einfach abgehört werden; erst der Einsatz von TLS führt zu Vertraulichkeit.

Vielleicht nutzen Sie bei einigen Webdiensten auch bereits eine Mehrfaktor-Authentifizierung: Da Passwörter z.B. durch clientseitige Malware oder eine Kompromittierung des Webservers ausgespäht werden können, werden neben dem Passwort verstärkt weitere Identitätsnachweise gefordert. Zunehmend populär, weil kostengünstig, sind insbesondere

- Smartphone-App-basierte Einmalkennwörter z.B. mit dem Google Authenticator. Die Smartphone-App zeigt regelmäßig wechselnde sechsstellige Zahlen an, die man beim Login zusätzlich zu seinem herkömmlichen Passwort angeben muss. Die ange-

zeigte Zahl ist im Wesentlichen ein Hash-Wert, der über einen geheim zu haltenden Startwert und die aktuelle Uhrzeit gebildet wird. Dieses Verfahren wird als Time-based One-Time Password (TOTP) bezeichnet.

- Die 2013 gegründete Vereinigung Fast IDentity Online (FIDO) spezifiziert Hardware und Kommunikationsprotokolle für Zweifaktor-Authentifizierung, die von aktuellen Betriebssystemen und Browsern bereits unterstützt werden. Anwender müssen ein nur wenige Euro teures Hardware-Token z. B. über die USB-Schnittstelle oder Bluetooth verwenden, das für jede Webseite ein eigenes Public-/Private-Key-Schlüsselpaar erstellt. Dem Benutzer bleibt dabei die softwareseitige Handhabung von X.509v3-Client-Zertifikaten erspart, und weil der Private Key das Hardware-Token nie verlässt, kann er durch Malware auf dem PC nicht kompromittiert werden.

Um zu vermeiden, dass Benutzer-Authentifizierungsinformationen bei jedem HTTP Request neu mitgeschickt werden müssen, schickt der Webserver nach erfolgreichem Login typischerweise ein HTTP Cookie an den Client. Dabei handelt es sich um kleine Datensätze, die der Client bei nachfolgenden Anfragen automatisch zusammen mit dem eigentlichen HTTP Request wieder an den Webserver schickt. Der Webserver wertet das Cookie aus und kann den HTTP Request so einem bereits eingeloggten Benutzer zuordnen. Viele Frameworks zum Programmieren von Webanwendungen unterstützen dies unter der Bezeichnung Session Management, deren Benutzeridentifikatoren aber nicht mit den Session-Ids des TLS Handshake verwechselt werden dürfen.

4.3.3 Angriffe auf HTTPS und TLS

Durch den engen Zusammenhang von TLS und HTTPS dienen Angriffe auf HTTPS-Verbindungen gerne als Beispiele, wenn es sich bei einer Sicherheitslücke eigentlich um einen Design- oder Implementierungsfehler von TLS handelt. Als Ziel des Angreifers wird dabei oft formuliert, dass vom Angreifer zumindest Teile einer TLS-geschützten Nachricht entschlüsselt werden können. Bevorzugt sollen dabei die HTTP-Cookies, die der Client an den Server schickt, im Klartext abgegriffen werden können, da sich der Angreifer damit dem Webserver gegenüber als der legitime Benutzer ausgeben kann (engl. *Cookie Theft Impersonation Attack*).

In den letzten Jahren haben Sie sicher auch schon in der Tagespresse wiederholt über Sicherheitslücken in TLS bzw. HTTPS gelesen – aufgrund der Tragweite gerade im E-Commerce ist das Thema längst nicht mehr nur für Fachpublikum relevant.

Beispiel 4.2:

Einige besonders gravierende Beispiele sind:

- BEAST (2011): Alle Versionen von SSL bis zu TLS 1.0 haben beim Einsatz von DES- und AES-Verschlüsselung den ersten Datenblock unzureichend verschlüsselt, sodass Angreifer erkennen konnten, wann derselbe Klartext erneut verschlüsselt übertragen wurde. Durch das Einfügen präparierter Klartext-Blöcke konnte damit das Session-Cookie Byte für Byte ermittelt werden.
- POODLE (2014): Der vom Google Security-Team beschriebene POODLE-Angriff ermöglicht es einem Man-in-the-Middle-Angreifer, Client und Server dazu zu bringen, eine SSL 3.0-Verbindung aufzubauen, obwohl beide Seiten neuere TLS-Versionen unterstützen würden. Bei SSL 3.0 sind wiederum Angriffe möglich, um u. a. das Session-Cookie im Klartext zu ermitteln.

- Heartbleed (2014): Der Heartbleed-Angriff nutzte einen Programmierfehler in der populären Open-Source-Implementierung OpenSSL aus, um RAM-Inhalte des Servers auszulesen; damit konnte z. B. der Private Key des Servers ausgespäht werden.
- FREAK (2015): Ähnlich zu POODLE konnten durch einen Man-in-the-Middle-Angriff die beiden Kommunikationspartner dazu gebracht werden, die RSA-Chiffre mit lediglich 512 Bit langen und damit unsicheren Schlüsseln zu verwenden.

Sicherlich werden im Lauf der Jahre noch weitere Sicherheitslücken gefunden und behoben werden. Wir müssen uns deshalb zwei wesentliche Probleme vergegenwärtigen:

a) Auch wenn mathematisch fundierte, als sicher geltende Kryptoverfahren eingesetzt werden, ist das Design von sicheren Kommunikationsprotokollen schwierig.

b) Selbst ein gutes Protokolldesign schützt nicht davor, dass Implementierungen fehlerhaft sind und der komplette Schutz ausgehebelt werden kann.

Auf Bruce Schneier geht die Aussage „*security is a process, not a product*“ zurück.[2] Unabhängig davon, ob Sie Protokolle wie TLS in selbst entwickelter Software einsetzen oder für die Konfiguration eines Webservers verantwortlich sind, wird es also nie ausreichen, Sicherheitsparameter wie Chiffren und Schlüssellängen nur einmalig einzustellen. Im Rahmen der Software- und Systemwartung muss immer auch sichergestellt werden, dass alle beteiligten Security-Komponenten und deren Konfiguration auf dem aktuellen Stand der Technik bleiben!

4.4 Sicherer Versand und Abruf von E-Mails

In Abschnitt 2.3 haben wir uns bereits angesehen, wie durch Verschlüsseln und Signieren eine Ende-zu-Ende-Sicherheit bei E-Mail-Korrespondenz erreicht werden kann. Wir haben aber auch festgestellt, dass diese Verfahren noch bei Weitem nicht flächendeckend eingesetzt werden: Fast alle E-Mails weltweit werden unverschlüsselt versandt.

Die Zustellung von E-Mails läuft über ein klassisches Store and Forward-Protokoll: Alice schickt eine E-Mail nicht direkt zu Bobs Rechner, sondern zunächst an den Mailserver, der für ihre eigene E-Mail-Adresse als ausgehender Mailserver zuständig ist. Dieser Mailserver ermittelt dann den für die Empfänger-E-Mail-Adresse zuständigen Mailserver und schickt die E-Mail weiter. Dort bleibt sie dann mindestens so lange liegen, bis sie von Bob irgendwann abgerufen und gelöscht wird.

Die Administratoren der Mailserver haben also unvermeidbar Zugriff auf den E-Mail-Inhalt; ohne E-Mail-Verschlüsselung auf Schicht 7 kann keine Ende-zu-Ende-Sicherheit erreicht werden. Zumindest setzt sich aber für die Kommunikation mit und zwischen Mailservern durch, dass die Übertragung abgesichert wird. Dies betrachten wir im Folgenden für die drei in Abb. 4.3 dargestellten Teilstrecken, die eine E-Mail zurücklegt, genauer.

2. https://www.schneier.com/crypto-gram/archives/2000/0515.html.

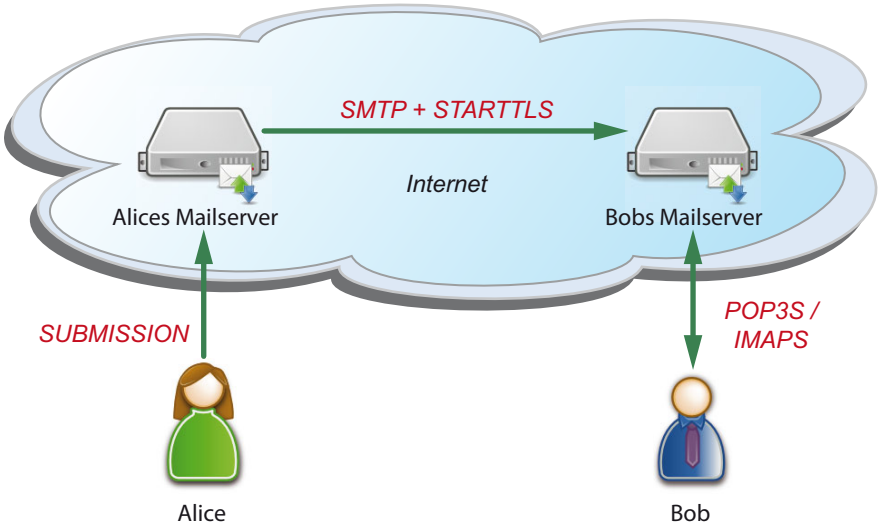

Abb. 4.3: Protokolle zum abschnittsweise gesicherten E-Mail-Transport

4.4.1 Sicherer E-Mail-Versand durch Benutzer

Im ersten Schritt muss der Absender einer E-Mail diese an einen Mailserver übergeben, der für ausgehende E-Mails der Absender-DNS-Domäne, also dem Teil rechts vom `@`-Zeichen der E-Mail-Adresse, offiziell zuständig ist. Damit soll verhindert werden, dass z. B. Spam-E-Mails von beliebigen Rechnern aus mit gefälschten Absenderadressen verschickt werden können. Dieser Mailserver sollte also nur von legitimen Benutzern genutzt werden können. In Firmen ist der Mailserver deshalb häufig nur im LAN bzw. über VPN erreichbar; mindestens die über das Internet allgemein erreichbaren E-Mail-Server führen eine Überprüfung von Benutzername und Passwort durch, damit sie nicht für den Spam-Versand durch Dritte missbraucht werden können.

Klassisch läuft die Kommunikation mit einem Mailserver zum Zweck des Mailversands über das Simple Mail Transfer Protocol (SMTP, TCP/IP Port 25). Wie viele andere Protokolle aus den Urzeiten des Internets ist es textbasiert und unverschlüsselt.Wie Sie wahrscheinlich schon von der Konfiguration Ihres E-Mail-Clients kennen, gibt es mittlerweile Abhilfe auf Basis von TLS:

- SMTPS (TCP/IP, Port 465) war der erste Ansatz, um das SMTP-Protokoll analog zum Übergang von HTTP zu HTTPS auf Basis von SSL bzw. TLS abzusichern. Offiziell ist SMTPS schon seit 1998 veraltet, da stattdessen empfohlen wurde, eine herkömmliche, zunächst unverschlüsselte SMTP-Verbindung auf Port 25 aufzubauen und dort das Kommando `STARTTLS` abzusetzen, mit dem ein TLS Handshake angestoßen wird.
- Moderne Mailserver verwenden für die *Message Submission* durch ihre Anwender den TCP-Port 587. Die Idee dahinter ist, Kommunikation auf TCP-Port 25 (SMTP) für die Kommunikation zwischen Mailservern zu reservieren und neu zu versendende E-Mails von Benutzern auf einem dedizierten Port entgegenzunehmen.

Für die Authentifizierung des Benutzers, der eine E-Mail verschicken möchte, kommt SMTP-Auth, eine Erweiterung des SMTP-Protokolls, zum Einsatz. Damit kann eine Überprüfung des Benutzernamens, der oft gleich der Absende-E-Mail-Adresse ist, und

des zugehörigen Passworts vorgenommen werden. Da die Verbindung TLS-gesichert ist, kann der Versender anhand des X.509v3-Server-Zertifikats sicher sein, mit dem richtigen Mailserver verbunden zu sein, bevor er sein Passwort schickt.

4.4.2 Sichere Kommunikation zwischen Mailservern

Wenn ein Mailserver eine E-Mail entgegengenommen hat, die keinem seiner eigenen Benutzer zugestellt werden soll, hat er die Aufgabe, sie zum richtigen Mailserver des Empfängers zu übertragen. Über das DNS-Protokoll (siehe Abschnitt 4.2) fragt er dazu den sogenannten Mail-Exchange-Record (MX) der DNS-Domäne ab, die aus der Empfänger-E-Mail-Adresse hervorgeht. Zu dem so ermittelten Mailserver würde er im klassischen Fall eine unverschlüsselte SMTP-Verbindung aufbauen. Seit vielenJahren gilt als Stand der Technik aber eine TLS-gesicherte SMTP-Verbindung.

Beispiel 4.3:

Das Bayerische Landesamt für Datenschutzaufsicht hat bereits 2014 eine Untersuchung der Mailserver von über 2000 Unternehmen mit Sitz in Bayern durchgeführt. Dabei wurde überprüft, ob die Mailserver das `STARTTLS`-Kommando unterstützen, Perfect Forward Secrecy verwendet wird und ob die Heartbleed-Lücke gepatcht wurde. Über 700 Unternehmen wurden aufgefordert, identifizierte Defizite zu beseitigen.

Leider hat der Ansatz, sowohl für unverschlüsselte als auch per `STARTTLS` auf Verschlüsselung umgestellte Protokolle denselben TCP/IP-Port zu verwenden, einen gravierenden Nachteil: Ein Man-in-the-Middle-Angreifer, über den sämtliche Datenpakete laufen, könnte auf das `STARTTLS`-Kommando einfach antworten, dass die Gegenseite dieses Verfahren nicht unterstützt. Damit die E-Mail überhaupt zugestellt werden kann, bleibt es dann bei einer unverschlüsselten Verbindung. Hierfür gibt es folgende Lösungsansätze:

- Sehr pragmatisch, aber durchaus bewährt ist das TOFU-Prinzip: *Trust On First Use*. Man geht davon aus, dass bei der allerersten Verbindung kein Man-in-the-Middle-Angreifer aktiv ist, und merkt sich, ob die Gegenseite `STARTTLS` unterstützt. Falls dies der Fall ist, verwendet man auch zukünftig nur TLS-gesicherte Verbindungen bzw. bricht die Verbindung ab und schlägt Alarm, wenn `STARTTLS` auf einmal nicht mehr unterstützt wird.
- Neuer und noch nicht flächendeckend verbreitet ist die Verwendung von DNSSEC und DANE (vgl. Abschnitt 4.2). Dabei legt ein per DNSSEC abgesicherter DNS-Eintrag fest, ob zu einem Mailserver eine TLS-Verbindung aufgebaut werden soll.

Beispiel 4.4:

Im Rahmen der Initiative *E-Mail Made in Germany*, an der mehrere große deutsche E-Mail-Provider beteiligt sind, wurden 2014 alle Verbindungen zwischen den beteiligten Mailservern auf TLS-Verschlüsselung umgestellt. Während die Initiative von einigendafür kritisiert wurde, dass dies eigentlich nur dem üblichen Stand der Technik entsprach, zeigt es doch, dass eine Umstellung darauf in der Praxis mit größerem Aufwand verbunden sein kann; insgesamt ergab sich für die beteiligten Provider eine sehr positive Außendarstellung.

Auch wenn es mit der Sicherung der Netzwerkkommunikation zwischen zwei Mailservern nicht direkt zu tun hat, ist beachtenswert, dass TLS-gesicherte SMTP-Verbindungen nur einen kleinen Teil des Problemraums darstellen. Noch wesentlich aufwendiger für den empfangenden Mailserver ist der Schutz vor Spam-E-Mails: Neben Blacklists für als „Spam-Schleudern" bekannte Quell-Mailserver kommt meist eine Kombination mehrerer, in Teilen wiederum DNS- und DNSSEC-basierter Protokolle wie SPF, DKIM und DMARC zum Einsatz. Diese sollen sicherstellen, dass der Quell-Mailserver wirklich berechtigt ist, E-Mails mit der angegebenen Absenderadresse zu verschicken.

4.4.3 Sicherer Abruf von E-Mails durch Benutzer

Die von Alice geschickte E-Mail liegt nun auf Bobs Mailserver und wartet darauf, gelesen zu werden. Sicherlich haben Sie in Ihrem Bekanntenkreis Personen, die E-Mails nur in geringem Umfang verwenden und sich z. B. eine kostenlose E-Mail-Adresse bei einem der großen Freemail-Provider angelegt haben. Dieser Benutzerkreis greift gerne per Web-Interface auf seine E-Mails zu. Der E-Mail-Dienstleister betreibt also einen Webserver, über den E-Mails gelesen und geschrieben werden können. Dafür kommt hoffentlich HTTPS mit einer ausreichend starken Benutzerauthentifizierung zum Einsatz, wie wir es in Abschnitt 4.3 schon kennengelernt haben. Erfreulicherweise bieten solche Provider auch vermehrt Browser-Plugin- oder JavaScript-basierte Lösungen an, um E-Mails wie in Abschnitt 2.3 beschrieben Ende-zu-Ende abzusichern, ohne dass der eigene Private Key beim Provider gespeichert werden muss.

Bei intensiverer E-Mail-Nutzung ist hingegen die Verwendung eines dedizierten E-Mail-Clients üblich. Damit Bob die E-Mails von seinem Mailserver abrufen kann, kommt in der Regel eine der folgenden Varianten zum Einsatz:

- POP3(S): Das Post Office Protocol in Version 3 (POP3, TCP-Port 110) ist der Klassiker unter den Protokollen für den E-Mail-Abruf. Nachdem sich der Benutzer mit Username und Passwort authentifiziert hat, kann er eingegangene E-Mails vom Server abrufen und löscht sie dort anschließend üblicherweise. POP3S (TCP-Port 995) entspricht einem TLS-gesicherten POP3, wobei viele POP3-Server inzwischen auch auf TCP-Port 110 das `STARTTLS`-Kommando unterstützen; wie beim Protokoll SMTP kann ein Man-in-the-Middle-Angreifer das Umschalten per `STARTTLS` aber einfach und möglicherweise unbemerkt verhindern.
- IMAP(S): Das Internet Message Access Protocol in Version 4 (IMAP, TCP-Port 143) ist das heute am weitesten verbreitete Protokoll zum Zugriff auf E-Mails. Anders als POP3 unterstützt es nicht nur das Herunterladen von E-Mails, sondern z. B. auch das Anlegen von Ordnern zur strukturierten Ablage von E-Mails auf dem Server und serverseitige Suchfunktionen. Die wesentlichen Vorteile der serverseitigen E-Mail-Speicherung sind zentral durchgeführte Backups und die Möglichkeit, mit mehr als einem Client auf dieselben E-Mails zugreifen zu können – nicht nur mit dem einen Gerät, mit dem man die E-Mail per POP3 heruntergeladen hat. IMAPS (TCP-Port 993) ist schlichtweg eine TLS-gesicherte Version von IMAP.
- Proprietäre Protokolle: Insbesondere Groupware-Lösungen, die neben E-Mails auch z. B. Kalender mit Raumbuchungen und Aufgabenverwaltung integrieren, liefern eigene Client-Software aus, die über proprietäre Protokolle mit den E-Mail- bzw. Groupware-Servern kommuniziert.

Beispiel 4.5:

Ein weitverbreitetes Groupware-Produkt ist Microsoft Exchange: Als Client kommt das Programm Microsoft Outlook zum Einsatz, das über Microsofts MAPI-Schnittstelle (Messaging Application Programming Interface) oder EWS (Exchange Web Services) auf die E-Mails zugreift.

4.5 Secure Messaging und Online-Chat

Früher kamen für den Nachrichtenaustausch meist E-Mails zum Einsatz und textbasierte Online-Gruppendiskussionen wurden überwiegend per Internet Relay Chat (IRC) abgewickelt. Mit der Verbreitung von Smartphones hat aber auch der Einsatz von Instant-Messaging-Apps wie WhatsApp massiv zugenommen. Während diese ursprünglich als kostensparende Alternative zum Versand von SMS-Nachrichten beliebt wurden, unterstützen sie mittlerweile längst auch Gruppen-Chats und zum Teil auch zusätzliche Funktionen wie Telefonie und Dateiaustausch.

Instant-Messaging-Protokolle sind ein gutes Beispiel dafür, dass zugunsten einer schnellen Markteinführung und umfangreicher Funktionalität das Thema Sicherheit gerne hintangestellt wird: Kaum ein Messaging-Dienst hat initial verschlüsselte und integritätsgesicherte Verbindungen angeboten, erst recht keine Ende-zu-Ende-Sicherheit: Die Apps auf Absender- und Empfängerseite haben zwar zum Teil TLS-gesichert mit einem zentralen Server kommuniziert, dieser hatte aber vollen Zugriff auf den Klartext der ausgetauschten Nachrichten.

Im Folgenden betrachten wir zwei Protokolle für Secure Messaging und Online-Chat, die auch Ende-zu-Ende-Sicherheit bieten: *Signal* entwickelt sich zum De-facto-Standard beim Messaging auf mobilen Endgeräten und *XMPP* (früher Jabber) wird auch von einer Vielzahl von Chat-Programmen für Desktop-PCs unterstützt.

4.5.1 Protokoll Signal

Messaging-Anwendungen, beispielsweise für Smartphones, müssen mit folgender Situation umgehen: Die beiden (bei Gruppen-Chats noch mehr) Kommunikationspartner sind nicht notwendigerweise gleichzeitig online. Zudem kennen sie zumindest am Anfang z.B. ihre gegenseitigen IP-Adressen nicht und können allein schon deshalb keine direkte IP-Verbindung zueinander aufbauen. Deshalb kommt fast zwangsweise ein zentraler Server oder ein ganzes Netz von Vermittlungssystemen zum Einsatz, um Nachrichten zwischenzuspeichern und dem Empfänger baldmöglichst zuzustellen. Die Serverbetreiber erlangen im Regelfall also mindestens Kenntnis über Kommunikationsmetadaten – wer wann und mit wem kommuniziert; bei Ende-zu-Ende-Verschlüsselung bleibt aber zumindest der Nachrichteninhalt vertraulich.

Das Protokoll Signal wurde ursprünglich für die gleichnamige Messaging-App entworfen und stellt eine Weiterentwicklung der Vorgängerprotokolle *Off-the-Record-Messaging und TextSecure* dar; es wird inzwischen auch von anderen Apps wie WhatsApp, Facebook Messenger und Google Allo eingesetzt.

Signal verwendet AES zur Verschlüsselung und SHA-256 (entspricht SHA-2 mit 256 Bit langer Ausgabe) zur Integritätssicherung. Eine Besonderheit besteht bei Signal in dem Verfahren, mit dem die entsprechenden Schlüssel dafür erzeugt werden. Grundlegend

kommt dabei die Diffie-Hellman-Variante mit elliptischen Kurven zum Einsatz (ECDH, siehe Abschnitt 2.2.2). Signal zielt dabei aber darauf ab, die verwendeten Schlüssel häufig zu erneuern; damit soll nicht nur Perfect Forward Secrecy erreicht werden (siehe Abschnitt 2.2.4), sondern es sollen auch die Folgen einer Schlüsselkompromittierung für zukünftig ausgetauschte Nachrichten minimiert werden (*future secrecy*).

Das entsprechende Verfahren haben die Entwickler *Double Ratchet* genannt: Wenn Alice an Bob eine Nachricht gesendet hat und Bob darauf mit einer oder mehreren Nachrichten antwortet, erneuert er den verwendeten Schlüssel durch einen Diffie-Hellman-Schlüsselaustausch. Dabei kombiniert er den eigenen neuen Schlüssel mit dem bereits bekannten Schlüssel von Alice, sodass das Verfahren auch funktioniert, wenn Alice gerade nicht online ist. Sobald genügend Nachrichten für das gesamte Diffie-Hellman-Verfahren zwischen Alice und Bob ausgetauscht wurden, wird auf den so ausgehandelten neuen Schlüssel gewechselt. Damit dieses Verfahren anlaufen kann, also bereits die allerersten Nachrichten zwischen Alice und Bob abgesichert werden können, hinterlegt Bob mit seinem *Identity Public Key* signierte Schlüssel auf dem Server, die Alice als *Prekey Bundle* abrufen kann. Ob es sich um den richtigen Identity Public Key von Bob handelt, muss Alice allerdings anderweitig, also *out-of-band*, überprüfen.

Als weitere Besonderheit wird seit dem *Off-the-Record-Messaging*-Protokoll ganz bewusst Abstreitbarkeit erreicht: Die zur Integritätssicherung verwendeten Signaturschlüssel werden nachträglich veröffentlicht. Beim Empfang der Nachricht kann der Empfänger die Integrität also noch zuverlässig überprüfen; nach Veröffentlichung des verwendeten Schlüssels hätte aber jeder die Nachricht erstellen und so signieren können. Es soll also kein Kommunikationspartner später nachweisen können, dass bestimmte Inhalte übertragen wurden.

4.5.2 Jabber-Protokoll XMPP

Ende der 1990er-Jahre wurde Instant Messaging über PC-Anwendungen durchaus beliebt. Haben Sie in Ihrer Kindheit vielleicht Programme wie ICQ, Yahoo Messenger oder Windows Live Messenger verwendet? In vielen Unternehmen ist ein solcher Online-Chat zwischen den Beschäftigten als „schlanke“ Alternative zu E-Mail-basierten Diskussionen heute durchaus gewünscht. Damit keine Interna über externe Anbieter laufen, werden aber gerne Protokolle eingesetzt, für die man eigene Chat-Server betreiben kann.

Besonders weit verbreitet ist dabei das Extensible Messaging and Presence Protocol (XMPP), das vor seiner Standardisierung durch die IETF als Jabber-Protokoll bekannt war. Es verfolgt einen Ansatz, der sehr ähnlich zur globalen E-Mail-Infrastruktur ist (vgl. Abschnitt 4.4):

- Die clientseitigen Chat-Anwendungen der Benutzer verbinden sich mit einem XMPP-Server. Diese Verbindungen sind typischerweise TLS-gesichert. Benutzer müssen sich i. d. R. per Passwort authentifizieren; die Benutzernamen sehen wie E-Mail-Adressen aus, der Teil rechts vom `@`-Zeichen identifiziert aber den verwendeten XMPP-Server.
- Der XMPP-Server leitet eine verschickte Nachricht entweder an einen anderen XMPP-Server weiter, falls sie für keinen seiner eigenen Benutzer bestimmt ist; oder er übergibt sie dem Empfänger, sobald dieser online, also ebenfalls mit dem XMPP-Server verbunden ist. Für die Verbindungen zwischen XMPP-Servern kommt inzwischen ebenfalls meist TLS-Sicherung zum Einsatz.

Ein XMPP-Server kann also entweder isoliert, z.B. nur firmenintern, betrieben oder in einen globalen Verbund integriert werden. Wie bei E-Mails stellt sich dabei das Problem, dass zwar die einzelnen Übertragungsabschnitte TLS-gesichert werden können, die beteiligten Server die Nachrichten aber im Klartext vorliegen haben.

Um eine Ende-zu-Ende-Sicherheit zu erreichen, lässt sich die flexible Erweiterbarkeit von XMPP – das `x` steht ja für *extensible* – nutzen; in mehreren XMPP-Clients implementiert sind insbesondere

- OpenPGP: Wie bei E-Mails (vgl. Abschnitt 2.3.2) können damit Nachrichten signiert und verschlüsselt werden, es wird also insbesondere auch Nichtabstreitbarkeit gewährleistet.
- Off-the-Record-Messaging: Wie beim oben beschriebenen Signal-Protokoll wird damit neben Verschlüsselung trotz Integritätssicherung auch Abstreitbarkeit ermöglicht.

Über die XMPP-Erweiterung *Jingle* kann zudem eine direkte Verbindung zwischen zwei XMPP-Clients initiiert werden; sie kommt zum Einsatz, wenn keine kurzen Textnachrichten, sondern umfangreichere Datenströme für Audio und Video übertragen werden sollen. Jingle verhält sich dabei ähnlich wie das für Voice-over-IP-Telefonie eingesetzte SIP-Protokoll, das wir uns als Nächstes ansehen.

4.6 20. Voice-over-IP-Telefonie

Voice over IP (VoIP), also das Abwickeln von Telefonaten über ein Datennetz bzw. das Internet, ist in mehrfacher Hinsicht aus Kostengründen attraktiv: Wenn die Gesprächspartner global verteilt sind, aber VoIP unterstützen, fallen neben dem sowieso schon vorhandenen Internetzugang keine Kosten für einzelne Telefonate an, wie man sie bei einem herkömmlichen Telefonanschluss kennt. Auch die Infrastrukturkosten – beispielsweise der Hausanschluss von Neubauten und die Etagenverkabelung vieler Büros – reduzieren sich, wenn keine zusätzlichen Telefonie-Kupferkabel verlegt werden müssen.

Für internetbasierte Sprachkommunikation gibt es verschiedenste Ansätze. Oben haben wir schon gesehen, dass Messaging-Anwendungen um Sprach- und Videoübertragung erweitert wurden. Für speziellere Anwendungen wie Gruppengespräche bei Onlinespielen existieren proprietäre Protokolle wie TeamSpeak und offene Alternativen wie Mumble. Ein wichtiger Aspekt dabei ist, dass die Datenübertragung in diesen Fällen bevorzugt über UDP-Pakete, also nicht über TCP erfolgt: Einzelne verlorene UDP-Pakete stören die vommenschlichen Empfänger wahrgenommene Qualität weniger als die Verzögerungen, die sich bei TCP durch einen erneuten Versand verlorener Pakete ergeben würde. TLS ist aber auf TCP/IP-Verbindungen ausgelegt (siehe Abschnitt 3.5). Deshalb wird beispielsweise auch vom kommerziellen Videokonferenzsystem Cisco WebEx zunächst über eine TLS-Verbindung Schlüsselmaterial ausgehandelt, mit dem anschließend die UDP-basiert übertragenen Nutzdaten verschlüsselt und signiert werden.

Im Folgenden betrachten wir die IETF-Standards Session Initiation Protocol (SIP) und Real-Time Transport Protocol (RTP). SIP und RTP bilden die am weitesten verbreitete Grundlage für VoIP; über sogenannte Breakout- und Breakin-Verbindungen kann auch mit Teilnehmern in öffentlichen Telefonnetzen telefoniert werden. Die Betreiber dieser

Schnittstellen zu den öffentlichen Telefonnetzen verlangen üblicherweise Gesprächsgebühren und müssen gesetzliche Vorgaben, z. B. zur Überwachung und zur Durchführung von Notrufen, einhalten.

Beispiel 4.6:

Wenn Sie zu Hause einen DSL-Anschluss mit „Internet-Telefonie“ haben, kommen SIP und RTP zum Einsatz.

4.6.1 Session Initiation Protocol

Ein Telefonat bzw. eine Videokonferenz lässt sich grob in die drei Phasen Verbindungsaufbau, Gesprächsübertragung und Verbindungsabbau einteilen. SIP ist für die erste und die letzte Phase zuständig, dient also der sogenannten Signalisierung. Es ist wie HTTP ein textbasiertes Protokoll und weist den beteiligten Geräten folgende Rollen zu:

- Der *User Agent* ist die Schnittstelle zum Anwender, also beispielsweise ein SIP-fähiges Telefon oder eine entsprechende Software (Softphone). Teilnehmer werden durch Adressen im URI-Format beschrieben, z.B. `sips:username@domain`für TLS-gesicherte SIP-Verbindungen oder `tel:+49-123-456789` für Rufnummern im öffentlichen Telefonnetz.
- Ein *Proxy Server* sorgt ähnlich wie ein Mailserver dafür, dass SIP-Nachrichten im Auftrag eines Absenders näher zum Empfänger transportiert werden, übernimmt also Routing-Aufgaben. Dabei kann insbesondere auch überprüft werden, ob die SIP-Nachricht weitergeleitet werden soll, also z. B. ob ein Benutzer berechtigt ist, einen bestimmten Anruf zu tätigen.
- Der *Registrar Server* ist diejenige SIP-Komponente, an der sich User Agents anmelden. Durch einen erfolgreichen `REGISTER`-Request wird insbesondere eine Zuordnung einer Teilnehmer-Adresse, also einer SIP-Telefonnummer, zur aktuellen IP-Adresse des User Agent hergestellt. Diese wird benötigt, um SIP-Nachrichten wie eingehende Anrufe zustellen zu können. In der Praxis sind der pro Benutzer verwendete Proxy und Registrar meist derselbe Server.
- Ein *Gateway* ist eine Schnittstelle zu anderen Netzen wie dem öffentlichen Telefonnetz.

SIP (TCP- oder UDP-Port 5060) wird in der TLS-gesicherten Variante als SIPS (TCP- oder UDP-Port 5061) bezeichnet. Wenn Alice einen VoIP-Anruf bei Bob tätigen möchte, schickt ihr User Agent eine `INVITE`-Nachricht für Bobs Teilnehmer-Adresse an ihren eigenen SIP-Proxy. Dieser gibt die Nachricht, ggf. nach Berechtigungsprüfung, weiter an denjenigen SIP-Proxy, der für die in Bobs Teilnehmer-Adresse genannte Domain zuständig ist. Wenn Bob dort registriert, also gerade online ist, kann die Nachricht anBobs User Agent zugestellt werden.

Anders als bei herkömmlicher analoger oder ISDN-Telefonie müssen sich die User Agents über SIP zunächst auf die konkreten Verbindungsmodalitäten einigen. Beispielsweise können verschiedene Audio-Codecs mit unterschiedlichen Qualitätsstufen und Bandbreitenanforderungen eingesetzt werden oder z. B. auch Videoübertragung aktiviert werden. Zu diesem Zweck werden Nachrichten gemäß Session Description Protocol (SDP) in die SIP-Nachrichten eingebettet. Über die SDP-Nachrichten einigen sich die

User Agents von Bob und Alice auf die zu verwendenden Codecs, teilen sich ihre IP-Adressen gegenseitig mit und legen ggf. auch den Schlüssel für eine symmetrische Chiffre zur Verschlüsselung des Gesprächs fest.

Übung 4.2:

Welche Nachteile hat es, wenn Alice einen von ihr festgelegten Schlüssel für eine symmetrische Chiffre per SIP-/SDP-Nachricht an Bob überträgt?

Für die Übertragung des eigentlichen Gesprächs wird das Protokoll RTP bzw. SRTP verwendet, das wir uns im nächsten Abschnitt anschauen. Die über die SIP-Server vermittelten SIP-Nachrichten dienen also wie in Abb. 4.4 dargestellt primär dazu, diese direkte RTP-Verbindung zwischen den Gesprächspartnern zu initiieren. Über eine erneute `INVITE`-Nachricht können die Parameter der Gesprächsübertragung aber jederzeit geändert werden. Zum Ende des Gesprächs wird von der Seite, die zuerst „auflegt“, eine `BYE`-Nachricht zur Gegenseite geschickt.

Beispiel 4.7:

Die Trennung in SIP für die Signalisierung und RTP für die Gesprächsübertragung stellt eine gewisse Schwachstelle dar: Nehmen wir an, dass ein VoIP-Telefonat z.B. auf Basis seiner Dauer in ganzen Minuten abgerechnet werden soll. Manipulierte User Agents könnten über SIP eine Verbindung aufbauen und, sobald die RTP-Verbindung eingerichtet wurde, gleich eine `BYE`-Nachricht schicken. Aus Sicht der SIP-Proxies ist das Telefonat dann beendet, obwohl die RTP-Verbindung noch weiterläuft.

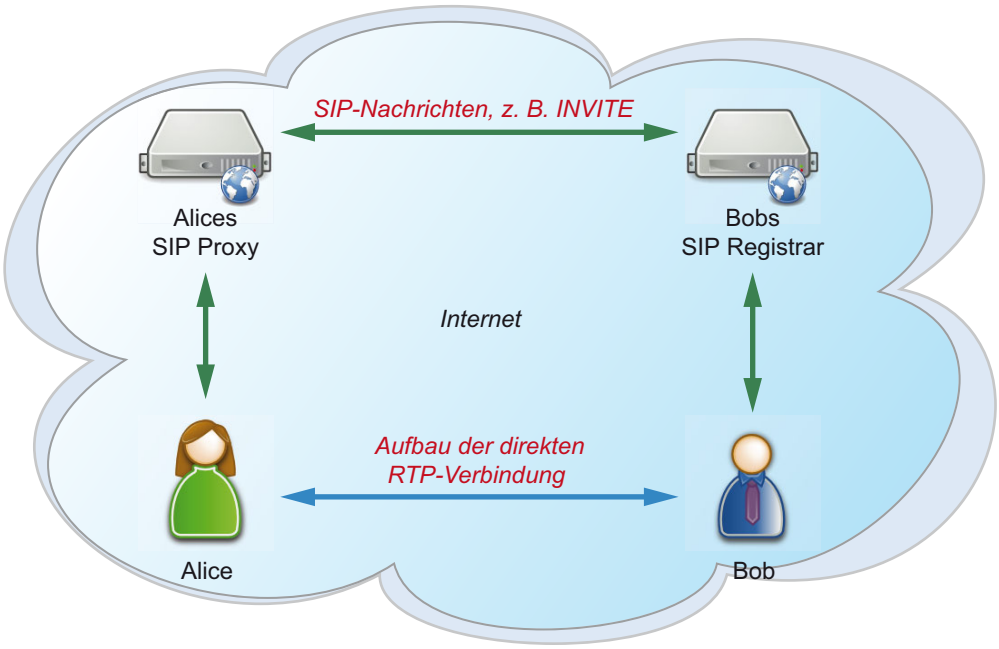

Abb. 4.4: Austausch von SIP-Nachrichten zum Aufbau einer RTP-Verbindung

4.6.2 Real-Time Transport Protocol

Das Real-Time Transport Protocol überträgt die Gesprächsdaten UDP-basiert direkt zwischen den User Agents, also nicht über die SIP Proxies. Für RTP ist kein fester Port reserviert, sondern es wird ein gerade freier Port verwendet, der im Rahmender SDP-Nachricht genannt wurde, die ihrerseits in der SIP-`INVITE`-Nachricht enthalten ist. Durch diese dynamische Portwahl können sich Verbindungsprobleme z. B. durch Paketfilter-Firewalls oder NAT-Gateways auf Empfängerseite ergeben, die den Port nicht freischalten bzw. weiterleiten. In diesem Fall müssen z. B. die Session Traversal Utilities for NAT (STUN, IETF RFC 5389) eingesetzt werden; dabei baut der Empfänger zuerst eine Verbindung von sich aus nach außen auf, damit er über den entsprechenden Port anschließend auch Datenpakete von außen empfangen kann.

Zur Verschlüsselung und Integritätssicherung von RTP kann Secure RTP (SRTP) eingesetzt werden. Es setzt sich erst allmählich durch, da viele VoIP-Tischtelefone der ersten Gerätegenerationen es noch nicht implementiert hatten und selbst bei der Möglichkeit, Firmware-Upgrades durchzuführen, die Rechenleistung nicht ausreichend ist. Da VoIP-Telefone aber noch recht teuer sind, fehlt bei vielen Unternehmen die Bereitschaft, große Stückzahlen an VoIP-Telefonen nach nur wenigen Jahren Nutzung nur der Verschlüsselung wegen komplett zu erneuern.

SRTP ist durch seine Spezifikation auf AES-Verschlüsselung und SHA-1-Prüfsummen festgelegt. Die verwendeten Schlüssel werden mithilfe einer *Key Derivation Function* (KDF) aus einem *Master Key* abgeleitet; diese KDF wird auch zur regelmäßigen Ableitung neuer Schlüssel verwendet und erzielt Perfect Forward Secrecy (vgl. Abschnitt 2.2.4). Wesentlich ist also das Verfahren, mit dem der Master Key festgelegt wird. Nennenswerte Verbreitung haben bislang folgende Verfahren:

- SDES (Session Description Protocol Security Descriptions, IETF RFC 4568): In diesem einfachsten Fall wird der zu verwendende Schlüssel in den SDP-Teil der SIP-`INVITE`-Nachricht geschrieben. Dies hat die in Übung 4.2 behandelten Nachteile.
- MIKEY (Multimedia Internet KEYing, IETF RFC 3830) transportiert die Schlüsselinformationen ebenfalls in den SDP-Nachrichten, kann den zu verwendenden Schlüssel aber auch asymmetrisch mit dem Public Key der Gegenseite verschlüsseln oder per Diffie-Hellman-Verfahren aushandeln.
- ZRTP[3] (IETF RFC 6189) führt das Diffie-Hellman-Verfahren zu Beginn der SRTP-Verbindung durch, also nicht über SDP-Nachrichten.

Bei Verwendung von ZRTP ergibt sich folgende Besonderheit: Da das Diffie-Hellman-Verfahren keine zuverlässige Authentifizierung der Gegenseite zulässt, wird ein kompakter Hashwert des ausgehandelten Schlüssels berechnet. Die VoIP-Telefone von Alice und Bob müssen diesen Wert auf dem Display anzeigen; dann kann z. B. Alice den bei ihr angezeigten Wert vorlesen und Bob muss überprüfen, ob bei ihm derselbe Wert angezeigt wird. Falls die Werte nicht übereinstimmen, liegt ein Man-in-the-Middle-Angriff vor und das Gespräch wird vermutlich abgehört.

3. Das Z steht für Phil Zimmermann, der auch das in Abschnitt 2.3.2 behandelte PGP erfunden hat.

4.6.3 Telefonie mit Skype

Eine weitere beliebte Internet-Telefonie-Applikation, die auch Videoübertragung und Instant Messaging unterstützt, ist Skype. Skype kam ursprünglich ohne zentrale Server aus, da alle Nachrichten über ein Peer-to-Peer-Netz verschickt werden. Skype-Rechner mit guter Bandbreite und ohne Firewall-Restriktionen konnten in diesem Netz sogenannte Supernodes werden, über die unter anderem die aktuell erreichbaren Skype-Teilnehmer gefunden werden konnten. Nach dem Kauf von Skype durch Microsoft wurde das Netzdesign aber geändert, sodass die Supernodes nur noch in den Microsoft-Rechenzentren stehen.

Das Skype-Protokoll wurde bislang nicht offengelegt. Reverse-Engineering-Versuche haben die Aussagen des Herstellers bestätigt, dass Skype die Gesprächsdaten mit AES und die Signalisierungs- bzw. Steuerungsdaten mit RC4 verschlüsselt sowie die zufällig gewählten Verbindungsschlüssel auf Basis von RSA asymmetrisch verschlüsselt austauscht. Die Enthüllungen von Edward Snowden zum Überwachungsprogramm PRISM haben 2013 aber nahegelegt, dass Textnachrichten und Audio- sowie Videodatenströme seit 2011 von Geheimdiensten abgehört werden können. Auch die inzwischen von Microsoft vorgegebenen Nutzungsbedingungen legen nahe, dass beim Einsatz von Skype keine Ende-zu-Ende-Sicherheit erreicht wird.

Übung 4.3:

Die Funktion „Skype Translator“ bietet die Möglichkeit, Skype-Telefonate automatisiert in andere Sprachen zu übersetzen. Informieren Sie sich auf der Hersteller-Webseite darüber, wie dies technisch funktioniert. Was bedeutet dies für die Vertraulichkeit der geführten Gespräche?

4.7 Online-Banking und Bezahlen im Internet

Wie Sie wahrscheinlich anhand Ihres Bekanntenkreises bestätigen können, ist Online-Banking eines der wenigen Anwendungsgebiete, bei dem sich größere Teile der Bevölkerung um das Thema Sicherheit sorgen – schließlich geht es um das hart verdiente eigene Geld. Etwas überraschend ist deshalb vielleicht, dass gerade dieses Anwendungsgebiet im Hinblick auf sichere Netzwerkkommunikation recht unspektakulär ist.

4.7.1 Online-Banking

Betrachten wir zunächst das klassische Online-Banking, für das es im Wesentlichen drei Nutzungsmöglichkeiten gibt: Fast jede Bank stellt eine Webanwendung bereit, die mit einem Webbrowser genutzt werden kann. Viele Banken bieten außerdem eigene Apps insbesondere für Smartphones und Tablets an. Ein nicht unerheblicher Teil unterstützt schließlich auch die Financial Transaction Services (FinTS), die früher als HBCI-Schnittstelle (Home Banking Computer Interface) bekannt waren und einen bankenübergreifenden Standard darstellen.

Die Nutzung der Webanwendungen erfolgt über HTTPS. Die einzige Besonderheit besteht – zumindest bei fast allen deutschen und vielen europäischen Banken – darin, dass bestimmte Aktionen durch Eingabe eines Einmalpassworts bestätigt werden müssen. Diese sogenannte Transaktionsnummer (TAN) wird über einen separaten Kommu-

nikationskanal, beispielsweise eine SMS an das Mobiltelefon des Bankkunden, übermittelt und stellt damit eine Zwei-Faktor-Authentifizierung dar. Angriffe auf diese Form des Online-Bankings basieren fast ausschließlich auf Problemen der Systemsicherheit, beispielsweise malwareinfizierte Kunden-PCs oder geklonte Handy-SIM-Karten, weshalb wir hier nicht näher darauf eingehen.

Bei Sicherheitstests der Online-Banking-Apps einiger Banken hat sich gezeigt, dass sie in Einzelfällen Fehler bei der Implementierung z.B. der Überprüfung der X.509v3-Serverzertifikate enthielten. Dadurch konnten u.a. Man-in-the-Middle-Angriffe demonstriert werden. Abgesehen von der Relevanz der Anwendungsdomäne Online-Banking stellt auch dies allerdings keine Besonderheit dar – Hunderte anderer Apps hatten und haben sehr ähnliche Probleme, die mit mehr Sorgfalt bei der Softwareentwicklung vermieden werden können.

Auch FinTS, das 2004 in Version 4.0 erschienen und zuletzt 2014 auf Version 4.1 aktualisiert wurde, setzt beim TAN-Verfahren auf HTTPS zum Nachrichtentransport. Zur Absicherung der verwendeten XML-basierten Datenstrukturen im Zusammenspiel mit Chipkarten kommt RSA für Signaturen zum Einsatz; zur Verschlüsselung wird eine symmetrische Chiffre eingesetzt: Hier hat AES in FinTS Version 4.1 den in der Finanzwelt noch weitverbreiteten TripleDES abgelöst.

4.7.2 Online-Payment

Neben Online-Schnittstellen zu klassischen Banken existieren viele andere Zahlungsmöglichkeiten; deren Besonderheiten liegen aber wiederum stärker im Bereich der eigentlichen Anwendungsprotokolle und -architekturen als in der Art der Absicherung der Netzwerkkommunikation. Wir betrachten sie deshalb nur im Überblick:

- Dienste wie PayPal, die auch über eine Banklizenz verfügen, unterstützen Geldtransaktionen über eine Webanwendung und Apps, bei denen die Empfänger nicht über herkömmliche Kontonummern und Bankleitzahlen (bzw. IBAN und BIC), sondern per E-Mail-Adresse identifiziert werden. Da auch die Empfänger PayPal -Kunden sein müssen, liegt eine recht einfache Architektur mit PayPal im Zentrum vor.
- Kreditkartentransaktionen unterliegen dem Data Security Standard der Payment Card Industry (PCI-DSS). Dieser gibt auch Online-Händlern umfassende organisatorische und technische Sicherheitsmaßnahmen vor. Ein Verstoß gegen diese Auflagen führt ggf. dazu, dass ein Händler keine Kreditkartendaten mehr verarbeiten darf, was im Online-Handel zum Ruin führen kann.
- Kryptowährungen wie Bitcoin basieren auf global verteilten Datenstrukturen, die als Blockchains oder Distributed Ledgers (verteilte Kontobücher) bezeichnet werden. In der Blockchain werden einzelne Einträge dadurch verkettet, dass sie eine kryptografische Prüfsumme des vorherigen Eintrags enthalten; ältere Einträge können also nicht nachträglich manipuliert werden. Neue Einträge werden vorgenommen, wenn sie von der Mehrheit der beteiligten verteilten Systeme akzeptiert werden. Auch wenn die genauen Abläufe je nach Kryptowährung unterschiedlich und komplexer sind, lässt sich festhalten, dass die globale Verteilung des Datenbestands auf Transparenz, nicht auf Vertraulichkeit abzielt. „Anonyme“ Zahlungen, mit denen diese Währungen oft in Verbindung gebracht werden, sind ohne Weiteres also nicht möglich, weil präzise nachvollzogen werden kann, wer wem wann welche Beträge überwiesen hat. Allerdings ist Geldwäsche vergleichsweise einfach umzusetzen, da Trus-

ted Third Parties das Geld so umverteilen können, dass Zahlungseingänge und -ausgänge nicht mehr einfach zugeordnet werden können und sich jeder Teilnehmer beliebig viele „Bankkonten“ anlegen kann, ohne namentlich erfasst zu werden. Spätestens die Auszahlung einer Kryptowährung auf ein herkömmliches „Echtgeld-Konto“ ist aber aufgrund internationaler Regulierungen wieder zuordenbar.

Auch andere Online-Bezahlmöglichkeiten wie die Paysafecard funktionieren ähnlich wie PayPal oder Prepaid-Kreditkarten: Kunden tauschen echtes Geld gegen eine Kontonummer ein und können über eine Webanwendung des Prepaid-Anbieters über den entsprechenden Betrag verfügen. Der Bezahlvorgang kann dabei anonym erfolgen; der Prepaid-Anbieter kann aber zu einem späteren Zeitpunkt ggf. im Rahmen einer Streitschlichtung oder Anfragen von Ermittlungsbehörden zur Deanonymisierung des Kunden beitragen.

Übung 4.4:

Suchen Sie im Internet nach Beispielen für Meldungen großer Internetdienste, die bekannt geben, dass sie gehackt wurden und dabei auch die personenbezogenen Daten ihrer Benutzer ausgespäht wurden. Warum wird so häufig betont, dass zwar z. B. E-Mail- und Lieferadressen in die Hände der Angreifer gefallen sind, aber keine „Bezahlinformationen“?

4.8 Protokolle zum Zugriff auf Fileserver

In lokalen Netzen erfolgt der Zugriff auf Fileserver oder das Dateisystem anderer PCs bei Microsoft Windows-Betriebssystemen überwiegend mittels SMB/CIFS; in der Linux- bzw. UNIX-Welt kommt meist NFS zum Einsatz.

Server Message Block (SMB), von dem das Common Internet File System (CIFS) eine Variante ist, stammt ursprünglich von IBM, wurde dann aber von Microsoft massiv weiterentwickelt. Seit der mit Windows 8 bzw. Windows Server 2012 eingeführten Version SMB 3.0 wird Ende-zu-Ende-Verschlüsselung mit AES-128 unterstützt.

Beispiel 4.8:

SMB 2 und SMB 3 sind in der Praxis ähnlich zum POODLE-Angriff auf TLS durch einen Man in the Middle angreifbar, der einen aus vermeintlichen Kompatibilitätsgründen erforderlichen Rückfall auf die alte Protokollversion SMB 1 anstößt.

Deswegen wird von Microsoft empfohlen, die serverseitige Unterstützung für SMB 1 explizit zu deaktivieren; zur Kompatibilität mit älteren Geräten mit Windows XP bzw. Windows Server 2003 ist SMB 1 per Voreinstellung auch in Windows Server 2016 noch aktiviert.

Das Network File System (NFS) ist in der Praxis häufig noch in der alten Version 3 im Einsatz. NFSv3 verwendet keine Verschlüsselung, schränkt den Zugriff für Clients nur aufgrund deren IP-Adresse ein und verlässt sich bei der Benutzerauthentifizierung auf die vom Client gemeldete numerische User-Id. Ein passiver Angreifer kann die Datenübertragung also im Klartext abhören, und wenn ein aktiver Angreifer die IP-Adresse eines legitimen NFSv3-Clients verwenden kann, erhält er Vollzugriff auf alle Dateien, die für den vermeintlich legitimen Client bereitgestellt werden.

NFSv4, das bereits im Jahr 2000 standardisiert wurde und immer noch weiterentwickelt wird, löst diese Probleme durch die Einführung von Verschlüsselung und „richtiger“ Benutzerauthentifizierung. Allerdings setzt NFSv4 dafür zwingend den verteilten Authentifizierungsdienst Kerberos voraus. Zwar gibt es mit MIT Kerberos und Heimdal freie Implementierungen von Kerberos. Die Komplexität ihrer Einrichtung und ihres Betriebs ist aber so hoch, dass Fileserver-Betreiber vor ihrem Einsatz zurückschrecken, wenn es in der Organisation nicht sowieso bereits eine Kerberos-Infrastruktur gibt.

Etwas ironisch mag dabei erscheinen, dass das Microsoft Active Directory eine Implementierung von Kerberos verwendet, die von zentraler Bedeutung für die Benutzerverwaltung in Microsoft-Windows-basierten Umgebungen ist. Organisationen, die eine solche Windows-Infrastruktur betreiben, setzen aber in der Regel auch auf SMB/CIFS als Fileserver-Protokoll. Da auch Linux- und UNIX-Clients und -Server auf SMB/CIFS-Fileserver zugreifen können, ist die Praxisrelevanz von NFSv4 bislang eng begrenzt. NFSv3 wird hingegen insbesondere in Rechenzentren zur direkten Vernetzung von Servern, bei der man die Verwendung von IP-Adressen durch Angreifer einfacher in den Griff bekommt, immer noch häufig eingesetzt.

SMB/CIFS und NFSv4 schützen die Daten nur beim Transport. Je nach Schutzbedarf sollte zusätzlich server- und clientseitige Festplattenverschlüsselung verwendet werden.

Zusammenfassung

In diesem Kapitel haben Sie eine Reihe von sicheren Schicht-7-Kommunikationsprotokollen für verschiedene Anwendungsschwerpunkte betrachtet; der Fokus lag dabei auf den verwendeten kryptografischen Bausteinen, Abläufen sowie Stärken und Schwächen.

SSH als Standardprotokoll für den kommandozeilenbasierten Fernzugriff hat einen ähnlichen Aufbau wie TLS und kann neben seiner Hauptanwendung noch weitere TCP/IP-Verbindungen sicher tunneln. Wie Sie gesehen haben, rüstet DNSSEC Authentizitätsüberprüfungen und Integritätssicherung für das globale Domain Name System nach, verzichtet dabei aber auf Verschlüsselung.

Mit HTTP, SMTP, XMPP haben wir drei Protokolle kennengelernt, die durch den Einsatz von TLS vergleichsweise einfach nachträglich abgesichert werden konnten. Demgegenüber zeigte uns das Instant-Messaging-Protokoll Signal, wie bestehende Konzepte mit guten Ideen und einfachen Tricks weiterentwickelt werden können, um u. a. häufige Schlüsselerneuerung auch bei zeitversetzt ablaufender Kommunikation effizient durchführen zu können. Anschließend haben wir festgestellt, dass Online-Banking aus Perspektive dersicheren Netzwerkkommunikation keine großen Besonderheiten aufweist, aber mit verschiedenen branchenspezifischen Verfahren einhergeht.

Schließlich waren die Protokolle SMB/CIFS und NFS Beispiele dafür, zu welchen Ergebnissen die Weiterentwicklung von nicht TLS-basierten Protokollen führen kann. Während SMB/CIFS eher die Abwärtskompatibilität als Achillesferse hat, ist NFS in der Version 4 zwar „sicher“ geworden, aber leider auch so komplex zu betreiben, dass es stark an Attraktivität verloren hat.

Aufgaben zur Selbstüberprüfung

4.1 Wie kann der Empfänger einer DNSSEC-gesicherten Antwort auf eine DNS-Anfrage die Authentizität der enthaltenen Informationen überprüfen?

4.2 Wie funktionieren Downgrade-Angriffe wie POODLE auf TLS oder auf SMB (bzw. CIFS) prinzipiell?

4.3 Nennen Sie zwei Gründe, warum eine per `STARTTLS` gesicherte SMTP-Verbindung zwischen zwei E-Mail-Servern sinnvoll ist, selbst wenn die Kommunikationspartner ihre E-Mails bereits auf Basis von S/MIME verschlüsseln.

4.4 Wie erreicht Off-the-Record-Messaging gezielt die Abstreitbarkeit ausgetauschter Nachrichten, ohne die Integritätsprüfung einzuschränken?

4.5 Wie läuft der Aufbau einer verschlüsselten VoIP-Telefonieverbindung mit SIP ab?

5 Zusammenspiel mit dedizierten Sicherheitskomponenten

Nach Bearbeitung dieses Kapitels können Sie dedizierte Sicherheitskomponenten aus Betreiber- und Anwendersicht im Kontext sicherer Netzwerkkommunikation beurteilen. Sie wissen, welche Komponenten unter welchen Randbedingungen zu einer Erhöhung der Sicherheit der Netzwerkkommunikation beitragen können. Sie verstehen aber im Gegenzug auch, welche Grenzen diesen Komponenten z.B. durch eine Ende-zu-Ende-Verschlüsselung gesetzt sind. Sie kennen Ansätze, um diese Ende-zu-Ende-Verschlüsselung zu umgehen, und können beurteilen, in welchen praktischen Anwendungsfällen die daraus resultierenden Nachteile akzeptabel sein können. Sie wissen, mit welchen beobachtenden und welchen aktiven Maßnahmen wichtige Eigenschaften der sicheren Netzwerkkommunikation überprüft werden können und welche Werkzeuge dafür eingesetzt werden.

5.1 Firewalls

Firewalls haben Sie in Ihrem Studium bereits als essenzielle Bausteine für die Steuerung und Kontrolle der Datenflüsse in Netzen kennengelernt. Wir rekapitulieren kurz folgende Unterscheidungen:

- ISO/OSI-Arbeitsschicht der Firewall: Klassische *Paketfilter-Firewalls* arbeiten auf den Schichten 3 und 4, wenden Filterregeln also insbesondere auf Basis von IP-Adressen und TCP- oder UDP-Portnummern an. Firewalls, die Paketinhalte auch auf der Anwendungsschicht inspizieren, werden allgemein als *Application Level Gateways* bezeichnet; konkrete Ausprägungen, z.B. für HTTP, werden oft griffiger bezeichnet, z.B. als Web Application Firewall (WAF).
- Platzierung der Firewall: Neben *Host Firewalls* (auch als *Personal Firewalls* bekannt), die auf dem zu schützenden Endgerät selbst betrieben werden, kommen Firewalls üblicherweise an den Grenzen von Netzzonen zum Einsatz; sie zielen also auf einen Perimeterschutz ab, den sie als Passierstelle für Datenpakete erreichen.

Wie wirkt sich nun der Einsatz von sicheren – insbesondere verschlüsselten – Kommunikationsprotokollen auf den Einsatz von Firewalls aus? Die Auswirkungen auf reine Paketfilter-Firewalls sind vermeintlich überschaubar, da Verbindungen anhand ihrer Charakteristika auf den Schichten 3 und 4 zugelassen oder blockiert werden.

Beispiel 5.1:

Dennoch muss beim Betrieb sorgfältig vorgegangen werden, wie folgende typische Stolperfallen zeigen:

- Einige Protokolle wie FTP handeln dynamisch Ports für Datenübertragungen aus. Wenn die Firewall diese Informationen nicht „mitlesen“ kann, kann sie auch die erforderlichen dynamischen Ports nicht freischalten.
- IPsec AH und IPsec ESP sind dedizierte Schicht-4-Protokolle wie TCP und UDP. Um sie zuzulassen, muss man folglich keine TCP-/UDP-Ports „freischalten“, sondern entsprechende Firewall-Regeln für ganze Protokolltypen anlegen.

- Einige Dienste wie Webserver sollen sowohl ungesichert als auch gesichert erreicht werden können. Es müssen also immer Regelpaare, beispielsweise für HTTP und HTTPS, angelegt werden. Bei Änderungen an der Infrastruktur muss darauf geachtet werden, dass diese nicht inkonsistent werden. Beispielsweise könnte ein Webserver außer Betrieb genommen werden und der Firewall-Administrator löscht nur die Regel für die HTTP-Verbindung, übersieht aber diejenige für HTTPS.
- Die Portnummern allein sind keine Garantie dafür, dass auch wirklich das entsprechende Protokoll zum Einsatz kommt. Ein falsch konfigurierter Webserver könnte auch Verbindungen auf dem HTTPS-Port 443 annehmen, obwohl er darüber dann nur ungesichertes HTTP spricht.

Übung 5.1:

Welche Verbindungen muss eine Paketfilter-Firewall zulassen, damit das Internet-Key-Exchange-Protokoll für IPsec genutzt werden kann?

Wenn in der clientseitigen Organisation auch verschlüsselte Verbindungen auf Schicht 7 analysiert werden sollen, muss die Verschlüsselung wie bei einem Man-in-the-Middle-Angriff aufgebrochen werden. Für HTTPS kommen dabei meist sogenannte *HTTPS-Interception*-Produkte zum Einsatz. Um Warnungen oder Fehler bei der X.509v3-Zertifikatsprüfung auf dem überwachten Endgerät zu vermeiden, müssen dort zum HTTPS Interceptor passende Zertifikate installiert und als vertrauenswürdig gekennzeichnet werden. Der HTTPS Interceptor baut dann seinerseits eine HTTPS-Verbindung zum vom Endgerät gewünschten Webserver auf. Da alle Anfragen und Antworten über ihn laufen, kann er die Inhalte der Kommunikation analysieren.

Offensichtlich geht bei diesem Ansatz der Ende-zu-Ende-Schutz der Kommunikation verloren. Diverse Produkte für HTTPS Interception hatten ihrerseits auch Implementierungsfehler, die z.B. dazu geführt haben, dass HTTPS-Verbindungen mit unsicheren, veralteten TLS-Parametern zu externen Webservern aufgebaut wurden. Das US-CERT empfiehlt deshalb, die Vor- und Nachteile der HTTPS Interception genau abzuwägen.[4] Beim praktischen Einsatz von HTTPS Interception müssen auch rechtliche Randbedingungen eingehalten werden, um nicht gegen § 202a–c StGB und die EU-Datenschutz-Grundverordnung zu verstoßen; typischerweise müssen mit Betriebs- oder Personalrat entsprechende Vereinbarungen getroffen werden, damit HTTPS Interception in einer Organisation überhaupt eingesetzt werden darf.

Beispiel 5.2:

HTTPS Interception ist – oft mit denselben Implementierungsproblemen – auch Bestandteil vieler Antivirus-Software-Suites für Endgeräte. Die betroffenen Hersteller werden dafür verstärkt kritisiert, die Sicherheit mit ihren Produkten dadurch zu schwächenstatt zu stärken.[5]

4. https://www.us-cert.gov/ncas/alerts/TA17-075A.
5. Vgl. z.B. https://heise.de/-3620159.

5.2 Service Load Balancer und Application Delivery Controller

Bei intensiv genutzten Diensten wie populären Webseiten oder E-Mail-Servern großer Unternehmen bzw. Provider ergibt sich das Problem, dass die Leistung eines einzelnen Servers nicht mehr ausreicht, um alle Client-Anfragen zeitnah zu beantworten. Die Lastmuss deshalb auf mehrere Server verteilt werden. Dazu wird jede eingehende Anfrage zunächst von einer vorgeschalteten Komponente, dem Service Load Balancer (SLB), entgegengenommen und geeignet an einen der verfügbaren Server verteilt. Neuere Gerätegenerationen werden (durch Marketingstrategien motiviert) auch als Application Delivery Controller (ADC).

Während HTTPS Interception netztopologisch in der Nähe des clientseitigen Endgeräts durchgeführt wird, brechen SLBs bzw. ADCs wie in Abb. 5.1 bei entsprechender Konfiguration die z.B. TLS-gesicherten Verbindungen unmittelbar vor denServern auf. Dies kann insbesondere folgenden Zwecken dienen:

- TLS Offloading: Die kryptografischen Operationen zum Ver- und Entschlüsseln sowie Integritätssichern und -prüfen sollen den eigentlichen Servern abgenommen werden, um die Gesamtlast noch besser zu verteilen. SLB-/ADC-Geräte können dazu dedizierte Hardware eingebaut haben.
- Affinity Scheduling: Weitere Anfragen desselben Clients sollen oft möglichst an denselben Server weitergereicht werden, der auch die bisherigen Anfragen bearbeitet hat. Die Client-IP-Adresse reicht aber oftmals nicht zur eindeutigen Identifizierung eines Clients aus; es könnte sich z.B. um einen VPN-Server oder NAT-Gateway handeln, der von Hunderten Anwendern gleichzeitig verwendet wird. Der Load Balancer muss deshalb z.B. das Session-Cookie einer Webanwendung analysieren, um die Last besser verteilen zu können.

Eine Terminierung der Ende-zu-Ende-Sicherheit beim SLB/ADC ist im Allgemeinen unkritischer zu bewerten als clientseitige HTTPS Interception. Zum einen können die richtigen X.509v3-Zertifikate der den Dienst erbringenden Organisation eingesetzt werden; zum anderen werden SLB/ADC und die Server i.d.R. räumlich eng zueinander betrieben, z. B. im selben Raum eines Rechenzentrums. Dadurch können mögliche Angriffe auf den unverschlüsselten Datenverkehr zwischen SLB/ADC und den einzelnen Servern zugunsten der Lastverteilung vernachlässigt werden. Im Gegenzug muss aber berücksichtigt werden, dass der SLB/ADC selbst ein möglicherweise lukratives Angriffsziel darstellt und aufgrund seiner Aufgabe von außen für jeden Dienstnutzer erreichbar ist. Ein kompromittierter SLB/ADC führt unweigerlich dazu, dass alle über ihn abgewickelten Verbindungen auch kompromittiert sind.

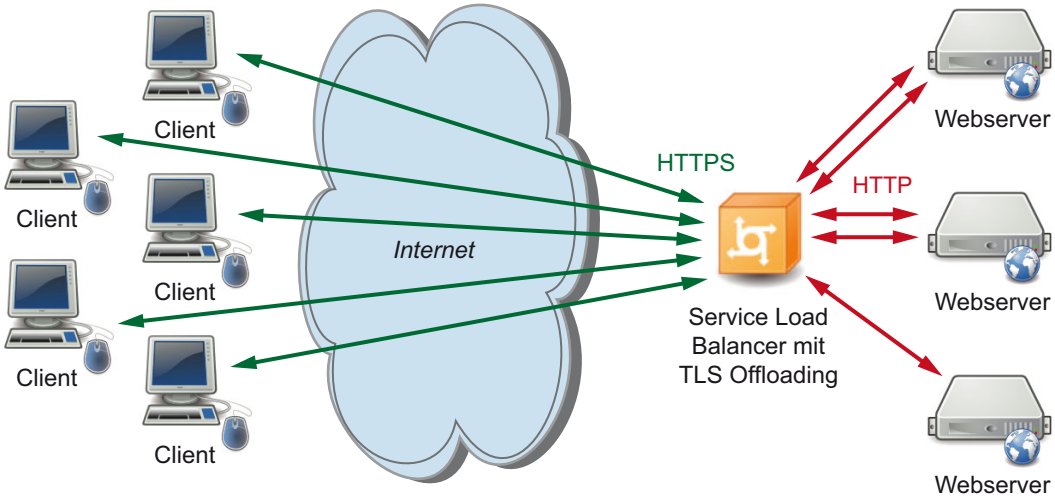

Abb. 5.1: Service Load Balancer mit TLS Offloading

5.3 Network-Intrusion-Detection- und Prevention-Systeme

Netzbasierte Intrusion-Detection-Systeme (NIDS) beobachten z. B. mithilfe eines Mirror-Ports am Switch oder Router den gesamten Datenverkehr in einem Netzsegment. Ähnlich zu Antivirus-Software versuchen sie, Angriffe entweder anhand von Signaturen, also Datenmustern bereits bekannter Angriffsvarianten, zu erkennen; oder sie arbeiten heuristikbasiert, beispielsweise indem sie den regulären, normalen Datenverkehr „lernen“ und bei statistisch signifikanten Abweichungen davon einen potenziellen Angriff melden.

NIDS gehören damit zu den Security-Monitoring-Werkzeugen, die durch verschlüsselte Kommunikation weitgehend außer Gefecht gesetzt werden:

- Durch die Verschlüsselung ist keine Deep-Packet-Inspection, also keine Auswertung der transportierten Nutzdaten mehr möglich. Es kann also kein Vergleich mit bekannten Schicht-7-Signaturen erfolgen.
- Gut verschlüsselte Datenströme sehen von außen wie Folgen von Zufallszahlen aus. Eine heuristische Auswertung kann sich also nur noch auf Metadaten wie die Anzahl, Größe und Reihenfolge einzelner Datenpakete beziehen. Abgesehen von primitiven Angriffen wie DDoS, die allein schon durch ihr hohes Paket- und Datenvolumen erkannt werden können, reichen diese Charakteristika in der Praxis bislang nicht für die zuverlässige Erkennung von Angriffen aus, obwohl es vielversprechende Forschungsansätze dazu gibt. Insbesondere kann die Anzahl von Fehlalarmen (sogenannte *false positives*) derart in die Höhe schnellen, dass das NIDS nicht mehr sinnvoll betrieben werden kann.

Übung 5.2:

Warum kann es trotzdem sinnvoll sein, mit einem NIDS auch TLS-gesicherte Verbindungen auszuwerten?

Intrusion-Prevention-Systeme (IPS) verwenden dieselben Mechanismen zur Angriffserkennung wie NIDS. Sie sind den überwachten Systemen aber vorgeschaltet und können Verbindungen bei erkannten Angriffen unterbinden, woraus sich der Begriff *Prevention* im Namen ergibt. Insbesondere hat man dabei die Möglichkeit, wie bei einem SLB bzw. ADC ein TLS Offloading durchzuführen, sodass das IPS die Daten im Klartext analysieren kann. Aus diesem Grund sind Kombinationen aus WAF, SLB und IPS eine typische Gattung kommerzieller Produkte, die z. B. als ADC vermarktet wird.

5.4 Aktive Netzüberwachung mit Portscans und Penetrationstests

Bislang haben wir Mechanismen betrachtet, die von anderen initiierte Netzverbindungen beobachten und ggf. Alarm schlagen oder diese unterbinden. Im Kontext von Abschnitt 1.4 handelt es sich also um detektierende und reagierende Sicherheitsmaßnahmen. Die aktive Netzüberwachung soll präventiv wirken, also Verwundbarkeiten finden, bevor diese von Angriffen ausgenutzt werden.

Ein einfaches, aber effektives Werkzeug zur Sondierung der Angriffsfläche sind Portscanner wie `nmap`, die Sie bereits vom Testen von Firewall-Konfigurationen kennen. Sie dienen im ersten Schritt dazu, eine vollständige Liste von IP-Adressen und TCP-/UDP-Ports zu ermitteln, auf denen im eigenen Netz Serverdienste angeboten werden. Der typische Funktionsumfang von Portscannern ermöglicht aber auch eine Abfrage der Version der jeweils eingesetzten Serversoftware, sodass z. B. Maschinen mit veralteten Installationen aufgespürt werden können. Gute Portscanner erkennen auch, dass auf einem Port TLS-gesicherte Dienste laufen, und führen ein TLS Handshake durch. Dabei kann z. B. ermittelt werden, ob eingesetzte Server-Zertifikate schonabgelaufen sind oder ob noch veraltete Verschlüsselungs- und Prüfsummenverfahren verwendet werden, die auf dem Server deaktiviert werden sollten.

Beispiel 5.3:

Internetdienste wie die Qualys SSL Labs[6] können genutzt werden, um die TLS-Konfiguration einzelner Webserver bewerten zu lassen. Für den Dauerbetrieb sind aber automatisierte Lösungen sinnvoller, die z. B. mindestens einmal pro Tag das gesamte Netz scannen. Idealerweise kommen dabei mehrere Scanner zum Einsatz, um eine Organisation z. B. von außen über das Internet und von innen z. B. sowohl aus dem Server- als auch einem Arbeitsplatzrechner-Netz zu überprüfen.

Penetrationstests (kurz: Pentests) gehen darüber hinaus, indem auch gezielt nach Konfigurationsfehlern der eingesetzten Serversoftware oder, z. B. bei selbst entwickelter Software, nach Implementierungsfehlern gesucht wird. Pentester sind also gutartige Angreifer, die Verwundbarkeiten suchen und berichten, bevor ein bösartiger Angreifer sie ausnutzt. Werkzeuge wie die Open-Source-Software `Nessus` können für einige Routineüberprüfungen eingesetzt werden, aber ein guter Pentester wird sich immer im Detail mit dem jeweiligen Server auseinandersetzen und braucht dafür Erfahrung und Zeit. Pentests werden deshalb meist als externe Aufträge vergeben und sind mit Kosten in Form mehrerer Tagessätze verbunden. Sie werden deshalb meist bei der Inbetriebnahme eines neuen Dienstes und bei größeren Updates durchgeführt.

6. https://www.ssllabs.com/.

5.5 Security-Information- and Event-Management-Systeme

Security-Information- and Event-Management(SIEM)-Systeme dienen wie in Abb. 5.2 der Zusammenführung und Auswertung von Sicherheitsmeldungen mehrerer Systeme und Komponenten. Sie fungieren im ersten Schritt also als zentrale Logging-Server z. B. für die von Firewalls, NIDS und Webservern protokollierten Sicherheitsereignisse. Die Zusammenführung dieser Meldungen ermöglicht die systemübergreifende Korrelation von Ereignissen: Wenn beispielsweise von einer externen IP-Adresse zunächst ein Portscan-Versuch unternommen wird, der sich in den Firewall-Protokollen niederschlägt, anschließend das NIDS einen potenziellen Angriffsversuch von derselben IP-Adresse registriert und zeitnah im Webserver-Log ungewöhnliche Einträge auftauchen, kann ein SIEM-System eine hoch priorisierte Warnung ausgeben, um die sich ein Sicherheitsverantwortlicher kümmern sollte. Hätten hingegen die Administratoren der drei Komponenten, falls überhaupt, nur jeweils ihre lokalen Protokolle ausgewertet, hätten sie diesen Einträgen vielleicht keine große Bedeutung beigemessen.

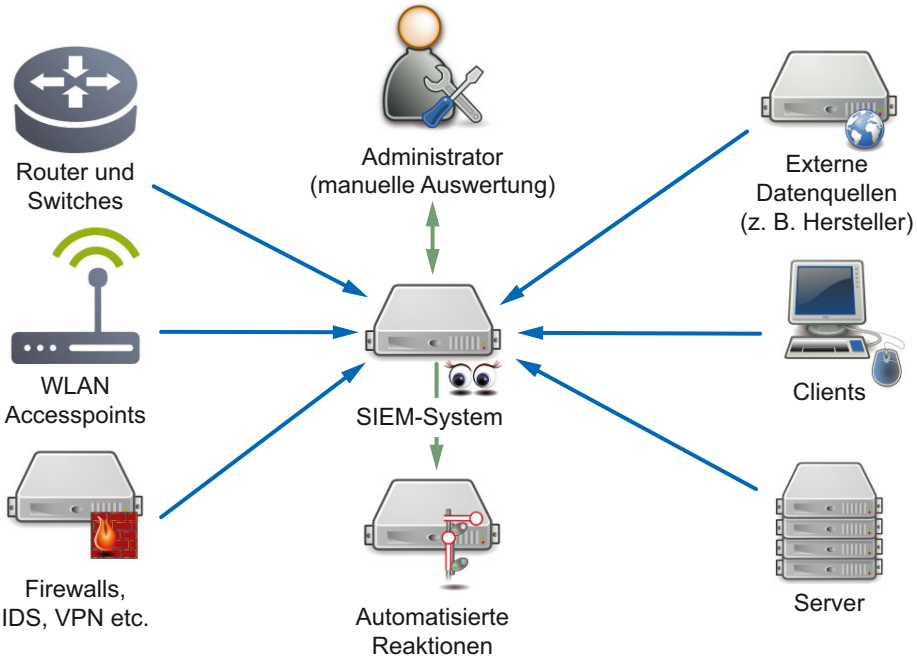

Abb. 5.2: SIEM-System zur Aggregation und Korrelation von Sicherheitsmeldungen

Insbesondere kommerzielle SIEM-Systeme werden zudem ähnlich wie Antivirus-Software regelmäßig, ggf. auch mehrmals pro Tag, vom Hersteller mit aktuellen Informationen versorgt. Dazu können beispielsweise Listen als „böse“ bekannter IP-Adressen, z. B. von zu Botnetzen gehörenden Geräten im Internet, gehören. Diese Informationen können in die Korrelation einfließen und zur besseren Klassifizierung und Priorisierung erkannter Angriffe beitragen. Geschickte Korrelation kann so insbesondere auch dazu beitragen, „blind spots“ zu kompensieren, die sich daraus ergeben, dass einzelne Komponenten nur verschlüsselten Datenverkehr zu sehen bekommen.

Beispiel 5.4:

SIEM-Systeme sollten immer mit dem organisationsinternen Asset Management verknüpft werden. Damit ist dem SIEM-System beispielsweise bekannt, dass ein bestimmter Webserver unter Linux mit der Software Apache HTTP Server betrieben wird. Nehmen wir an, dass ein NIDS dann für diesen Webserver einen Angriff meldet, der eine Verwundbarkeit für unter Windows mit dem Microsoft Internet Information Server betriebene Webserver ausnutzt. Die NIDS-Meldung kann dann automatisiert anders behandelt, z. B. ignoriert werden, als wenn ein Sicherheitsverantwortlicher nur die NIDS-Meldung vorliegen hätte und dieser von Hand nachgehen müsste.

Zusammenfassung

In diesem Kapitel haben Sie die Auswirkungen verschlüsselter Verbindungen auf klassische Sicherheitskomponenten wie Firewalls und Intrusion-Detection-Systeme kennengelernt. Sie haben gesehen, dass mit HTTPS Interception eine Überwachung von TLS-gesicherten HTTP-Verbindungen auf Clientseite möglich ist. Sie birgt zugleich aber das Risiko, das Sicherheitsniveau nach außen gehender Verbindungen zu reduzieren. Auf Serverseite haben Sie sich mit Service Load Balancern und Application Delivery Controllern auseinandergesetzt, die ein TLS Offloading und eine Lastverteilung auf mehrere Server anbieten.

Als wichtigen Betriebsaspekt haben Sie die proaktive Netzüberwachung identifiziert. Mit erweiterten Portscans können so beispielsweise veraltete TLS-Konfigurationen ermittelt werden. Schließlich haben Sie SIEM-Systeme betrachtet, die durch die systemübergreifende Korrelation von Sicherheitsmeldungen und den Einbezug weiterer externer und organisationsinterner Informationen die Qualität von Sicherheitsalarmen deutlich steigern können.

Aufgaben zur Selbstüberprüfung

5.1 Welchen Nachteil hat TLS Offloading an einen Application Delivery Controller?

5.2 Beschreiben Sie Arten von Angriffen, die ein NIDS auch bei verschlüsseltem Datenverkehr erkennen kann.

5.3 Nennen Sie zwei Möglichkeiten, um im eigenen Netz Server mit ausgelaufenen X.509v3-Zertifikaten zu ermitteln, ohne Systemzugriff auf die Maschinen zu haben.

6 Verschleierte Kommunikation

Nach Durcharbeiten dieses Kapitels kennen Sie praxisrelevante Ansätze, um nicht nur Kommunikationsinhalte kryptografisch zu schützen, sondern den gesamten Kommunikationsvorgang zu verschleiern. Sie wissen, welche dieser Ansätze typischerweise von regulären Anwendern, von Systemadministratoren bzw. von Angreifern gewählt werden. Sie verstehen die Funktionsweisen und die Grenzen jedes dieser Ansätze und wissen entsprechend auch, wie verschleierte Kommunikation prinzipiell entdeckt und unterbunden werden kann.

6.1 Primitive Ansätze über VPN und Proxies

Aus den Medien sind Ihnen sicherlich Fälle von „Abmahnwellen“ bekannt, bei denen Anwälte im Auftrag der Musik- und Filmindustrie die Nutzer von Peer-to-Peer-basierten Internet-Tauschbörsen aufspüren und mit Kosten verbundene Unterlassungserklärungen einfordern. Auch im Rahmen der behördlichen Strafverfolgung kann u.a. in Deutschland bei Internet-Providern abgefragt werden, welcher Kunde zu einem bestimmten Zeitpunkt eine bestimmte IP-Adresse zugewiesen hatte. Wenn Kommunikationsvorgänge im Netz beobachtet oder serverseitig protokolliert werden, lässt sich also anhand der Client-IP-Adressen oft ermitteln, welche Person dahintersteckt, selbst wenn die Datenübertragung verschlüsselt erfolgt. Wenn Sie beispielsweise an Dissidenten in totalitären Ländern denken, die ihre politische Meinung nicht ohne Lebensgefahr offen äußern kennen, stellen Sie aber fest, dass eine auch von staatlichen Einrichtungen nicht zurückverfolgbare Kommunikation über das Internet durchaus moralisch und ethisch erwünscht sein kann. In diesem Kapitel befassen wir uns deshalb mit der Frage, wie Kommunikation so verschleiert werden kann, dass für Außenstehende nicht mehr nachvollziehbar ist, dass und wann zwei IP-fähige Endgeräte miteinander kommuniziert haben.

Nehmen wir als Beispiel an, dass die Kommunikation mit einem Webserver von Bob verschleiert werden soll. Mit VPNs, bei denen der gesamte Client-Datenverkehr von Alice über den VPN-Server von Trent umgeleitet wird, haben wir in Abschnitt 3.4 schon einen ersten, einfachen Ansatz kennengelernt: Der eigentliche Client kommuniziert mit dem Webserver nur mittelbar über den VPN-Server. Der Webserver protokolliert als IP-Adresse der Gegenseite also diejenige des VPN-Servers, und wenn ein Angreifer die gesamte Kommunikation zu Bobs Webserver überwacht, erkennt er nicht, dass auch Alice zu den Nutzern gehört.

Für diesen Ansatz können neben VPN-Servern auch einfache Proxies verwendet werden. Wie Ihnen aus dem Studium von Firewalltechnologien bekannt ist, könnte Trent in unserem Beispiel auch einen Application Level Gateway für HTTP oder einen SOCKS-Proxy betreiben, über den Alice mit dem Webserver von Bob kommunizieren kann. Für die folgenden Überlegungen macht es keinen Unterschied, welche der drei Varianten verwendet wird.

Alice tritt bei diesem Ansatz nach außen zwar mit der IP-Adresse von Trent auf, geht aber folgende Risiken ein:

- Alice muss Trent vertrauen, dass sein VPN-Server oder Proxy ihre richtige Client-IP-Adresse nicht protokolliert und diese Information später an Dritte weitergegeben wird.

- Da der gesamte Datenverkehr über Trent läuft, erlangt Trent insbesondere bei unverschlüsselten Verbindungen Kenntnis der Inhalte. Selbst bei gesicherten Verbindungen weiß zumindest Trent sehr genau, wann und mit wem Alice kommuniziert.
- Wenn ein Angreifer die Internetanbindung von Trent überwacht, sieht er, dass Alice zu den Nutzern des VPN-Servers bzw. Proxies gehört. Anhand von Eigenschaften wie den Größen von und Zeitabständen zwischen Datenpaketen kann er die von Alice eingehenden und zum Webserver von Bob ausgehenden Datenpakete auch recht einfach zuordnen. Dies funktioniert auch dann zuverlässig, wenn zeitgleich noch andere Nutzer den Server von Trent als Zwischenstation verwenden. Der Angreifer findet also heraus, dass Alice mit Bob kommuniziert, und kennt Alices richtige IP-Adresse.

Insgesamt ist Alice also stark von der Integrität von Trent und der richtigen Konfiguration seines Servers abhängig; zudem ist sie selbst gegen einfache Angreifermodelle wie das Abhören ebendieses Servers nicht geschützt. Wenn Alices Leben davon abhängt, dass ihre Kommunikation mit Bob nicht bekannt wird, sollte sie also besser einen anderen Lösungsweg wählen.

6.2 Mixnetze, Onion Routing und Overlay-Netze

Die auf der Vermittlung von IP-Paketen basierende Kommunikation lässt sich mit formalen Informatikmethoden gut modellieren. Für das Modell kann dann mit mathematischen Beweisen nachgewiesen werden, dass bestimmte Sicherheitseigenschaften erfüllt sind bzw. bestimmte Angriffe nicht mehr praktikabel sind. Einen wichtigen Beitrag dazu hat das deutsche Forschungsprojekt AN.ON geliefert, an dem u. a. die Universität Regensburg und die TU Dresden beteiligt waren. Aus dem Projekt ist die Software JAP hervorgegangen, die seit dem Ende der staatlichen Projektförderung vom Startup JonDos weiterentwickelt wird.[7]

JAP basiert auf dem Konzept von Mixnetzen, die durch mehrere hintereinandergeschaltete Proxies realisiert werden:

- Da jeder einzelne Proxy von einem Angreifer betrieben werden könnte, werden mehrere Proxies verwendet. Das Verfahren ist nachweisbar sicher, wenn wenigstens *ein* integrer Proxy dabei ist.
- Damit nicht jeder Proxy die Nachrichteninhalte mitlesen kann, wird die Nachricht mehrfach verschlüsselt. Jeder Proxy entfernt wie bei einer Zwiebel eine Schicht dieser Mehrfachverschlüsselung, woraus sich die Bezeichnung Onion Routing ableitet, die für den unten betrachteten Ansatz Tor namensgebend war.
- Die verwendete Kette von Proxies wird als Mixkaskade bezeichnet. Damit ein- und ausgehende Datenpakete von einem externen Beobachter nicht zugeordnet werden können, muss dieselbe Mixkaskade zum einen von möglichst vielen Benutzern verwendet werden. Zum anderen sammelt jeder Proxy zunächst mehrere eingehende Datenpakete, sortiert ihre Reihenfolge um und gibt sie dann gebündelt zum nächsten Proxy weiter. Da alle Datenpakete zudem (durch Fragmentierung oder Padding) dieselbe Länge verwenden, kann ein externer Beobachter somit keine korrekte Zuordnung korrespondierender Datenpakete mehr vornehmen, selbst wenn der Datenverkehr aller Proxies überwacht wird.

7. https://www.anonym-surfen.de/.

Der Ansatz hat den Nachteil, dass durch die Sammlung von Datenpaketen bei jedem Proxy Verzögerungen entstehen. Er eignet sich deshalb gut für asynchrone Protokolle wie den E-Mail-Versand, schränkt aber interaktive Anwendungen wie das Websurfen spürbar ein. Zudem bleiben Angriffsmöglichkeiten: Wenn ein Proxy immer n Datenpakete sammelt und ein Angreifer selbst $n - 1$ Datenpakete dazu beisteuert, kann er das *eine* von Alice verbleibende Datenpaket doch wieder zuordnen, sobald es den letzten Proxy derKaskade verlässt. Eine staatliche Zensur kann auch versuchen, den Zugang zur Kaskade z. B. über Firewalls zu unterbinden. Schließlich bleiben auch Denial-of-Service-Angriffe als realistische Möglichkeit, Anwendern wie Alice die Nutzung dieses Dienstes zu verleiden.

International bekannter und noch intensiver genutzt ist Tor, das in der Schreibweise TOR früher als Akronym für *The Onion Router* verwendet wurde. Es wurde als Forschungsprojekt an der Universität Cambridge mehrere Jahre lang u. a. von der amerikanischen DARPA gefördert. Die Weiterentwicklung wird auch durch Spenden sichergestellt, zudem werden mittlerweile mehrere Tausend Tor-Server von individuellen Organisationen und Privatpersonen betrieben.

Tor unterscheidet sich von JAP u. a. dadurch, dass es keine festen Mixkaskaden verwendet, sondern wie in Abb. 6.1 aus dem Pool verfügbarer Tor-Knoten drei auswählt und diese Kette auch öfter wechselt: ein Entry Node, ein mittlerer Knoten und ein Exit Node. Zwar verwendet auch Tor die namensgebende Mehrfachverschlüsselung; allerdings werden keine immer gleich langen Nachrichten verwendet und diese nicht gesammelt und umsortiert, sondern direkt weitergegeben. Dies soll insbesondere dieInteraktivität verbessern.

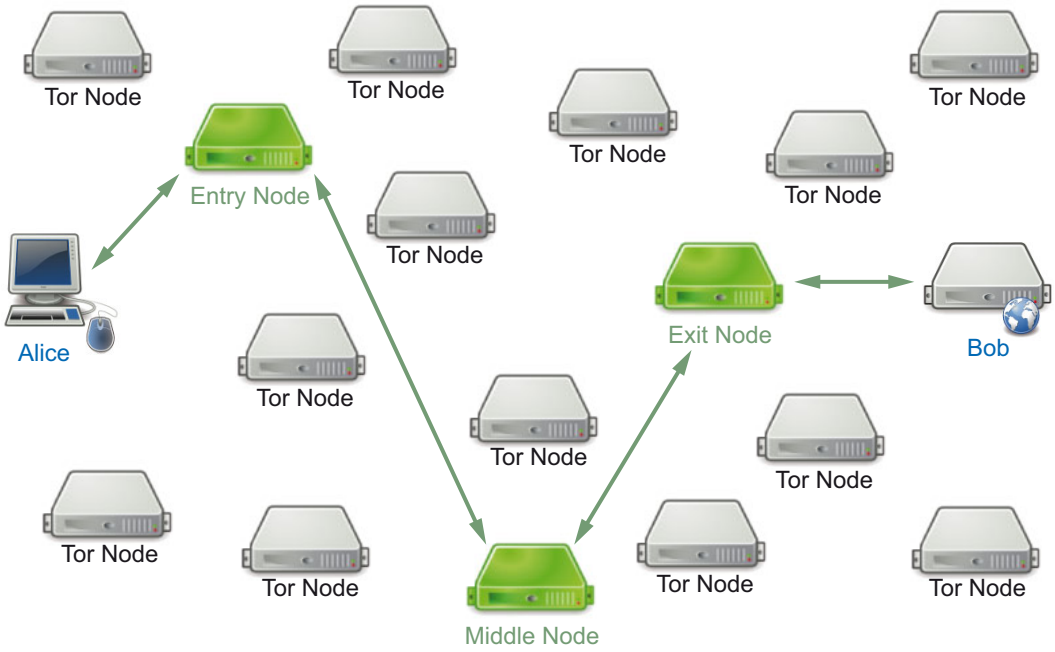

Abb. 6.1: Tor-Verbindung über drei ausgewählte Exit, Middle und Exit Nodes

Dadurch ist Tor allerdings wesentlich einfacher angreifbar als JAP. Durch eine Korrelation der beim Entry Node eingehenden mit den beim Exit Node ausgehenden Datenpaketen, die anhand von Größe, zeitlichem Abstand und Reihenfolge möglich ist, können die beiden Kommunikationspartner identifiziert werden. Nun könnte man aufgrund der

Vielzahl von Tor-Knoten meinen, dass eine derartige Korrelation nur einem Angreifer vom Kaliber NSA gelingt, der den gesamten globalen Datenverkehr überwachen kann.[8] Allerdings haben Experimente gezeigt, dass es ausreicht, in *einem* eigenen Netz mehrere Tor-Knoten zu betreiben, aus denen im Laufe der Zeit für einen Großteil der Clients sowohl Entry als auch ExitNode gewählt werden.[9] Innerhalb von drei Monaten konnten so 95 % der Benutzer deanonymisiert werden; wenn der Angreifer mindestens zwei getrennte Netze mit Tor-Knoten betreibt, reduzieren sich die drei Monate auf einen Tag. Da vermutlich ein großer Teil der Tor-Knoten von staatlichen Einrichtungen betrieben wird, sollte man den in der Realität erzielten Grad an Anonymität eher konservativ einschätzen.

Übung 6.1:

Informieren Sie sich über den Harvard University Bomb Hoax aus dem Jahr 2013. Wie konnte der Verdächtige trotz Tor-Nutzung ermittelt werden?

Tor bietet auch die Möglichkeit, Server anonym zu betreiben, indem sie nur über das Tor-Routing zugänglich gemacht werden und so die reale Server-IP-Adresse den Clients verborgen bleibt. Vermutlich haben Sie davon schon unter dem Begriff *Dark Web* gehört, da in der Presse gerne über die „dunklen Seiten" des Internets, in denen man vermeintlich einfach und jenseits staatlicher Kontrolle alles mögliche Illegale von Drogen bis zum Auftragsmord einkaufen kann. Allerdings nutzen auch bekanntelegal operierende Dienste wie Facebook diese Möglichkeit, um in vielen Ländern trotz staatlicher Zensur genutzt werden zu können.

Übung 6.2:

Der über Tor zugängliche illegale Umschlagplatz Silk Road wurde 2013 vom FBI beschlagnahmt. Wie konnte die IP-Adresse des Servers laut FBI-Angaben ermittelt werden?

Eine konsequente Weiterentwicklung dieses Ansatzes ist z. B. im Invisible Internet Project (I2P) zu sehen. I2P fungiert als Overlay-Netzwerk, es baut also ein abgeschlossenes Netz im Internet auf. In diesem können beliebige Dienste betrieben werden, die nur über I2P zu erreichen sind. Anders als Tor verwendet I2P zudem unidirektionale Verbindungen, d. h. Anfragen an und Antworten vom Server laufen über verschiedene Knotenketten.

Für den Einsatz der inzwischen vielen von Tor inspirierten Anonymisierungsansätze gilt dieselbe Regel wie für Verschlüsselungsverfahren: Nur wenn die Konzepte und Implementierungen von Fachleuten analysiert wurden, weiß man, ob die gemachten Versprechungen erfüllt werden und welche Risiken bei der Anwendung bleiben.

8. Vgl. z. B. https://heise.de/-2458992.
9. Vgl. z. B. https://heise.de/-2268444 und dort verlinkte Literatur.

6.3 Port Knocking im Serverbetrieb

Stellen Sie sich vor, dass Sie bei einem Hosting-Provider einen eigenen Server angemietet haben. Sie verwenden ihn, um darauf über das Internet Backups ihrer privaten Geräte zu speichern, damit Ihre Daten auch dann noch zur Verfügung stehen, wenn alle lokalen Speichermedien z. B. durch einen Brand oder Einbruch verloren gehen. Selbstverständlich übertragen und speichern Sie die Daten verschlüsselt.

Um den Server konfigurieren und solche Backups aufspielen zu können, muss darauf ein Serverdienst wie SSH (siehe Abschnitt 4.1) betrieben werden. Genau darin sehen Sie einen Schwachpunkt Ihres Konzepts: Wenn die verwendete SSH-Software eine Implementierungslücke hat, könnte ein Angreifer möglicherweise Ihren Server kompromittieren. Folglich möchten Sie sicherstellen, dass nur Sie allein den SSH-Dienst ansprechen können und ein Angreifer z. B. mit einem Portscan auch nicht herausfinden kann, dass auf dem Server überhaupt SSH läuft.

Der naheliegende Ansatz über eine Paketfilter-Firewall kommt leider nicht infrage: Ihr Internet-Provider weist Ihnen zu Hause jeden Tag eine neue IP-Adresse zu und Sie möchten Backups auch unterwegs auf Dienstreisen und im Urlaub erstellen können. Also müssten Sie so viele IP-Adressbereiche freischalten, dass der gewünschte Schutzgrad – ausschließlich autorisierte Personen sollen den Dienst überhaupt ansprechen können – nicht erreicht würde.

Für diesen Zweck kann ein Verfahren namens Port Knocking eingesetzt werden: Per Default sind alle Dienste auf dem Server durch eine Firewall komplett abgeschirmt und nicht ansprechbar. Erst wenn ein Client bestimmte Datenpakete an denServer schickt, also das richtige „Klopfzeichen“ verwendet, werden in der Firewall für genau seine Client-IP-Adresse bestimmte Ports und damit Dienste freigeschaltet. Das Klopfzeichen kann dabei z. B. wie in Abb. 6.2 dargestellt aus UDP- und ICMP-Paketen bestehen, die in einer bestimmten Reihenfolge an bestimmte Ports des Servers geschickt werden. Die Firewall erkennt den Eingang dieser Datenpakete und kann dann entsprechend handeln.

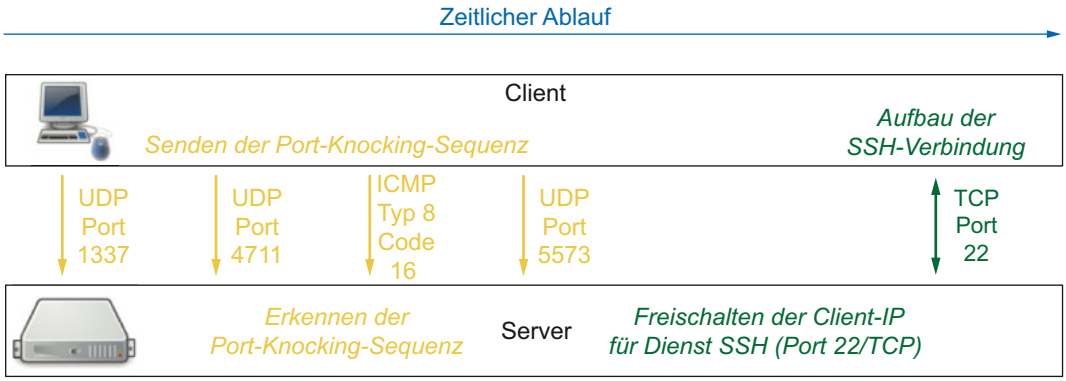

Abb. 6.2: Beispiel für den Ablauf beim Port Knocking

Im einfachsten Fall wird ein starres Klopfzeichen verwendet. Das hat allerdings den Nachteil, dass ein Angreifer, der den Netzverkehr abhört, es einfach in Erfahrung bringen und dann selbst anwenden kann. Entsprechend würde die Maßnahme in die Kategorie *security by obscurity* fallen: Die Hürde für den Angreifer wird etwas höher, objektiv wird die Sicherheit aber nicht verbessert. Deshalb werden bevorzugt dynamische Port-Knocking-Sequenzen eingesetzt, die mit den in Abschnitt 4.3.2 beschriebenen

TOTP-Einmalpasswörtern vergleichbar sind: Aus dem zeitabhängigen Einmalpasswort werden die Ports und deren Reihenfolge für das Klopfzeichen abgeleitet. Das erreichte Schutzniveau entspricht also einem Shared Secret, das nur dem Server und den autorisierten Benutzern bekannt ist. Alternativ kommen einzelne Datenpakete für Port Knocking zum Einsatz, die alle benötigten Autorisierungsinformationen in ihrem Inhalt (Header und ggf. Nutzdaten) transportieren; dieser Ansatz wird auch als Single Packet Authorization bezeichnet.

Übung 6.3:

`fwknop` ist eine populäre und noch aktiv weiterentwickelte Open-Source-Software für Port Knocking und Single Packet Authorization. Welcher Betriebsvariante wird im Quick-Start-Tutorial zu `fwknop` empfohlen und welche Parallelen zu SSH erkennen Sie bei der Ersteinrichtung des Zugangs?

6.4 Verdeckte Datenexfiltration durch Malware

Viele Organisationen speichern und verarbeiten Daten, die interessante Angriffsziele für klassische Spionage oder Industrie- bzw. Forschungsspionage darstellen. Vom einfachen Exploit für einen Serverdienst über Social Engineering wie eine gut gemachte Spear-Phishing-E-Mail bis hin zum Abfangen und Manipulieren von Hardwarelieferungen im Stil der NSA gibt es Hunderte Möglichkeiten, wie man Opfer eines erfolgreichen Spionageangriffs werden kann.

Im Regelfall gelingt es dem Angreifer nicht, mit seinem ersten Angriff direkt dasjenige System zu kompromittieren, auf dem die für ihn interessanten Daten liegen. Vielmehr muss er sich nach der Erstkompromittierung eines Systems erst im lokalen Netz „umsehen“ und in einer *Lateral-Movement*-Phase zum richtigen System vortasten. Der Angriff soll dabei möglichst lange nicht erkannt werden: Aus Sicht des Angreifers ist der Idealfall, dass alle Spuren nach erfolgreichem Ausspähen der Daten beseitigt werden können und das Opfer von dem ganzen Vorgang nichts bemerkt.

Bei derartigen Angriffen kommt fortgeschrittene Malware wie Turla, Duqu, Regin und ProjectSauron zum Einsatz. Wenn ein Angriff rechtzeitig bemerkt wird, kann die eingesetzte Schadsoftware sichergestellt und analysiert werden. Dabei zeigt sich, dass professionelle Angreifer besonders viel Wert darauf legen, bei der Exfiltration der Daten möglichst unauffällig zu bleiben. Dabei ist generell zwischen der Datenexfiltration in der Lateral-Movement-Phase und dem zeitlich nachgelagerten Exfiltrieren der ausgespähten Daten zu unterscheiden: Im ersten dieser beiden Schritte fallen nur kleinere Datenmengen an, beispielsweise Informationen zur ermittelten Netztopologie und weiteren möglicherweise kompromittierbaren Systemen. Im zweiten Schritt kann es sich um Datenvolumina in der Größe von Terabyte handeln. Ein unvorsichtiges Vorgehen könnte dazu führen, dass im Netzwerk-Monitoring auffällt, dass eine Maschine plötzlich damit beginnt, pro Tag mehrere Gigabyte an Daten zu bis dahin unbekannten Servern im Internet zu schicken.

Die Angreifer versuchen deshalb, die von ihnen angestoßene Kommunikation möglichst gut zu verschleiern. Dabei sind der Fantasie bei der Implementierung fast keine Grenzen gesetzt, weshalb wir nur einige charakteristische Beispiele betrachten:

- ProjectSauron[10] kann u. a. DNS-Pakete zur Exfiltration kleinerer Datensätze verwenden. Dabei werden DNS-Anfragen zu Einträgen in Domains gestellt, die der Angreifer unter Kontrolle hat, d. h., er betreibt einen autoritativen Nameserver, an den diese Anfragen gestellt werden. Dabei werden aber nicht herkömmliche Einträge wie `www.example.com` abgefragt. Vielmehr enthalten die DNS-Anfragen Angaben wie `R2FuekdlaGVpbSEh.example.com`, die sich nicht auf bestehende Einträge beziehen, sondern bei dieser Gelegenheit die Information `R2FuekdlaGVpbSEh` an den DNS-Server des Angreifers übermitteln. ProjectSauron überträgt Daten mit dieser Methode in 30 Byte kleinen Häppchen.
- Für größere Datensätze kann ProjectSauron von kompromittierten Clients aus E-Mails verschicken. Eine solche E-Mail enthält nur einen kurzen unverfänglichen Text; die ausgespähten Daten, um die es eigentlich geht, werden verschlüsselt als Attachment angehängt. Die Malware verwendet dabei Empfänger-E-Mail-Adressen nur einmalig und stellt sicher, dass keine Kopie der E-Mail im Ordner für gesendete E-Mails des Opfers gespeichert werden.
- Große Datenmengen werden typischerweise über HTTP oder HTTPS exfiltriert, da die entsprechenden Ports in vielen Firmen-Firewalls freigeschaltet sind und das Verfahren auch funktioniert, wenn Proxies verwendet werden müssen. Dabei erfolgt meist eine zweistufige Tarnung: Zum einen werden Daten nur von Systemen aus exfiltriert, von denen der Angreifer bereits weiß, dass sie oft und viel mit dem Internet kommunizieren. Wenn beispielsweise ein kompromittierter Arbeitsplatz-PC dafür verwendet wird, wird die Schadsoftware nur aktiv, wenn der legitime Benutzer einen Webbrowser gestartet hat und selbst gerade aktiv ist. Zum anderen erfolgt die Exfiltration zu kompromittierten Webservern, deren Nutzung bei manueller Durchsicht von Proxy-Logfiles nicht auffällig ist.

Beispiel 6.1:

Bei einem Angriff auf die schweizerische RUAG AG[11] wurden die Daten u. a. über kompromittierte Webserver einer Musikschule und eines Bestattungsunternehmens exfiltriert.

Alle Exfiltrationsvarianten könnten anhand ihrer Charakteristika erkannt werden, wenn die betroffene Organisation im Voraus wüsste, worauf sie achten soll: Viele seltsame DNS-Anfragen an eine Domain sind genauso auffällig wie erste und einmalige E-Mails an externe Empfänger mit großen Dateianhängen oder große Datenmengen, die von Clients an nicht für umfangreiche Datei-Uploads bekannte Webserver übertragen werden. Die Schwierigkeit besteht einerseits darin, dass zur Überwachung solcher verdeckten Kanäle bislang keine Werkzeuge existieren, die automatisiert – also ohne großen Betriebsaufwand – funktionieren. Andererseits hat der Angreifer meist ausreichend Zeit, um über sein Opfer ausreichend interne Informationen zu sammeln; damit kann er Exfiltrationsvarianten wählen, die mit hoher Wahrscheinlichkeit unter dem Radar des vorhandenen Netz- und Security-Monitorings bleiben.

10. Siehe Bericht des Kaspersky Lab: https://kas.pr/a9sn.
11. Siehe Bericht unter https://www.govcert.admin.ch/blog/22/technical-report-about-the-ruag-espionage-case.

Zusammenfassung

In diesem Kapitel haben Sie verschiedene Ansätze kennengelernt, um Netzwerkkommunikation zu verschleiern. Sie haben gesehen, dass VPN- oder Proxy-Lösungen dafür nur sehr eingeschränkt geeignet sind, weil die Anwender von der absoluten Integrität des Dienstes abhängig sind und eine Beobachtung des ein- und ausgehenden VPN- oder Proxy-Datenverkehrs unmittelbar zur Deanonymisierung führt.

Mit JAP, Tor und I2P haben Sie Verfahren betrachtet, die von Anwendern und z.T. von Serverbetreibern genutzt werden können, um Dienste (vermeintlich) anonym nutzen bzw. anbieten zu können. Dabei haben Sie einige Beispiele dafür untersucht, dass die in der Praxis erreichbare Sicherheit oft hinter den Erwartungen zurückbleibt: Beispielsweise kann ein Großteil der Tor-Nutzer selbst mit moderatem Ressourceneinsatz durch den Betrieb eigener Tor-Nodes über einen längeren Zeitraum deanonymisiert werden.

Als Möglichkeit zur Reduktion der Angriffsfläche auf eigene Server haben Sie sich mit Port Knocking beschäftigt. Einzelne Clients werden dabei in der Firewall erst freigeschaltet, wenn sie sich mittels eines Klopfzeichens authentifiziert haben, das z.B. aus kleinen Datenpaketen an verschiedene Ports in der richtigen Reihenfolge zusammensetzt. Schließlich haben Sie Methoden kennengelernt, mit denen u.a. bei professioneller Industriespionage ausgespähte Daten bislang meist unauffällig über das Internet zum Angreifer transportiert werden können.

Aufgaben zur Selbstüberprüfung

6.1 Bewerten Sie folgende Variante des VPN-/proxybasierten Ansatzes zur verschleierten Kommunikation: Alice leitet ihren HTTPS-Datenverkehr zunächst über einen VPN-Server und von diesem über einen Proxy-Server zum Empfänger.

6.2 Woraus ergibt sich die Sicherheit der Kommunikationsverschleierung bei Mixnetzen wie JAP?

6.3 Warum funktioniert Port Knocking, obwohl die Firewall alle eingehenden Verbindungen zurückweist?

6.4 Welche Maßnahme empfehlen Sie, um DNS-basierte Datenexfiltration zu erkennen?

Schlussbetrachtung

Im Kern dieses Studienhefts haben Sie eine ganze Reihe von Protokollen für sichere Netzwerkkommunikation betrachtet. Damit wissen Sie zum einen, wie der organisationsinterne und Internet-Datenverkehr für Anwendungen wie E-Mail, Fileserver, Instant Messaging, Voice-over-IP und das Web heute abgesichert wird. Zum anderen haben Sie die Anwendung von Chiffren und Prüfsummen z. B. in IPsec und TLS kennengelernt. Damit können Sie sich beliebige weitere sichere Kommunikationsprotokolle erschließen, diese bewerten und in eigene Anwendungen und IT-Infrastrukturen integrieren.

Sie haben aber sicherlich auch gemerkt, dass die typischerweise eingesetzten Sicherheitsmaßnahmen überwiegend auf die Sicherstellung von Vertraulichkeit und Integrität abzielen. Das ebenfalls essenzielle Ziel der Verfügbarkeit kann aus der Sicht der Kommunikationsendpunkte – also Alice und Bob bzw. beliebige Clients und Server – nur dahingehend beeinflusst werden, dass sie möglichst wenig Angriffspunkte für einfache Störangriffe bieten. Die Vielzahl unzureichend abgesicherter Endgeräte z. B. im Internet of Things und ihr Missbrauch für Denial-of-Service-Angriffe stellt somit ein praktisch ebenso relevantes Risiko dar wie fehlerhaft implementierte Verschlüsselungsverfahren oder erfolgreich geknackte kryptografische Hashfunktionen.

Sie haben ferner gesehen, dass insbesondere die Ende-zu-Ende-verschlüsselte Datenübertragung einige andere Sicherheitsmaßnahmen, insbesondere zum Security-Monitoring im Netz, mindestens erschwert und zum Teil unmöglich macht. In der Praxis müssen deshalb immer Kompromisse eingegangen werden, da die Schutzziele und Prioritäten pro konkretes Anwendungsgebiet unterschiedlich sein können und mit anderen Aspekten wie dem Betriebsaufwand in Einklang gebracht werden müssen.

Schließlich haben Sie sich mit verschleierter Kommunikation auseinandergesetzt. Sie ist dadurch motiviert, dass Metadaten – also u. a. wer wann mit wem kommuniziert hat – auch bei einer Ende-zu-Ende-Verschlüsselung erhalten bleiben. Bei der Beispielanwendung E-Mail hat daran selbst die Kombination aus IPsec, TLS und Schicht-7-Verschlüsselung mit S/MIME oder PGP nichts geändert. Hier haben Sie festgestellt, dass einerseits gegen global abhörende Angreifer selbst bekannte Verfahren wie Tor keinen ausreichenden Schutz bieten; andererseits ist es für Organisationen außerordentlich schwierig, z. B. im Rahmen von Industriespionage durchgeführte Datenexfiltration im lokalen Datenverkehr zu erkennen.

Für Ihr Berufsleben gilt es deshalb, am Ball zu bleiben: Kommunikationsprotokolle entwickeln sich inzwischen genauso schnell weiter wie die restliche IT. Mit den Grundlagen, die Sie in diesem Studienheft erarbeitet haben, und dem Überblick über den aktuellen Stand der Technik sollte Ihnen das nicht nur leichtfallen, sondern auch Spaß machen. Viel Erfolg!

A. Lösungen der Übungen im Text

1.1 Intuitiv erwartet man Vertraulichkeit, wenn personenbezogene Daten (wie Anschriften und Kontonummern) involviert oder Daten mit anderen Benutzern ausgetauscht werden sollen. Vertraulichkeit ist aber auch z.B. bei Nachrichten-Websites sinnvoll, da Angreifer sonst z.B. Interessenprofile einzelner Nutzer erstellen können.

1.2 Der Angreifer könnte neben der Nachricht auch die Prüfsumme manipulieren und auf einen korrekten Wert setzen.

1.3 Bei cloudbasierten Backup- und Archivierungslösungen sind Vertraulichkeit und Integrität noch wichtiger als durchgängige Verfügbarkeit. Bei einem Börsenticker ist vor allem die Integrität wichtig. Bei Streaming-Diensten und Onlinespielen steht oft die Verfügbarkeit im Vordergrund.

1.4 Eckpunkte des Angreifermodells sind: Angriffe über das Internet; die Angreifer waren überwiegend Amateure, die ihre Rechner mit der LOIC-Software zur Fernsteuerung durch Dritte zur Verfügung gestellt haben; die Hauptmotive waren Rache und Geltungsbedürfnis; DDoS-Angriffe sind immer aktive Angriffe.

1.5 Die drei Maßnahmen können wie folgt eingeordnet werden: 1) organisatorisch-reagierend, 2) technisch-präventiv, 3) technisch-detektierend.

2.1 Es ergeben sich die Nachrichten $A = 5^6 \mod 23 = 8$ und $B = 5^{15} \mod 23 = 19$. Der gemeinsame Schlüssel ist somit $19^6 \mod 23 = 8^{15} \mod 23 = 2$.

2.2 Eine Beschreibung des Vorfalls liefert z.B. https://cryptome.org/0005/diginotar-insec.pdf. Die Angreifer können sich eigene gültige X.509v3-Zertifikate für beliebige Domains ausstellen; der Endanwender bzw. sein Browser können sie nicht von legitimen Zertifikaten unterscheiden.

2.3 Wenn im Diffie-Hellman-Verfahren zufällige Werte für x und y verwendet werden, ist PFS gewährleistet. Beim zweiten Verfahren kann ein Angreifer den Schlüssel in Erfahrung bringen, sobald Bobs Private Key kompromittiert wurde, und damit aufgezeichnete Nachrichten entschlüsseln.

2.4 Die Benutzerfreundlichkeit u.a. für die Ersteinrichtung der Schlüssel ist in vielen E-Mail-Programmen immer noch unzureichend für einen Einsatz in der Breite. Ungesicherte E-Mails gelten als Normalfall, d.h., die E-Mail-Programme motivieren dieAnwender auch kaum dazu, entsprechende Schlüsselpaare anzulegen.

3.1 Im Windows-Dialog zu den „Eigenschaften“ der Netzwerkkarte kann im Reiter „Erweitert“ die Eigenschaft *Network Address* auf einen beliebigen gültigen Wert geändert werden. Diverse *MAC Address Changer* Tools leisten dasselbe.

3.2 Zum Einsatz kommt meist ein Server, der das Protokoll *RADIUS* zur Authentifizierung anbietet. Entweder werden die Passwörter dort lokal eingetragen, oder der RADIUS-Server greift seinerseits auf einen zentralen Benutzerdatenbestand zu, z. B.einen LDAP-Server oder ein Microsoft Active Directory.

3.3 Da IPsec fest zu IPv6 gehört, müssten alle IPv6-fähigen Geräte auch IPsec beherrschen. Dies ist bei allen gängigen Desktop- und Mobilgeräte-Betriebssystemen seit mehreren Jahren der Fall. Viele kleine Embedded-Systeme haben aber nur rudimentäre IP-Implementierungen und unterstützen entsprechend kein IPsec.

3.4 ESP im Tunnel-Modus verschlüsselt das ursprüngliche IP-Paket bei der Übertragung zwischen den beiden Security Gateways. Ein Angreifer kann also nur sehen, dass Daten zwischen den Security Gateways ausgetauscht werden. Er sieht aber nicht, welche Endgeräte miteinander kommunizieren. Dies schützt die Endgeräte vor Verkehrsflussanalysen.

3.5 Das IETF-Standarddokument RFC 7296 verweist darauf, dass sich aktuelle Angaben unter https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml finden. Dort sind als Verschlüsselungsalgorithmen u. a. DES, TripleDES, AES und CAMELLIA genannt; für Prüfsummen sind u. a. MD5, SHA-1 und SHA-2 vorgesehen. Aus Gründen der Abwärtskompatibilität mit älteren IPsec-Appliances halten sich hier also auch veraltete Hashfunktionen noch etwas länger. Dabei geht man pragmatisch davonaus, dass es einem Angreifer immer noch nicht gelingen kann, bei der Paketmanipulation in Echtzeit eine Kollision zu erzeugen.

3.6 Durch Verwendung eines VPN-Servers im Ausland konnte man dem Dienst gegenüber mit dessen IP-Adresse als Client auftreten. Viele Anbieter führen inzwischen Blacklists bekannter VPN-Server oder probieren durch Verbindungsversuche aus, ob auf der vermeintlichen Client-IP-Adresse ein VPN-Dienst läuft.

4.1 Wenn ein Angreifer den Client eines Benutzers kompromittiert, hat er Zugriff auf dessen Dateien und damit den SSH Private Key. Wird dieser nicht durch eine Passphrase geschützt, erhält der Angreifer auch unmittelbar Zugriff auf den SSH-Server.

4.2 Bei unverschlüsseltem SIP kann ein Angreifer den Schlüssel mithören. Zudem erlangen mindestens die SIP-Server Kenntnis des Schlüssels.

4.3 Die Spracherkennung erfordert die Analyse des Gesprächs durch Server des Anbieters. Es liegt also keine Ende-zu-Ende-Vertraulichkeit mehr vor.

4.4 Durch die Vorgaben der Payment Card Industry müssen Bezahlinformationen besonders geschützt werden. Zur Vermeidung von Strafzahlungen bis hin zum Entzug der Lizenz zum Verarbeiten von Kreditkartendaten sind die Dienstleister motiviert, diese Vorgaben tatsächlich umzusetzen oder diesbezügliche Defizite so lange abzustreiten, bis sie nachgewiesen werden.

5.1 IKE läuft über UDP/IP auf Port 500. Eine Firewall muss also aus- und eingehende Verbindungen auf diesem Port von und zu allen IPsec-nutzenden Endgeräten wie IPsec Security Gateways zulassen.

5.2 Ein NIDS kann immer noch die TLS Handshakes beobachten und z. B. bei abgelaufenen Zertifikaten, nicht vertrauenswürdigen Certificate Authorities oder unsicheren kryptografischen Verfahren Alarm schlagen.

6.1 Der verdächtige Student hat die E-Mail zwar über Tor verschickt, war zu diesem Zeitpunkt aber wohl der einzige oder einer von nur wenigen Tor-Nutzern im überwachten Campusnetz der Universität. Damit konfrontiert hat er seine Tat gestanden.

6.2 Auf der Loginseite von Silk Road wurde das CAPTCHA-Verfahren verwendet, um automatisierte Logins zu unterbinden. Nach Darstellung des FBI war die verwendete CAPTCHA-Software so konfiguriert, dass sie die richtige IP-Adresse des Servers (also nicht nur seine Tor-Adresse) verraten hat. Durch dieses *Information Leak* konnte der Standort des Servers bestimmt und dieser beschlagnahmt werden.

6.3 `fwknop` empfiehlt Single Packet Authorization. Ähnlich wie bei SSH-Schlüsselpaaren wird clientseitig zunächst ein Schlüssel erzeugt, der anschließend serverseitig hinterlegt werden muss. Dabei können sowohl symmetrische (AES) als auch asymmetrische Schlüssel (auf Basis von GnuPG) eingesetzt werden.

B. Lösungen der Aufgaben zur Selbstüberprüfung

1.1 Der Angriff wird von jemandem durchgeführt, der bereits Zugriff auf eine der beteiligten Komponenten hat. Der Angriff ist z.B. auf einem Mailserver als Administrator technisch einfach durchführbar; beim Ausleiten von Datenverkehr an Lichtwellenleitern oder Netzkomponenten ist der Aufwand höher und setzt z.B. geeignete Filtersoftware voraus. Die Motivation kann von persönlicher Neugierde bis zu Interessen des Staatsschutzes reichen. Es handelt sich um einen rein passiven Angriff.

1.2 Der Schutzbedarf hängt immer von konkreten Einzelfall ab. Mögliche Argumentationen sind:

- Aufgrund z.B. von Gehaltsinformationen sind Vertraulichkeit und Integrität des Personalverwaltungssystems besonders wichtig. Zu bestimmten Zeitpunkten wie der monatlichen Gehaltszahlung ist die Verfügbarkeit ebenfalls kritisch, ansonsten sind kürzere Ausfallzeiten vertretbar.
- Da die Nutzdaten öffentlich sein sollen, ist die Vertraulichkeitsanforderung nur niedrig ausgeprägt. Angesichts der Zielgruppe ist auch der Bedarf an Integrität und Verfügbarkeit niedrig.
- Da es sich um öffentliche Daten handelt, sind die Anforderungen an die Vertraulichkeit niedrig. Die Integrität der gelieferten Informationen ist aber als wichtig anzusehen. Die Verfügbarkeitsanforderungen können schwanken, z.B. je nachdem, ob der Dienst von Privatanwendern nur gelegentlich oder von Systemen zum Handeln automatisiert eingesetzt wird.

1.3 Verschlüsselung ist eine technische Maßnahme, die dem Mithören durch Dritte vorbeugen soll, also präventiv wirkt. Da sie nicht vor blinder Manipulation schützt, muss sie i.d.R. mit integritätssichernden Maßnahmen kombiniert werden, um aktive Angriffe detektieren zu können.

2.1 Weil der zur Entschlüsselung der E-Mail benötigte Schlüssel mit dem Public Key des Empfängers geschützt wird, kann ein Angreifer früher abgehörte E-Mails entschlüsseln, sobald er den Private Key des Empfängers kompromittiert hat. PFS wird also nichterreicht.

2.2 Die Global PKI besteht aus weltweit Hunderten von Certificate Authorities, die beliebige Zertifikate ausstellen können. Einige davon stehen unter staatlicher Kontrolle und es sind diverse Fälle von gehackten CAs bekannt. Da jede CA der Global PKI für beliebige Domains Zertifikate erstellen kann, besteht in Kombination mit Man-in-the-Middle-Angriffen die Gefahr, dass ein Angreifer die vermeintlich vertrauliche Kommunikation abhören kann.

3.1 Technisch ist die Kombination problemlos möglich, da die Verfahren auf unterschiedlichen Schichten arbeiten. Sinnvoll ist sie, wenn IPsec z.B. im Tunnel Mode mit einem Security Gateway eingesetzt wird und somit keine Ende-zu-Ende-Sicherheit erzielt. Im Transport Mode ist sie redundant.

3.2 Da sich die Quell-IP-Adresse am NAT-Gateway, also beim Übergang zu im Internet gerouteten IP-Adressen ändert, stimmt die AH-Prüfsumme nicht mehr.

3.3 Über X.509v3-Zertifikate, die eine von der Gegenseite als vertrauenswürdig akzeptierte Certificate Authority voraussetzen, oder ein Shared Secret, z. B. ein Passwort, da vorab beide Seiten von Hand eingetragen haben.

3.4 Wenn die Variante mit Diffie-Hellman-Schlüsselaustausch verwendet wird und dabei jedes Mal gute, also nicht vorhersagbare Zufallszahlen zum Einsatz kommen.

3.5 Entweder werden vom VPN-Client nur die Daten, die für Maschinen im Netz des VPN-Servers bestimmt sind, übertragen oder der gesamte Datenverkehr des VPN-Clients wird über den VPN-Server geleitet.

4.1 Indem er die kryptografische Signatur des Eintrags mit dem Public Key der Zone prüft. Den Public Key erhält er, von der übergeordneten Zone signiert, ebenfalls über DNS.

4.2 Ein Man-in-the-Middle greift aktiv in die Kommunikation ein und gibt dabei vor, dass mindestens eine der beiden Seiten nur die alte Protokollversion unterstützt.

4.3 Zum einen kann durch Zertifikatsprüfung sichergestellt werden, dass die E-Mail beim richtigen Mailserver ankommt und dort zugestellt werden kann. Zum anderen werden Verkehrsflussanalysen (wer schickt wem wann eine E-Mail) durch Abhören auf dem Übertragungsweg verhindert.

4.4 Die zur Integritätssicherung verwendeten Schlüssel werden nach der Integritätsprüfung veröffentlicht. Damit hätte jeder diese Nachricht erstellen und so signieren können.

4.5 Die Kommunikationspartner einigen sich entweder über SIP-Nachrichten mit SDP-Anteil auf die zu verwendenden Schlüssel oder auf die Verwendung eines Protokolls wie ZRTP, das den Schlüsselaustausch beim Aufbau der Verbindung für die Gesprächsübertragung aushandelt.

5.1 Die Ende-zu-Ende-Sicherheit geht verloren, sodass die Sicherheit der Verbindung zwischen ADC und den Servern anderweitig sichergestellt werden muss. Zudem wirkt sich ein kompromittierter ADC mit TLS Offloading fatal auf sämtliche darüber abgewickelten Verbindungen aus.

5.2 Das NIDS kann weiterhin Angriffe erkennen, die sich anhand der Verbindungsmetadaten erkennen lassen; beispielsweise Denial-of-Service-Angriffe, bei denen ungewöhnlich viele oder große Datenpakete ans Zielsystem geschickt werden. Außerdem können z. B. die TLS Handshakes auf Indizien geprüft werden, dass Man-in-the-Middle-Angriffe wie POODLE oder FREAK durchgeführt werden.

5.3 Zum einen können NIDS eingesetzt werden, die wie bei der vorherigen Aufgabe die TLS Handshakes und die darin enthaltenen Serverzertifikate prüfen. Zum anderen können Portscans um die Durchführung von TLS Handshakes erweitert und somit Serverzertifikate explizit abgerufen werden, um sie überprüfen zu können.

6.1 Durch die Verwendung von HTTPS wird zumindest Ende-zu-Ende-Vertraulichkeit erzielt, d.h., VPN- und Proxy-Serverbetreiber können die Inhalte nicht mitlesen. Es bleibt aber das Problem, dass ein Angreifer, der den Datenverkehr sowohl beim VPN- als auch beim Proxy-Server beobachten kann, die Zuordnung der von Alice stammenden und vom Proxy zum Webserver geschickten Datenpakete vornehmen kann.

6.2 Die Sicherheit ergibt sich daraus, dass mindestens ein integrer Mix-Knoten eingehende Datenpakete von vielen Anwendern sammelt, die verschlüsselt sowie gleich groß sind und deren Reihenfolge verändert wird, bevor sie weitergeleitet werden. Ein von außen beobachteter Angreifer kann die ausgehenden Pakete nicht mehr korrekt den eingegangenen Paketen zuordnen.

6.3 Für das Verfahren reicht es aus, dass die Firewall die Port-Knocking-Pakete registriert, auch wenn damit kein Dienst (auf einem offenen Port) direkt angesprochen werden kann.

6.4 DNS-Anfragen könnten beispielsweise am lokalen DNS-Resolver protokolliert werden. Wenn zu einer Domain sehr viele verschiedene Anfragen gestellt werden, die von bekannten üblichen Namensschemata abweichen, kann dies als Indiz für eine Datenexfiltration gewertet werden.

C. Abkürzungsverzeichnis

ADC	Application Delivery Controller
AES	Advanced Encryption Standard
AH	IPsec Authentication Header
CA	Certificate Authority
CIFS	Common Internet File System
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
DANE	DNS-based Authentication of Named Entities
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DH	Diffie-Hellman
DNS	Domain Name System
DNSSEC	DNS Security Extensions
DoS	Denial of Service
EAP	Extensible Authentication Protocol
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ESP	IPsec Encapsulating Security Payload
HTTP(S)	HyperText Transfer Protocol (Secure)
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange Protocol
IP(v4)	Internet Protocol (Version 4)
ISMS	Informationssicherheitsmanagementsystem
ISO	International Standards Organization
JAP	Java Anon Proxy
LAN	Local Area Network

MAC	Message Authentication Code
MD	Message Digest (Prüfsumme)
MIME	Multipurpose Internet Mail Extension
MX	Mail Exchange
NAC	Network Access Control
NFS	Network File System
NIDS	Network Intrusion Detection System
NSA	National Security Agency
NIST	National Institute for Standards and Technology
OSI	Open Systems Interconnection
PFS	Perfect Forward Secrecy
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
POP3(S)	Post Office Protocol Version 3 (Secure)
PSK	Pre-Shared Key
RFC	Request for Comment der IETF
RSA	Chiffre von Rivest, Shamir und Adleman
RTP	Real-Time Protocol
SA	Security Association (im Kontext IPsec)
SDP	Security Description Protocol
SFTP	Secure File Transfer Protocol
SHA	Secure Hash Algorithm
SIEM	Security Information & Event Management
SIP	Session Initiation Protocol
SLB	Service Load Balancer
SMB	Server Message Block
SMTP(S)	Simple Mail Transfer Protocol (Secure)
SPI	Security Parameters Index
SSH	Secure Shell
SSL	Secure Socket Layer

TCP	Transmission Control Protocol
TOS	Type of Service
TLS	Transport Layer Security
TTL	Time To Live
UDP	User Datagram Protocol
UML	Unified Modeling Language
VoIP	Voice over IP
VPN	Virtual Private Network
WAF	Web Application Firewall
WLAN	Wireless LAN
XMPP	Extensible Messaging and Presence Protocol
XOR	Bitweise Exklusiv-Oder-Verknüpfung

D. Literaturverzeichnis

Anderson, R. (2008).
Security Engineering: A guide to building dependable distributed systems. 2. Auflage. Wiley.

Bauer, F. L. (2000).
Entzifferte Geheimnisse: Methoden und Maximen der Kryptologie. 3. Auflage. Springer.

Brenner, M.; Felde, N. gentschen; Hommel, W.; Metzger, S.; Reiser, H.; Schaaf, T. (2017).
Praxisbuch ISO/IEC 27001: Management der Informationssicherheit und Vorbereitung auf die Zertifizierung. 2. Auflage. Hanser.

Eckert, C. (2014).
IT-Sicherheit: Konzepte – Verfahren – Protokolle. 9. Auflage. Oldenbourg: De Gruyter.

Eren, E.; Detken, K.-O. (2007).
VoIP Security: Konzepte und Lösungen für sichere VoIP-Kommunikation. München: Hanser.

Eye, F. v.; Hommel, W.; Metzger, S. (2015).
Aufbau und Betrieb von organisationsweitem Security-Monitoring. IT-Administrator Magazin 03/2015. München: Heinemann.

Feuchtinger, D.; Hommel, W.; Reiser, H.; Schmidt, B.; Storz, M. (2015).
DNSSEC – Konzepte und Betriebsaspekte des Domain Name Systems der Zukunft. In: PIK – Praxis der Informationsverarbeitung und Kommunikation 38, Nr. 1–2.

Kappes, M. (2013).
Netzwerk-und Datensicherheit. 2. Auflage. Springer Vieweg.

Kraft, P.; Weyert, A. (2015).
Network Hacking. Franzis Verlag.

Paar, C.; Pelzl, J. (2016).
Kryptografie verständlich – Ein Lehrbuch für Studierende und Anwender. Springer Vieweg.

Reiser, H.; Hommel, W. (o.J.).
Skript zur Vorlesung IT-Sicherheit – Sicherheit vernetzter Systeme. WS 2015/16. http://www.nm.ifi.lmu.de/teaching/ Vorlesungen/2015ws/itsec/. 2016.

Schneier, B. (1996).
Angewandte Kryptografie. Protokolle, Algorithmen und Sourcecode in C. 5. Auflage. Addison-Wesley.

Schäfer, G. Roßberg, M. (2014).
Netzsicherheit. 2. Auflage. dpunkt.verlag.

Tanenbaum, A.; Wetherall, D. (o.J.).
Computernetzwerke. 5. Auflage. Pearson Studium.

E. Abbildungsverzeichnis

F. Sachwortverzeichnis

X

Z

G. Einsendeaufgabe Typ A

Sichere Netzwerkkommunikation

Einsendeaufgabencode:
SRN05-XX1-A02

Name:	Vorname:	Tutor/-in:
Postleitzahl und Ort:	Straße:	Datum:
Matrikel-Nr.:	Studiengangs-Nr.:	Note:
Heftkürzel: **SRN05**	Druck-Code: **0118A02**	Unterschrift:

Bitte reichen Sie Ihre Lösungen über StudyOnline ein. Falls Sie uns diese per Post senden wollen, dann fügen Sie bitte die Aufgabenstellung und den Einsendeaufgabencode hinzu.

1. Beschreiben Sie für die Anwendung „Online-Banking über HTTPS“, wie die fünf von der OSI Security Architecture geforderten Sicherheitsdienste durch HTTPS und auf Anwendungsebene umgesetzt werden können.

   20 Pkt.

2. Für die Datenübertragung von TCP/IP-Paketen zwischen Alice und Bob soll IPsec eingesetzt werden. Alice und Bob verwenden IPsec-fähige Endgeräte, die aber private IPv4-Adressen verwenden. An den Standorten von Alice und Bob stehen auch IPsec Security Gateways zur Verfügung.

   a) Welche Kombination aus IPsec-Betriebsmodi und -Protokollen kann verwendet werden, um Ende-zu-Ende-Authentizität zu erreichen und die Daten dabei verschlüsselt zwischen den Security Gateways auszutauschen?

   b) Beschreiben Sie den schematischen Paketaufbau eines so zwischen den Security Gateways übertragenen IP-Pakets.

   c) Welche IPsec Security Associations werden für den Gesamtablauf benötigt?

   40 Pkt.

3. Alice und Bob möchten sich gegenseitig möglichst sicher E-Mails schicken können. Sie haben eigene DNS-Domains, betreiben ihre Mailserver selbst und möchten IPsec, TLS und S/MIME möglichst flächendeckend und kombiniert einsetzen.

   a) Welche Kommunikationsprotokolle kommen insgesamt zum Einsatz, wenn Alice eine so verschlüsselte und gesicherte E-Mail an Bob schickt und dieser sie abruft?

   b) An welchen Stellen kommen typischerweise X.509v3-Zertifikate zum Einsatz? Welche davon könnten durch Shared Secrets bzw. Pre-Shared Keys ersetzt werden?

   c) Beschreiben Sie je einen sicherheitsspezifischen Vor- und Nachteil, wenn Alice und Bob stattdessen InstantMessaging mit dem Protokoll Signal verwenden.

   40 Pkt.