Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

Walter Kriha · Roland Schmitz

Internet-Security aus Software-Sicht

Grundlagen der Software-Erstellung für sicherheitskritische Bereiche

Springer

Walter Kriha
Roland Schmitz
Hochschule der Medien
Nobelstr. 10
70569 Stuttgart
schmitz@hdm-stuttgart.de
www.medieninformatik.hdm-stuttgart.de

ISBN 978-3-540-22223-1 e-ISBN 978-3-54068906-5

DOI 10.1007/978-3-54068906-5

ISSN 1439-5428

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

Einbandgestaltung: KünkelLopka Werbeagentur, Heidelberg

Gedruckt auf säurefreiem Papier

9 8 7 6 5 4 3 2 1

springer.com

Danksagung

Dieses Buch wäre nicht möglich gewesen ohne die Hilfe und die Unterstützung vieler Personen. An erster Stelle sind hier unsere Familien zu nennen, die ein großes Maß an Verständnis während der langen Entstehungszeit dieses Buches und des Folgebands aufbringen mussten: Elfi, Beate und Kerstin Kriha, Jutta und Paul Schmitz.

Unsere Freunde und Kollegen aus dem Bereich der IT-Security wurden nicht müde, mit uns wichtige Kernideen der Security zu diskutieren. Sie haben uns die vielleicht wichtigste Idee der Security vermittelt, nämlich bei Unklarheiten ohne Scheu so lange nachzufragen, bis man das Problem oder die Lösung verstanden hat.

Unser Verlag hat einen erstaunlich langen Atem bewiesen, obwohl wir Deadline nach Deadline platzen ließen und aus einem geplanten Buch letztlich zwei wurden. Für dieses Verständnis und die gewährte Unterstützung ein herzliches Dankeschön an unsere Lektoren beim Springer-Verlag.

Last but not least danken wir unseren Studenten und Kollegen im Studiengang Medieninformatik an der Hochschule der Medien Stuttgart für die vielen anregenden Diskussionen zum Thema Security. Wir haben enorm viel daraus gelernt.

Stuttgart,
im Oktober 2007

Walter Kriha
und Roland Schmitz

Inhaltsverzeichnis

Kapitel 1
Einführung und Motivation

Zwei Großkonzerne erstellten im Jahre 2004 zusammen ein Kundenportal, mit dem Kunden ihre eigenen Stammdaten und die bestellten Services bequem verwalten konnten. Allerdings gestattete das Portal den Kunden auch die Verwaltung der Daten *anderer* Kunden – durch einfachste Manipulationen an den URLs der herunter geladenen Seiten. Gewieftere Angreifer konnten sich mit wenig Aufwand Administratorrechte verschaffen, und zwar im gesamten internen Netzwerk des Konzerns (Details findet man bei [CCC]). Am Ende musste das Portal für gut zwei Monate abgeschaltet und neu implementiert werden. Die Rede ist vom Online Business Solution Operation Center (OBSOC), dem Portal der Deutschen Telekom zur Verwaltung von Kundendaten. Wie kann es sein, dass wichtige, von großen Konzernen entwickelte Systeme so unsicher sind?

Im Juli 2005 werden von einem amerikanischen Verarbeiter von Visa- und MasterCard-Transaktionen 40 Millionen Kundendaten entwendet – teilweise sogar mit PIN-Nummer. Wie kann es sein, dass auf den Desktops der Mitarbeiter dieser Firma hochbrisante Kundendaten herumliegen und gestohlen werden können? Attacken durch Buffer Overflows, Viren und Trojanische Pferde bedrohen nach wie vor die meisten Maschinen und Benutzer weltweit und verursachen enorme Schäden. Wie kann es sein, dass heutige Betriebssysteme und ihre Besitzer so schnell an Eindringlinge verraten, statt ihnen Hilfestellungen beim Erkennen von Malware zu geben – oder noch besser: das Gefahrenpotential der Malware zu verringern? Warum führt die kleinste Lücke in Applikationen und Servern schon dazu, dass alles verloren ist?

In der Vergangenheit haben sich die Bemühungen der IT-Sicherheitsexperten im Zuge der Erfolgsstory des Internets vor allem auf die Netzwerksicherheit konzentriert. Zwar sind auch auf diesem Gebiet noch längst nicht alle Probleme gelöst – wir verfügen jedoch zumindest über eine Reihe von leistungsfähigen kryptografischen Modulen und Sicherheitsprotokollen, die, wenn richtig implementiert, eine zuverlässige Absicherung der Kommunikation zwischen zwei Rechnern bieten.

Die oben genannten Beispiele zeigen jedoch deutlich, dass nicht die Kommunikation zwischen Client und Server, sondern die *Software* auf Client- und Serverseite das Hauptproblem ist. Unzureichende Validierung der Eingaben der Clients führt dazu, dass böswillige Nutzer sich Rechte auf dem Server verschaffen können,

die ihnen nicht zustehen. Auf der anderen Seite sind die PCs legitimer Nutzer durch Malware gefährdet, die ihre persönlichen Daten ausspäht. Aber wie ist das möglich?

Wieso wird so viel unsichere Software entwickelt? Diese Frage wurde in der Vergangenheit schon in ganz verschiedener Weise beantwortet. Von der mangelnden Haftung der Softwarehersteller ist die Rede sowie vom Zeitdruck und unmöglichen Deadlines bei der Entwicklung. All dies spielt gewiss eine Rolle, aber es ist aus unserer Sicht nicht entscheidend. Im vorliegenden Buch vertreten wir die These, dass die folgenden Gründe für die Existenz der meisten unsicheren Software verantwortlich sind:

- An erster Stelle steht die mangelnde Wahrnehmung von Sicherheitsproblemen auf Seiten der Softwareentwickler. Der Mangel kann dabei auf ganz verschiedenen Ebenen existieren und reicht von Missverständnissen bezüglich der beteiligten Personen und Geschäftsvorgänge bis hin zu einem grundsätzlichen Missverständnis der Kommunikation in verteilten Systemen.
- Hinzu kommt mangelndes Wissen der Entwickler über Sicherheitstechniken und Sicherheitsarchitekturen, vor allem über das Zusammenspiel von Software, Infrastruktur und Benutzern. Verstärkt wird dieses Problem durch ein begriffliches Chaos auf Grund konkurrierender Standards und Techniken gerade im Sicherheitsbereich.
- An dritter Stelle steht ein Mangel an konkreten Mustern (Patterns) zur Lösung von Sicherheitsproblemen.
- An vierter Stelle – jedoch keineswegs weniger wichtig als die anderen – stehen die sicherheitsrelevanten Eigenschaften aktueller Betriebsysteme und Programmiersprachen, die meist gegen bekannte Prinzipien der sicheren Software-Entwicklung verstoßen (Fail Gracefully, Principle-of-least-Authority (POLA)).

Komplexe Frameworks, wie sie im Falle von OBSOC zum Einsatz kamen, beinhalten umfangreiche Hilfestellungen für die Erstellung von Webapplikationen. Sicherer Code wird hier zu einem guten Teil durch das Framework möglich gemacht. Nur müssen die Entwickler die Problemfelder zumindest erkennen und außerdem die Mechanismen des Frameworks verstehen und einsetzen können.

Der gegenwärtige Trend zu immer mehr Software im „embedded control"-Bereich mit Anwendungen zum Beispiel in der Haustechnik oder der Automobilindustrie wird das Sicherheitsproblem noch verschärfen. Was passiert, wenn die momentane Art der Softwareentwicklung auf diese Systeme übertragen wird? Welche Qualitäts- und Sicherheitsprobleme gefährden die Systeme und ihre Nutzer dann? In diesem Bereich tritt die Zuverlässigkeit als Teilaspekt sicherer Software stärker in den Vordergrund als bei Internet-Applikationen. Wo sind hier die Systeme, die Software verschiedenster Hersteller nebeneinander laufen lassen können, ohne dass sich die Komponenten durch die gemeinsame Benutzung von CPU, Speicher oder Geräten gegenseitig beeinflussen könnten? Virtuelles Memory vermag zwar die gröbsten Übergriffe zu verhindern, spielt jedoch innerhalb moderner Application Server und ihrer Virtual Machines eine immer geringere Rolle. Und die ausgeklügelten Virtualisierungstechniken von Mainframes, die auf

Hardware- und Softwareebene eine gute Isolierung garantieren, sind leider bei den kleinen Systemen nicht in dieser Art verfügbar.

In der letzten Zeit hat sich mit dem Spannungsfeld Usability versus Security ein weiterer wichtiger Aspekt sicherer Software bemerkbar gemacht. Angesichts der großen Zahl an semantischen Attacken wie Phishing oder Identity Spoofing, die die Benutzer momentan gefährden, zeigt sich, dass Software nicht mehr nur theoretische Sicherheit gewährleisten, sondern für den Nutzer auch verständlich und beherrschbar bleiben muss. Die Forderungen nach besserer Usability dürfen dabei nicht nur für Endbenutzer gelten, sondern müssen auf die sicherheitsrelevanten Werkzeuge der Entwickler selbst ausgedehnt werden.

Auf eine fundamentale Besserung der Sicherheit von Software zu hoffen, ist zumindest für die nächsten Jahre nicht angebracht. Die Rede von Bill Gates bei der RSA Konferenz 2005 ([Gates]) hat deutlich gemacht, dass sich die Maßnahmen von Microsoft zur Sicherung der Heimrechner auf den Aufbau einer verteilten Infrastruktur konzentrieren – sozusagen der Nachbau der Verwaltungsstrukturen, wie sie in großen Firmen heutzutage üblich sind. Große Data Center überwachen die einzelnen Rechner, erkennen Attacken und installieren Patches automatisch – gegen Gebühr.

Man darf aber nicht verkennen, dass dies eine Maßnahme nach bekannt gewordenen Attacken darstellt, also gegen so genannte Zero-Day Attacken nicht helfen wird. Wer sich nun von Open-Source-Software grundlegend Besserung erhofft, wird enttäuscht: Auch Linux unterscheidet sich hinsichtlich der Sicherheitsarchitektur keineswegs so deutlich von einem Windows basiertem System, dass hier Wunder zu erwarten wären. Die gleiche Beobachtung lässt sich im Vergleich der Open-Source Web-Browser Mozilla bzw. Firefox und dem Microsoft Internet Explorer machen.

Realistisch bleibt deshalb als Lösung auf lange Sicht nur eine auf die Bedürfnisse der Softwareentwickler zugeschnittene Ausbildung im Bereich sicherer Software. Diese Ausbildung muss:

- die Wahrnehmung von Sicherheitsproblemen massiv verbessern. Die Erfahrung hat gezeigt, dass dies am Besten durch häufige Sicherheitsanalysen geschieht, in Verbindung mit der Einführung von Sicherheitsprinzipien, die dann am konkreten Fall erprobt bzw. wieder entdeckt werden. Dieses Buch soll genau diesem Zweck dienen, nämlich Sicherheit als Gesamtsystem begreifbar zu machen und die Verbindung von Policies und Mechanismen zu erläutern.
- das Verständnis von Sicherheitsmechanismen und Techniken vermitteln. Die Erfahrung hat jedoch gezeigt, dass dies am Besten im Kontext eines realen Sicherheitsproblems geschieht und nicht davon losgelöst. Im Anschluss an eine konkrete Anwendung empfiehlt sich dann eine vertiefte Bearbeitung ausgewählter Fragestellungen. Typisches Beispiel ist hier der unterschiedliche Einsatzzweck von Kanal- bzw. Nachrichtenorientierter Kommunikation.
- sicherheitsrelevante Softwarearchitektur im Bereich Authentisierung und Authorisierung am konkreten Beispiel (z. B. Single-Sign-On im Portal) lehren.

- grundlegende Prinzipien sicherer Software wie POLA und deren technische Realisation auf den verschiedensten Ebenen (Betriebssystem, Sprache, Applikationsarchitektur) verstehen und in konkrete Softwaretechnik umsetzen helfen. Design Patterns für sichere Software helfen dabei.
- „weiche Faktoren“, wie die konzeptuellen Modelle von Benutzern, aber auch Entwicklern, als wichtig erkennen. Hier überschreitet man schnell die Grenzen der Informatik hin zu einer ganzheitlichen Betrachtung des Phänomens sichere Software. Dazu gehört beispielsweise die Kenntnis von Usability-Kriterien, aber auch die Prinzipien von Angriffen, die auf der Täuschung der Nutzer basieren, wie sie Kevin Mitnick in [Mit] eindrucksvoll dokumentiert hat.

Kernziel dieses Buches ist deshalb das Aufzeigen der Zusammenhänge zwischen Sicherheitsaspekten in der Infrastruktur, in der Software der Applikationen und Systeme sowie den Benutzern und Entwicklern dieser Systeme. Kryptografische Grundlagen und Protokolle sind für uns hauptsächlich interessant in Bezug auf die Konsequenzen, die ihre Eigenschaften für die Anwendungen nach sich ziehen.

Gleiches gilt für den Bereich der Netzwerksicherheit: Auch sie ist für uns nur interessant in Bezug auf das Zusammenspiel mit der Applikationssoftware, aber nicht als eigenständiges Gebiet. Zudem gibt es für diesen Bereich bereits etliche hervorragende Einführungen ([Bless], [Schä], [Schw]).

Die Schwierigkeiten bei der Wahrnehmung von Sicherheitsproblemen darf jedoch nicht darüber hinwegtäuschen, dass für die Erstellung sicherer Systeme eine gewisse Menge an sicherheitstechnischen Grundlagen nötig ist. Dieses Buch versucht diese Grundlagen auf eine Weise zu erklären, die es Software-Entwicklern erlaubt, sie auch in der Praxis zu erkennen und korrekt anzuwenden.

Der Aufbau des vorliegenden Buches gliedert sich wie folgt:

Zunächst wird in konkreten Fallstudien die Aufmerksamkeit für Sicherheitsprobleme geschärft und die Sicherheitsproblematik eingebettet in das Gesamtkonzept von Applikationen und Geschäftsvorgängen. Zum Beispiel untersuchen wir die Sicherheitsproblematik, die hinter großen Portalen steht und versuchen dabei zu zeigen, dass eine künstliche Einschränkung auf ein einziges Bedrohungsmodell (z. B. das Internet-Bedrohungsmodell) nicht mehr sinnvoll ist. Stattdessen muss der Gesamtkontext bestehend aus Benutzer, Nutzer-PC, Internet, Applikation, Entwickler und Infrastruktur betrachtet werden. Zur Gesamtsicht eines Sicherheitskonzeptes gehört auch seine Auswirkung auf die Betriebsorganisation (Ausbildung von Personal, Hilfe-Center bei Problemen mit der Sicherheit etc.), die meist erst sehr spät in den Blickwinkel von Entwicklern geraten und dadurch unvorhergesehene Kosten oder neue Sicherheitslücken verursachen können.

An die Fallstudien schließt sich ein Grundlagenteil an, der einzelne Aspekte aus den Fallstudien herausgreift und vertieft behandelt. Er beginnt mit einigen Anmerkungen zur Sicherheitsanalyse und ihren Komponenten, wie zum Beispiel Single-Sign-On oder Delegation. So wird die Problematik der Delegation von Requests innerhalb von Infrastrukturen erklärt sowie auf die Unterschiede von kanalbasierter und objektbasierter Sicherheit eingegangen. Die Sicherheitstechni-

ken verteilter Systeme und ihrer Middleware werden ebenfalls in diesem Grundlagenteil diskutiert. Darunter fällt etwa die Frage der Authentisierung in verteilten Systemen. Aus dem Bereich der Kryptographie werden wesentliche Bausteine zum Bau sicherer Protokolle vorgestellt, die Voraussetzung für das Verständnis der Sicherheits-Frameworks in späteren Kapiteln sind. Von besonderer Bedeutung ist dabei der Unterschied zwischen kanalbasierter Sicherheit (z. B. durch die Verwendung von SSL) und der objektbasierten Sicherheit durch die Verwendung von signierten Nachrichten. Das Verständnis objektbasierter Sicherheit ist die Voraussetzung für die Einführung von neuen, flexiblen Geschäftsmodellen durch Föderation.

Das Buch wird abgeschlossen mit drei Kapiteln, in denen wir wichtige Anwendungsfelder betrachten: Die Sicherheit von Content-orientierten Systemen, föderative Systeme und der Aufbau einer sicheren Infrastruktur für Internet-Applikationen. Rollenkonzepte, Benutzerberechtigungssysteme und die Problematik von Stellvertretung werden anhand von Content Management Systemen diskutiert. Als Sicherheitsmechanismus auf Content-Ebene wird der Einsatz von Label-basierter Security (LBAC) untersucht und rollenbasierter Sicherheit (RBAC) gegenübergestellt. Besonderes Interesse verdient LBAC nicht zuletzt deshalb, weil es eine wichtige Rolle innerhalb der User Access Control des neuen Vista Betriebssystems von Microsoft spielt. Es ergänzt hier die rein Permission-bezogene Zugriffskontrolle durch Acess Control Lists (ACLs).

Föderative Systeme beinhalten als wesentlichen Punkt die Übernahme einer bereits erfolgten Authentisierung durch andere. Wir werden sehen, dass sich dadurch nicht nur Ressourcen einsparen lassen, sondern durchaus auch eine Steigerung der Sicherheit erreichbar ist. Außerdem wird die Technik der web-basierten Föderation von Identität unter Wahrung von Privatheit gezeigt.

Im Bereich der Infrastrukur wird ein Konzept zur konsequenten Reduktion von Angriffsoberflächen innerhalb der DMZ sowie des Intranets vorgestellt. Aus der Architektur der DMZ werden Anforderungen für eine damit kompatible Applikationsarchitektur abgeleitet: Was kann/soll ein Application Server tun bezüglich Authentisierung? Nützt ein Proxy in der DMZ und welche Auswirkungen hat er für die Applikationsarchitektur? Wie weit ist die Applikationsarchitektur durch Regeln der DMZ betroffen? Die Reverse Proxy Architektur zur Authentisierung und Authorisierung wird an einem konkreten Produkt erläutert.

Dem aufmerksamen Leser ist wahrscheinlich nicht entgangen, dass die Aufzählung der Themen in diesem Band nur einen Teil der oben von uns erwähnten Problematik bei der Entwicklung sicherer Systeme umfasst. Ein zweiter Band wird sich unter dem Titel „Sichere Systeme“ mit den existierenden Software-Frameworks zur Absicherung von Systemen beschäftigen. Die so genannte End-to-End Security innerhalb von Firmen wird darin ebenso behandelt wie die Absicherung von Servern auf den jeweiligen Plattformen. Anforderungen aus der DMZ Architektur werden in Form von Sicherheitsframeworks in J2EE bzw. Java erneut aufgegriffen und gelöst. Neben diesen softwaretechnischen Grundlagen erweitert der zweite Band das Thema der Software-Sicherheit um zwei wesentliche Dimensionen: Usa-

bility und Security einerseits, und andererseits die Einschränkung von Autorität als Grundprinzip bei der Entwicklung sicherer Systeme.

Der vorliegende Band basiert in Teilen auf Lehrveranstaltungen der Autoren zum Thema Internet-Sicherheit an der Hochschule der Medien in Stuttgart sowie auf eigenen Erfahrungen bei der Entwicklung von Portalen und anderen Applikationen. Es richtet sich vor allem an Softwareentwickler und Studierende der Informatik. Aber auch Spezialisten der Netzwerksicherheit, die eine Brücke zur Software-Security schlagen möchten (was in der letzten Zeit vor allem im Bereich Input Validierung geschieht), werden hoffentlich davon profitieren. Der Folgeband „Sichere Systeme“ geht darauf aufbauend sehr viel tiefer auf die Konstruktionsprinzipien von Software und ihren Auswirkungen für die Sicherheit von Systemen ein und richtet sich in noch stärkerem Maße an die Entwickler von Software. Nun aber zunächst viel Spaß beim Studium der Grundlagen der Internet-Security aus Software-Sicht.

Kapitel 2
Fallstudien

In diesem Kapitel werden wir uns mit zwei großen Internet-Portalen beschäftigen, die sicher jeder Leser schon einmal besucht hat, nämlich bahn.de und ebay.de. Portale bieten aus geschäftlicher Sicht sehr interessante Dienste an: Sie fassen die diversen Informationsquellen und Anwendungen der Unternehmen in einen homogenen Auftritt gegenüber Kunden und Mitarbeitern zusammen und heben dadurch die historisch bedingte Zersplitterung der internen Informationssysteme auf.

Aus IT-Sicht hingegen sind Portale komplexe Gesamtkunstwerke bestehend aus Hardware, Netzwerken und miteinander kommunizierenden Applikationen mit teils subtilen Abhängigkeiten untereinander. Vielen Softwareentwicklern – der Zielgruppe dieses Buches – bereiten gerade diese Abhängigkeiten zwischen der Software und der Netzwerkarchitektur große Probleme. Anders gesagt: Es sind die so genannten „Non-Functional Requirements“ hinter einem Portal – also Performance, Zuverlässigkeit etc., die von der Software zumindest mitbestimmt werden, den Entwicklern aber alles andere als klar sind, da sie sich über verschiedene IT-Bereiche einer Firma erstrecken.

Entsprechend interessant sind auch die Sicherheitsaspekte eines Portals, die sich von der – meist im Internet befindlichen – Kundenseite über die Ankopplung an das eigene Netz bis tief in die eigenen Backend Systeme des Intranets erstrecken. Häufig kommen noch Einbindungen externer Partner dazu.

Es gibt noch eine Reihe weiterer Gründe, warum die Sicherheit in Portalen oft als schwierig empfunden wird:

- Altsysteme müssen in großer Zahl angesteuert werden.
- Entwickler, die bisher interne Applikationen entwickelt haben, müssen sich in der raueren Welt der Internet Applikationen zurechtfinden.
- Inkompatible Systeme für Authentisierung und Authorisierung von Kunden und Mitarbeitern müssen integriert werden.
- Evtl. ist die Zusammenarbeit mit fremden Systemen nötig, um z. B. bereits authentisierte Kunden zu übernehmen oder Aktionen von Kunden an andere Firmen weiterzuleiten.

- Organisatorische Trennungen in Internet- und Intranetabteilungen erschweren den Wissensaustausch und schaffen Unklarheiten bezüglich der Sicherheitsstufe von Informationen.
- Komplexe Frameworks wie J2EE oder .NET unterstützen Entwickler zwar, indem sie den zu schreibenden Sicherheitscode reduzieren, allerdings um den Preis dass Entwickler die Abstraktionen dahinter verstehen müssen.

Somit sehen sich die Entwickler im Portalbau mit einer ganzen Reihe sicherheitsrelevanter Fragestellungen konfrontiert:

- Welche Eigenschaften muss ein Softwareprodukt besitzen, damit es in die existierende Sicherheitsarchitektur integriert werden kann? Dies ist wichtig bei Fremdprodukten, aber auch für die Entwicklung von Inhouse Software. Genügt es zum Beispiel, einen LDAP Anschluss anzubieten zur Authentisierung über ein Fremdsystem?
- An welcher Stelle in der Infrastruktur müssen die Portalrechner und Komponenten platziert werden damit sie aus dem Internet erreichbar sind und die interne Infrastruktur nutzen aber nicht gefährden können? Welche Anforderungen stellt eine solche „De-Militarized Zone“ (DMZ) an meine Software? Welche Ports werden normalerweise geöffnet? Welche Protokolle sind erlaubt, welche vorgeschrieben? Wie bekommt man die nötige Software in die DMZ? Wie erfolgt die Wartung und Administration?
- Wie muss eine solche Applikation abgesichert werden, das heißt sowohl netzwerktechnisch als auch in der internen Programmierung? Wie werden Server abgesichert? Was darf/soll die Applikation sicherheitstechnisch tun? Welche Services von Frameworks müssen genutzt werden?
- Mit welchen Attacken von intern oder extern muss gerechnet werden? Wie weit geht die Verantwortung der Applikation gegenüber dem Kunden?
- Wie müssen Rollen und Gruppen als Mittel der Zugriffskontrolle implementiert werden, damit sie skalieren, aber auch bei organisatorischen Änderungen leicht angepasst werden können?
- Welche Anforderungen an die Verfügbarkeit – ein wichtiger Sicherheitsdienst und wesentlicher Teil der Security eines Portals – werden gestellt und wie sind sie zu erfüllen?
- Mit welchen Identitäten bzw. Rechten müssen Infrastrukturkomponenten wie etwa Web Server laufen? Wie erfolgt dementsprechend der Zugriff auf Backend-Systeme und wie breit/offen ist dieser Zugriff?

Diese und weitere Fragen wollen wir auf den folgenden Seiten beantworten – zunächst im Kontext realer Anwendungen und dann anhand spezieller Modellkonfigurationen. Gerade der Kontext eines realen Portals ist wichtig für das erste Ziel dieses Buches, nämlich die Wahrnehmung von Sicherheitsproblemen bei Softwareentwicklern zu schärfen. Viele der Probleme und Begriffe, die wir in späteren Kapiteln dieses Buches genauer analysieren, werden in unseren Fallstudien eingeführt.

Als erstes konkretes Beispiel für ein komplexes Portal soll uns der Internet-Auftritt der Deutschen Bahn AG – bahn.de – dienen. Anhand der Geschäftsvorgänge für dieses Portal werden wir zentrale Sicherheitsanforderungen an das Portal ableiten. Als zweites betrachten wir die Internet-Auktionsplattform ebay.de, die in der Vergangenheit durch einige Angriffe in die Schlagzeilen geraten ist.

2.1 Online-Kartenkauf am Beispiel bahn.de

Die Sicherheit einer Applikation oder Infrastruktur beginnt nicht mit Protokollen oder Algorithmen wie SSL oder SHA-1, sondern mit dem Geschäftsmodell und den Geschäftsvorgängen, die durch die Applikation lediglich umgesetzt werden. Die Risiken hinter den Geschäftsvorgängen müssen bestimmt und in ihren Auswirkungen geschätzt werden, dann erst kann über Sicherheitstechniken nachgedacht werden, um die Risiken zu begrenzen. Und es muss genau überprüft werden, ob die gewählten Sicherheitstechniken den intendierten Geschäftszweck tatsächlich unterstützen, oder im Extremfall sogar neue Unsicherheiten mit sich bringen.

Damit soll nicht geleugnet werden, dass auch Sicherheitstechniken wie SSL unabhängig von ihrer Anwendung gewisse Risiken in sich bergen können: Der automatische Rückfall auf kryptografische Verfahren geringerer Qualität oder die Gefahren bei der Wiederaufnahme von Sessions sind Beispiele dafür, die wir weiter unten aufgreifen werden. Für Entwickler ist es aber zunächst am wichtigsten zu begreifen, dass es die Anforderungen aus den Geschäftsvorgängen sind, die die Sicherheitstechnik bestimmen.

Sicherheit ist zudem ein Prozess – wie der bekannte Security-Spezialist Bruce Schneier (siehe etwa [Schn01], S. 264) nicht müde wird zu betonen – und das heißt, dass alle Beteiligten – Manager wie Entwickler – auch über ein Betriebskonzept des Portals verfügen müssen. In diesem Sinne stellt die Entwicklung eines Sicherheitskonzeptes zwangsläufig einen Top-Down Prozess dar, der mit einem Überblick des Portals und seiner Funktionalität beginnt und den ganzen Lebenszyklus der Lösung umfasst.

2.1.1 Überblick

Eine Website zum Nachsehen von Fahrterminen und Preisen gibt es bei der Bahn schon lange. Im Sommer 2002 jedoch wurde die Funktionalität drastisch erweitert durch die Einführung des Online-Ticket-Verkaufs. Kunden können seitdem bequem von zuhause aus Tickets und Reservierungen bestellen und das Ticket komplett am Drucker daheim ausdrucken.

Diese Umstellung brachte einen großen Aufwand auch in der so genannten Betriebsorganisation mit sich: Schaffner mussten geschult werden und in der Lage sein, Online-Tickets direkt und unmittelbar zu validieren. Neue Geräte mussten

angeschafft oder alte angepasst werden, mit denen die Validierung durchgeführt werden konnte. Dies ist ein Aspekt, den gerade Entwickler gerne übersehen: Die eingesetzte Sicherheitstechnologie besitzt Auswirkungen auf die gesamte Arbeitsweise einer Firma und beeinflusst die Betriebsorganisation beträchtlich.

Gleichzeitig entwickelte sich das Portal immer mehr zu einer übergreifenden Plattform. Fremdwerbung tauchte auf, Partnerfirmen mit erweiterten Angeboten bis hin zur Kreditvergabe und Reiseplanung wurden eingebettet (s. Abb. 2.1). Aus dem Kundenportal der Bahn wurde damit ein so genanntes *föderatives Portal* – ähnlich der Entwicklung, wie man sie an den realen Bahnhöfen verfolgen kann, in denen die Bahn selbst häufig nur noch die Verkaufsplattform sowie den Kundenkontakt zur Verfügung stellt, aber die eigentlichen Verkaufsaktivitäten (Bäcker, Cafés, Bücher, Lebensmittelläden etc.) von Fremdfirmen erledigt werden.

In der Anfangsphase des Portals traten häufig Probleme im Bereich Performance auf. Wer am Sonntagabend einen Zug für Montagmorgen buchen wollte, sah sehr häufig die Meldung „Server nicht verfügbar“, speziell wenn es ans virtuelle Bezahlen ging. Falsche oder abgelaufene Zertifikate der Bahn selbst oder von Partnerfirmen trugen zusätzlich zur Verwirrung der Kunden bei.

Keineswegs einfach war auch die Umsetzung der Kontrollen des Online-Tickets – das sich übrigens im Laufe der Zeit kaum gewandelt hat – durch die Betriebsorganisation. Bei Reklamationen beispielsweise durfte der Online-Kunde sich nicht mehr an die überall verfügbaren Bahnschalter wenden. Kunden, die

Abb. 2.1 Online-Portal der Deutschen Bahn AG

einen Anschlusszug auf Grund von Verzögerungen des Zubringers nicht erreichen, müssen ihre verfallenen Reservierungen an einer bestimmten Stelle vorlegen. Offensichtlich sind also Online-Ticket und Normalticket in der IT-Infrastruktur in ganz verschiedenen Systemen untergebracht.

Sehr interessant war die Reaktion der beteiligten Schaffner die – ausgerüstet mit ihren mobilen Rechnern – die Online-Tickets prüfen mussten. In der Anfangszeit herrschte große Verwirrung über die Frage, wer wann welche Tickets zu prüfen hatte. Sollte nach jedem Umsteigen geprüft werden? Wie sollte das Ticket „entwertet", das heißt, wie kann die Mehrfachnutzung eines Tickets verhindert werden? Hier galt es für die Bahn ein hinreichend flexibles System zu finden, das einerseits umfassende Kontrollen erlaubt und andererseits den Betrieb nicht zu sehr aufhält (und vielleicht auch die Kunden nicht über Gebühr belastet). Schon diese anfänglichen Überlegungen zeigen, dass das Herstellen von „Sicherheit" aus geschäftlicher Sicht nicht unbedingt ausschließlich Risikovermeidung bedeutet, sondern häufig vielmehr den betriebswirtschaftlich sinnvollen Umgang mit Risiken beinhaltet.

Die Bedeutung von bahn.de für die zukünftigen Services der Bahn kann kaum überschätzt werden. So erzeugt das Portal bereits im Jahr nach der Einführung 1,6 Millionen Buchungen, 2004 wurden 3,8 Millionen und 2005 insgesamt 6,1 Millionen Tickets auf diesem Wege erworben. In 2006 wurden bereits 17% aller Buchungen online durchgeführt [heise83744].

2.1.2 Geschäftsmodell

Das Geschäftsmodell des Portals besteht aus dem Verkauf von Tickets und dem Anbieten weiterer Services, entweder durch die Bahn selbst oder vermittelt an Partnerfirmen. Die Besonderheit liegt in der hauptsächlich elektronischen Abwicklung aller Vorgänge, die dem Kunden einerseits Wege und Kosten spart und andererseits der Bahn Personalkosten in Verkaufstellen erspart. Ähnlich den elektronischen Vorräumen von Banken (e-banking, ATMs) besteht auch hier der Zwang, dass sämtliche Vorgänge automatisch und fehlerfrei ablaufen müssen – anderenfalls lassen sich die geplanten Einsparungen wegen der dann nötigen Mehrausgaben für den Kundendienst (zum Beispiel für Call Center) nicht erzielen. Eine besondere Rolle spielt dabei die Art und Weise, wie mit dem Kunden kommuniziert wird (über das Portal, via E-Mail, über eine Hotline etc.).

Kiosk-Systeme in Supermärkten und anderen öffentlichen Plätzen erlauben den Erwerb von Tickets oder die Servicenutzung auch ohne Heim-Computer. Es kann erwartet werden, dass Kunden in Zukunft vermehrt PDAs mit Telefoniefunktion bzw. Smartphones für den Zugriff auf das Portal verwenden werden.

Einen weiteren interessanten Aspekt stellt die Möglichkeit der Profilbildung durch die Daten aus Kundentransaktionen dar – entweder auf individueller Ebene oder als Aggregatdaten. Durch die Vielfalt der angebotenen Services entstehen natürlich auch sehr aussagekräftige Daten, die in sich bereits einen hohen Wert

darstellen. Dazu passt, dass das Rabattsystem der Bahn im Privatkundengeschäft auf Einzelpersonen bezogen ist, das heißt, eine Bahncard 50 wird speziell für einen Kunden ausgestellt. Wenn nun bei Bezahl- oder Kontrollvorgängen die Bahncard verwendet wird, entsteht automatisch eine lückenlose Buchführung zum Reiseverhalten des Kunden – ein drastischer Unterschied zum anonym bezahlten und „abgeknipsten“ Ticket alter Schule.

Es bleibt noch das Problem des gemischten Kaufverhaltens (teils online, teils traditionell) am Schalter oder Automaten zu lösen. Kaufvorgänge, die offline durchgeführt werden, können mangels Identifizierung dem Kundenprofil nicht zugeordnet werden. Hier hilft die Einführung eines Bonuspunkte-Systems, wie es auch Supermärkte oder Kaufhäuser bereits kennen. Solche Systeme sollen unter anderem den Kunden dazu bringen, bei allen Kaufvorgängen seine Identität preiszugeben. Bei der Bahn findet die Datenkonzentration über die Bahncard des Kunden statt. So wird der Kunde seither auch am Schalter gefragt, ob er Punkte sammeln möchte. Dazu muss er natürlich identifiziert werden und dies geschieht durch die elektronisch lesbare Bahncard (die ansonsten beim Kauf eines herkömmlichen, nicht-elektronischen Tickets nicht nötig ist). Die teils beträchtlichen Rabatte, die bei solchen Bonuspunktesystemen eingeräumt werden, sind ein klarer Hinweis auf den großen Wert, den derartig erstellte Kundenprofile für die Firmen haben.

Wesentlich für das Geschäftsmodell ist, dass der Kunde das Portal als Anlaufstelle für eine ganze Reihe von geschäftlichen Vorgängen ansieht und diese durch einen einheitlichen Zugang nutzen kann. Dies betrifft vor allem die Frage der *Authentisierung*, also dem Nachweis einer behaupteten Identität, gegenüber weiteren Diensten.

Somit kann sich das Bahnportal seinerseits gegenüber den verschiedenen Geschäftsstellen der Bahn sowie Fremd- und Partnerfirmen als zentrale Stelle für die Verwaltung von Kundendaten und die Abwicklung von Finanztransaktionen anbieten. Voraussetzung dieses Geschäftsmodells ist natürlich, dass die Kunden Tickets und Services sicher und einfach bestellen können und die Abrechnung der Leistungen korrekt erfolgt.

2.1.3 Kundensicht

Die Kundensicht bestimmt die Funktionsweise eines Portals zu einem großen Teil. Das Portal bahn.de konzentriert sich auf den Einzelkunden. In der Vergangenheit konnten Privatpersonen unter der Voraussetzung, dass eine Bahncard vorhanden war, Online-Tickets bestellen (mittlerweile hat sich dies geändert). Natürlich stellte sich schnell die Frage, was mit Firmenkunden passieren würde: In größeren Firmen werden Tickets oft zentral bestellt und abgerechnet, das heißt, es besteht in diesem Fall keine 1:1 Beziehung zwischen Kunde und Portal. Dieses Problem wurde über einen separaten Zugang für Firmenkunden gelöst, der von dem hier betrachteten Portal unabhängig ist.

Ebenfalls zur Kundensicht gehört die Frage, was mit den Mitarbeitern der Bahn selbst geschehen soll. Wenn der Besitz einer Bahncard plötzlich Voraussetzung für Online-Tickets ist, wie kommen dann beispielsweise Mitarbeiter der Bahn, die bisher nur einen speziellen Ausweis besaßen, zu Tickets? Wenn ein Mitarbeiter über das Portal etwas bestellt – handelt er dann als Mitarbeiter oder Kunde? Ein ähnliches Problem betrifft die externen Partnerfirmen und deren Mitarbeiter: Welche Rolle spielen sie und wie werden sie identifiziert? Die Administratoren eines solchen Portals sowie die Mitarbeiter des Call Centers (falls es eines gibt) gehören ebenfalls zur Gruppe derer, die letztlich Zugriff auf das Portal erhalten müssen. Kann diese Administration dann „outgesourced" werden? Welche Konsequenzen hätte dies für die Sicherheit des Zugangs?

Die Modellierung des Personenkreises, der Zugriff auf das Portal hat, ist eine der schwierigsten Aufgaben. Aus technischer Sicht tauchen die Probleme vor allem in den Gebieten *Authentisierung* (der sicheren Verifikation einer Kunden- oder Mitarbeiteridentität) und *Authorisierung* (der Zuweisung einer Rechtemenge an einen authentisierte Nutzer) auf. Ein klassisches Beispiel für diese Problematik stellen temporäre Kunden dar, die vielleicht im Rahmen einer Werbeaktion für begrenzte Zeit kostenlosen Zugriff auf bestimmte Services erhalten. Sie brauchen eine echte Identität im System, die jedoch automatisch nach festgelegter Zeit verfallen muss – oder der Kunde wird ein „fester" Kunde: Können seine bisherigen Einstellungen und die Historie seines Verhaltens migriert werden oder beginnt er als registrierter Kunde mit einer neuen Identität?

Eng mit dem Konzept Kunde bzw. Partner ist auch die Frage der Identität im System verknüpft: Darf der Kunde sich einen eigenen Namen/UserID aussuchen? Gibt es eine vorgeschriebene Syntax? Legt das Portal hier Regeln fest?

Schließlich ist ein letzter Punkt von wesentlicher Bedeutung für das Geschäftsmodell: Tritt bahn.de als selbständige Authentisierungsstelle auf, das heißt identifiziert das Portal die Benutzer selbständig (was auch die initiale Registrierung bisher unbekannter Kunden einschließt) oder schließt es sich einem Verbund mehrerer Firmen an, in dem die Authentisierung von einer gemeinsamen, zentralen Instanz übernommen wird? Hier hat sich die Bahn zunächst für die Selbständigkeit entschieden – wohl nicht zuletzt, um die wertvolle direkte Kundenbeziehung nicht zu gefährden. Der Preis dafür ist die Notwendigkeit, eine eigene Infrastruktur für das Identitätsmanagement aufbauen zu müssen, wie sie weiter unten beschrieben wird. Damit bildet bahn.de eine eigene so genannte Sicherheitsdomain oder *Realm*.

2.1.4 Geschäftsvorgänge

Welche Geschäftsvorgänge müssen beim Aufbau eines Sicherheitskonzepts für das Portal betrachtet werden? Hier eine (sicher unvollständige) Liste:

- Ein neuer Kunde registriert sich und erhält so genannte *Credentials* (das heißt Username und Passwort) für zukünftige Anmeldungen.
- Ein unbekannter Kunde informiert sich anhand der öffentlichen Seiten des Portals.
- Eine Firma bestellt ein Ticket für eine Mitarbeiterin (ausgelagert in Firmenportal).
- Ein Kunde storniert einen Auftrag und muss eine Vergütung erhalten.
- Ein Kunde meldet sich an, kauft ein Ticket und benutzt Services und Angebote von Partnerfirmen.
- Ein Kunde verliert oder vergisst seine Credentials und fordert neue an.
- Ein Sachbearbeiter vollzieht einen problematischen Vorgang anhand von Vorgangsdaten nach.
- Mitarbeiter von Partnerfirmen beantragen Zugriff auf Stammdaten der Kunden.
- Ein Kaufvorgang wird abgerechnet, z. B. über Kreditkartenfirmen.
- Ein Kunde bestellt kurz vor Abfahrt des Zuges ein Ticket.
- Ein Kunde bestellt eine Reservierung.
- Ein Kunde will seine Stammdaten ändern (andere Karten, Adressen etc.).

Es ist sicher sinnvoll, auch negative Geschäftsvorgänge aufzunehmen:

- Ein Kunde verwendet eine gestohlen oder verloren gemeldete Bahncard.
- Zwei Kunden versuchen, das gleiche Online-Ticket zu verwenden.
- Ein Kunde hat sein Online-Ticket vergessen, hat jedoch eine elektronische Kopie auf dem Laptop dabei.
- Ein Kunde streitet einen Ticket- oder Servicekauf ab.
- Ein Kunde kann Tickets oder Services nicht bezahlen.

Die reguläre Bearbeitung von Vorgängen, aber auch administrative Tätigkeiten erfolgen schließlich durch das interne Komponentenmodell. Datenarten, Datenhaltung etc. werden hier modelliert und für Bearbeiter zugreifbar.

2.1.5 Sicherheitsanforderungen

Aus den Geschäftsvorgängen lässt sich dann unter Berücksichtigung verschiedener Bedrohungsmodelle der so genannte *Security Context* erstellen, in dem eine ganze Reihe von Sicherheitsanforderungen an das Portal festgehalten werden. Auf diese Anforderungen gehen wir im Folgenden näher ein.

2.1.5.1 Schutz der öffentlichen Bereiche

Das Portal soll frei zugängliche Informationen anbieten, und zwar über Fahrziele und Fahrpläne, Preise, Sonderangebote und sonstige Dienstleistungen. Diese Informationen sollen die korrekte Antwort auf entsprechende Anfragen der Kunden dar-

stellen. Ebenso soll der Kunde sicher sein können, dass die in seinem Browser angezeigten Informationen korrekt sind und auch tatsächlich von der Bahn stammen.

Diese recht einfach klingenden Anforderungen beinhalten bereits eine Fülle von Sicherheitsproblemen und offenen Fragen, die sich auf drei Bereiche konzentrieren:

- Identifikation des Portals
- Produktion der Information
- Auslieferung der Information

Zunächst stellt sich für die Bahn das Problem, wie der Kunde überhaupt das Bahnportal findet und erkennt. Firmen meinen oft, das Problem der Identifikation durch die Verwendung einprägsamer Domain-Namen und eines durchgängig gestylten „Look&Feel“ lösen zu können. Um darüber hinaus die Kunden vor Angreifern zu schützen, die ihr Look&Feel kopieren und leicht abgeänderte Domain-Namen verwenden, verwenden die Firmen meist das Sicherheitsprotokoll SSL (siehe Kap. 7). Dabei kommen so genannte *Zertifikate* zum Einsatz, das sind elektronische Bestätigungen durch eine vertrauenswürdige Instanz, dass der Server mit dem Namen www.bahn.de der Deutschen Bahn AG gehört und darüber hinaus einen bestimmten öffentlichen Schlüssel besitzt. Auf die damit verbundenen Mechanismen der Identifikation und Authentisierung wird weiter unten noch genauer eingegangen, hier nur eine etwas pessimistische Vorbemerkung dazu: Es gibt kein rein technisches Mittel, mit dem man sicherstellen könnte, dass die Intention des Kunden über die Identität seines Kommunikationspartners mit der Realität übereinstimmt, denn das Look&Feel des originalen Bahn-Servers lässt sich fälschen, und welcher Kunde würde schon auf den Unterschied zwischen der „Deutsche Bahn AG“ und einer (fiktiven) „Deutsche Bahn GmbH“ achten, die im Server-Zertifikat als Inhaber genannt werden?

Letztlich ist dies auch das Problem, das dem in der letzten Zeit so populär gewordenen *Phishing* (ein Kunstwort aus „Password“ und „Fishing“) zugrunde liegt, bei dem Kunden durch eine authentisch aussehende E-Mail und einen ähnlich lautenden Domänennamen auf eine identisch aussehende Website gelockt wird. Wenn der Kunde seine Credentials (UserID, Passwort) dort eingibt, um sich zu authentisieren, übergibt er sie tatsächlich an einen Angreifer, der sie sofort missbraucht.

Dieses grundsätzliche Identifikationsproblem im Internet lässt sich nur mit Hilfe aufwändiger Maßnahmen wie etwa vorinstallierten Applikationen, so genanntem Key Continuity Management oder der separaten Signierung jeder einzelnen Transaktion durch den Kunden in den Griff bekommen (so genannte *objektbasierte Sicherheit*), die alle auch mit gewissen Kosten verbunden sind.

Die Produktion der korrekten Informationen und ihre Veröffentlichung im Internet ist eine interne Angelegenheit des Portals. Da sich die publizierte Information nicht vollständig formal und automatisch prüfen lässt, sichern die meisten Firmen ihre Informationen durch eine Mischung aus technischen Prozessen und organisatorischen Maßnahmen. Dies dient sowohl der Qualitätssicherung als auch

dem Schutz vor Klagen auf Grund falscher Informationen. Zu den Anforderungen an einen sicheren Veröffentlichungs-Prozesses gehören:

- Abklärung, welche Daten öffentlich sind (Datenklassifizierung)
- Authentisierung aller Redakteure
- Zuteilung und Verwaltung von Gebieten und Zuständigkeiten (Authorisierung) durch Rollen
- Versionskontrolle aller Änderungen
- Archivierung einmal gezeigter Information
- Absicherung dynamischer Inhalte durch rollenbasierte Zugriffsmodelle auf Datenbanken und die Verwaltung von Templates durch einen Software-Deploymentprozess
- Auditing aller Zugriffe von extern wie intern
- Monitoring der Verfügbarkeit der Systeme

Portale, die außerdem extrem kritische Daten beinhalten, können als Ergänzung zu diesen organisatorischen und technischen Maßnahmen auch noch ein so genanntes *Mandatory Access Control* (MAC) System einsetzen, das durch *Daten-Labeling*, also ein Einteilen der Dokumente in unterschiedliche Vertraulichkeitsstufen, verhindert, dass geheime Daten auf diese Weise nach draußen gelangen können.

Schließlich soll der Kunde sicher sein, dass die Informationen von der korrekten Quelle ausgeliefert werden und auf korrekte, unverfälschte Weise bei ihm ankommen. Was aber bedeutet „korrekt“ in diesem Zusammenhang? Soll der Kunde beispielsweise die Informationen in einer Form erhalten, die er später in einer rechtlichen Auseinandersetzung verwerten kann? Soll der Kunde also im Nachhinein beweisen können, dass bestimmte Informationen so und nicht anders auf www.bahn.de zu sehen waren? Dann müssen die Inhalte durch das Portal digital signiert werden. Oder genügt es bahn.de sicherzustellen, dass die Informationen auf dem Weg zum Kunden nicht verfälscht werden können? Dann muss das Portal lediglich für einen sicheren Kanal zum Kunden sorgen. Oder sollte das Portal davon ausgehen, dass die von ihm angezeigten öffentlichen Informationen ohnehin nicht gefälscht werden und überträgt sie deshalb ohne Absicherung? Dies spart auf jeden Fall Infrastrukturkosten auf Seiten des Portals. In diesem Zusammenhang muss geklärt werden, ob bereits die Aufforderung zur Authentisierung über eine sichere Verbindung erfolgen soll, oder ob diese erst bei der Übertragung der Daten aufgebaut wird. Selbst Bankportale (z. B. das der Citibank) haben sich in der Vergangenheit hier falsch entschieden.

Sollte sich das Portal für eine sichere Übertragung oder gar für signierte Dokumente entscheiden, dann muss es auch die Verantwortung dafür übernehmen, das heißt, es ist Aufgabe des Portals, sicherzustellen, dass die dazu nötigen Verfahren zur Absicherung mit ausreichender Sicherheit durchgeführt werden (was Konsequenzen für die Einstellungen der Server, die verwendeten Schlüssel etc. mit sich bringt).

Manche Portale versuchen ihre Seiten und Links so zu gestalten, dass es möglich ist festzustellen, ob eine Seite von einem bestimmten Portal erzeugt und aus-

geliefert wurde. Dies hilft, gewisse Attacken (so genanntes *Cross-Site-Request-Forging*) zu vermeiden. Hier muss zwischen den Gütern „Performance durch Caching“ und „Sicherheit“ abgewogen werden. Öffentliche read-only Informationen können beispielsweise gut in einem Cache gespeichert werden – gegebenenfalls am so genannten „Rand“ des eigenen Netzwerks (edge caching) – und damit häufige und teure Zugriffe auf die Backendsysteme vermieden werden.

Der Zugriff auf öffentliche Informationen durch die Kunden birgt für das Portal jedoch noch eine weitere Gefahr: Dadurch ist es *jedem* Client erlaubt, Requests an das Portal zu stellen. Somit können also auch automatische Scripts, die auf Hunderten von Rechnern laufen, gleichzeitig solche Requests stellen – eine so genannte *Distributed Denial of Service* (DDoS) Attacke, bei der eine große Zahl von Rechnern unter der Kontrolle von „Bots“ eine automatische Attacke gegen einen bestimmten Host fahren.

Wie sich das Portal gegen solche Attacken wehren kann, wird weiter unten im Kapitel „Sicherheit der Infrastruktur“ beschrieben. Hier wollen wir einstweilen nur festhalten, dass öffentliche Zugriffe keine Gefahr für interne Netzwerke des Portals darstellen dürfen und dass der Aufbau einer Sicherheitszone (DMZ) für das Portal und seine öffentlichen Inhalte Pflicht ist.

Eine weitere wichtige Sicherheitsfrage im Zusammenhang mit öffentlichen Inhalten ist, ob und welche Schreibvorgänge erlaubt sein sollen. Sollen zum Beispiel auch nicht-registrierte Kunden ein Feedback Formular ausfüllen oder eine Nachricht senden dürfen? Der technische Hintergrund hierfür besteht darin, dass das Halten von State serverseitig besonders kritisch ist da es Ressourcen verbraucht. Dies kann unter Umständen ebenfalls zu DoS-Attacken führen.

2.1.5.2 Gesicherte, personalisierte Bereiche

Bestimmte Dienste wie die Bestellung von Tickets haben Vertragscharakter und benötigen dazu die Identität der Vertragspartner. Bestimmte Daten wie die Tickethistorie eines Kunden sind privater Natur und dürfen nach außen nur für den Kunden zugänglich sein.

Voraussetzung für die Nutzung dieser Dienste und Daten ist daher eine vorhergehende Authentisierung des Kunden. Jeder Zugriff ohne vorherige Authentisierung muss vom Portal abgewiesen werden. Insbesondere darf es einem Kunden nicht möglich sein, die Daten anderer Kunden zu sehen oder zu verändern (wie bei OBSOC geschehen). Das System hat also geeignete Vorkehrungen zu treffen, dass Kunden voneinander getrennt behandelt werden. Dies ist ein wichtiges Detail für die spätere serverseitige Implementation.

Diese 1:1 Beziehung zwischen Portal und Kunde wird in dem Moment aufgeweicht, in dem z. B. kollaborative Services wie Nutzerforen oder Wikis angeboten werden. Dann treten Kunden direkt miteinander in Kontakt und es entstehen neue Angriffsmöglichkeiten. Die Einführung solcher Services ist daher sowohl aus Business-Sicht als auch aus Sicht der Security gut zu überlegen.

2.1.5.3 Sichere Administration

Die Systemverwaltung stellt einen sehr kritischen Dienst dar. Die Dienste der Systemverwaltung des Portals sollten deshalb nur aus dem Intranet der Firma zur Verfügung stehen und nur einem ausgewählten Personenkreis. Aufgrund der Sensitivität dieses Dienstes gelten hier besonders hohe Sicherheitsanforderungen:

- Für die Anmeldung in einer Administratorrolle ist eine besondere Güte der Authentisierung vorauszusetzen. Das bedeutet insbesondere, dass Passwörter von hoher Qualität (zufällig gewählt, ausreichend lang) für die Administratorrolle zum Einsatz kommen und regelmäßig gewechselt werden. Oder noch besser: Es wird grundsätzlich auf stärkere Mechanismen der Authentisierung als Passwörter gesetzt.
- Die Administratorfunktion muss in mehrere Rollen aufgeteilt sein, das heißt, es darf keinen allmächtigen Systemverwalter geben wie das bei Unix der „Root“ oder bei Windows der „Administrator“ ist. Vielmehr ist darauf zu achten, dass jede Rolle nur die für die Erfüllung ihrer Pflichten nötigen Rechte erhält (*Need-to-know/Need-to-do* Prinzip).
- Des Weiteren müssen einzelne Abteilungen die Möglichkeit haben, eigene Subadministratoren zu benennen, die für Teilbereiche verantwortlich sind. Innerhalb dieser Teilbereiche können sie Rollen und Rechte von Sachbearbeitern festlegen (*Service Management Delegation*).
- Jede administrative Tätigkeit muss ausführlich protokolliert werden. Die Protokolle werden auf einem write-once Medium gespeichert (*Mandatory Audit*).

2.1.5.4 Sichere Sachbearbeitung

Innerhalb der Betriebsorganisation des Portals und der anschließenden Dienste der Bahn erhalten Mitarbeiter Zugriff auf Daten von Kunden zur weiteren Verarbeitung. Dazu gehören neben den Stammdaten natürlich auch Kreditkartennummern, Kartenprüfnummern, Bankverbindungen und weitere persönliche Daten. Als Teilnehmer am Kreditkartensystem unterliegt das Portal ohnehin den Regeln der Kreditkartenunternehmen, die die Einhaltung dieser Regeln durch so genannte *Audits* (online und lokal bei der jeweiligen Firma) prüfen. VISA z. B. hat eine Liste von Anforderungen veröffentlicht, die von beteiligten Firmen erfüllt werden müssen.

Neben allgemeinen Sicherheitsmaßnahmen z. B. durch Firewalls und Virenscanner setzt eine sichere interne Verarbeitung die Einführung von Authentisierung und Autorisierung auf Basis von Rollenkonzepten voraus. Konkrete Aktionen von Mitarbeitern sind anschließend zu protokollieren.

Dies ist ohne Zweifel einer der schwierigsten Aspekte sicherer Software und dennoch leider unumgänglich, wie Fälle wie der von Elliot Castro zeigen: Castro missbrauchte jahrelang fremde Kreditkarteninformationen, an die er z. B. über Tätigkeiten in Call Centern oder Firmen gelangte (s. [Castro]). Allerdings darf nicht verschwiegen werden, dass es häufig die mangelnde Betriebsorganisation

der beteiligten (Karten)-Firmen war bzw. der Wunsch nach mehr verkauften Karten, der die Betrügereien wesentlich erleichterte. Einen sehr interessanten Einblick in die Online-Unterwelt, in der Kreditkartendaten gesammelt, angeboten und weiter verkauft werden, gibt Tobias Knecht aus dem 1und1 Abuse Center in seinem Vortrag [Knecht].

2.1.5.5 Sichere Anmeldemöglichkeiten

Damit eine Bestellung einem bestimmten Kunden zugeordnet und der Bezahlvorgang eingeleitet werden kann, ist eine sichere Anmeldung der Kunden beim Portal scheinbar unabdingbar. Dazu werden üblicherweise einmalig die Stammdaten des Kunden abgefragt und dann ein Login mit zugehörigem Geheimnis erzeugt. Der Kunde ist dann gezwungen, bei jeder Buchung diesen Login zu verwenden.

Wie Erfahrungen mit Online-Shops zeigen, möchten jedoch viele Kunden gerne einfach mit ihrer Kreditkarte jeden Kauf bezahlen und dabei auf einen Login verzichten – wer will sich schon gerne für jeden Shop ein Passwort merken? Für diese Fälle hat bahn.de die Bezahlung mit Kreditkarte eingeführt – und zwar ohne zusätzlichen Login. Der Verzicht auf einen Login zeigt klar die Grenzen von Authentisierung im Geschäftssinn auf: Karten, Tickets, Zertifikate etc. ermöglichen Geschäftsvorgänge auch ohne vorherige genaue serverseitige Authentisierung. In Zukunft ist damit zu rechnen, dass sich diese Form von Absicherung von Geschäftsvorgängen weiter ausdehnen wird. Dafür ist die so genannte objektbasierte Sicherheit nötig, auf die wir später zurück kommen werden. Im Fall der direkten Buchungen mit Kreditkarte ist z. B. die Bestellung durch jemand anderen als den Kartenbesitzer kein Problem – im guten wie im schlechten Sinne. Hier gilt es im Geschäftsmodell die Vorteile durch die höhere Bequemlichkeit für den Kunden zu vergleichen mit den besseren Betrugsmöglichkeiten durch gestohlene Karteninformationen, und dann zu einer quantitativen Einschätzung des Risikos zu kommen.

Gehen wir nun davon aus, dass sich die Kunden tatsächlich bei bahn.de einloggen und authentisieren. Dann sind die wesentlichen Anforderungen des Portals bzgl. der Authentisierung:

- dass sich Benutzer selbständig online auf sichere (das heißt insbesondere vertrauliche und unverfälschte) Weise registrieren können;
- dass Benutzer ihre Stammdaten selbständig online verwalten können (z. B. Updates der Kreditkarteninformationen oder der Adresse), um Kosten zu sparen;
- dass die Authentisierung genügend sicher ist, um Tickets verkaufen zu können. Insbesondere darf es für einen Angreifer nicht möglich sein, sich unter einem falschen Namen beim Portal anzumelden;
- dass der Kunde eindeutig als rechtlich eigenständige Person identifizierbar sein muss, das heißt ein bloßes Pseudonym als Identität – wie es um Beispiel in Internet-Foren völlig ausreicht – ist hier nicht möglich. Anders ausgedrückt: der

Kunde muss unter einer Adresse erreichbar sein, unter der notfalls Forderungen eingetrieben werden können – eine bloße E-Mail Adresse reicht hierfür normalerweise nicht aus.

Ein interessantes Problem besteht in der Frage, ab wann ein neu registrierter Kunde kostenpflichtige Services nutzen kann. Darf nämlich jemand sofort nach der Registrierung beliebige Tickets bestellen, so kann das für Denial-Of-Service-Attacken ausgenutzt werden, sei es manuell oder durch Scripts, die automatisch neue Benutzer anlegen und Bestellungen erzeugen.

Wie bereits erwähnt, war lange Zeit der Besitz einer BahnCard Voraussetzung für die Nutzung des bahn.de Portals zur Bestellung von Online-Tickets. Dadurch konnte bereits bei der Registrierung ein Missbrauch eingeschränkt werden. Dies ist ein Beispiel für die geschickte Nutzung bereits bestehender Authentisierungen bei der Registrierung von Neukunden.

2.1.5.6 Single-Sign-On

Single-Sign-On (SSO) bedeutet, dass sich ein Kunde nur ein einziges Mal mit Username und Passwort authentisieren muss, um in der Folge mehrere unterschiedliche Dienste nutzen zu können. Dies betrifft:

- die Verwendung mehrerer portalinterner Dienste
- die indirekte – das heißt über Portaldienste vermittelte – Nutzung externer Dienste
- die Weiterleitung des Kunden an externe Dienste

Für ein Portal wie bahn.de, dessen Geschäftsmodell unter anderem in der Vermittlung der Kunden an externe Dienstleister besteht, ist Single-Sign-On sicher eine wünschenswerte Systemeigenschaft. Natürlich gelten für die initiale Anmeldung am Portal die gleichen Sicherheitsanforderungen wie oben besprochen. Single-Sign-On beinhaltet aber zusätzlich das sichere „Weiterreichen“ der authentisierten Kunden an weitere, evtl. auch externe Dienste. Die hierfür eingesetzten Techniken und die damit verbundenen Problematiken werden uns im späteren Verlauf noch mehrere Male begegnen.

In dem Maße, wie das das Portal „fremde“ Authentisierungen annehmen möchte – es muss sich hier nicht um „Sessions“ handeln, sondern Kunden könnten auch Gutscheine anderer Organisationen zur Bezahlung etc. einreichen – wird die Frage fremder Autoritäten eine größere Rolle spielen. Das gleiche gilt für Firmen, die sich dem Bahnportal anschließen wollen und dessen Authentisierungen bzw. Autorisierungen übernehmen möchten. Hier existieren sehr viele verschiedene Möglichkeiten, die von der Einrichtung vieler unabhängiger Instanzen zur Authentisierung über föderative Strukturen bis hin zu zentralen Systemen mit Identity Provisioning für beteiligte Firmen reichen. Wir werden diese Möglichkeiten im Kapitel zu föderativer Sicherheit besprechen.