Fachbereich 4: Informatik

Erstellung eines Kriterienkataloges zur Gewährleistung der technischen Sicherheit von E-Commerce-Webseiten

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Wirtschaftsinformatik
vorgelegt von

Leif Thorvald Franker

Betreuerin: Dipl. Inf. Anastasia Meletiadou
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich Informatik

Erstgutachter: Prof. Dr. Rüdiger Grimm
Institut für Wirtschafts- und Verwaltungsinformatik, Fachbereich Informatik

Zweitgutachter: Jean-Marc Noel, Geschäftsführer
Trusted Shops GmbH

Koblenz, im Mai 2007

Abstract

Das Ziel dieser Diplomarbeit ist es einen Kriterienkatalog zu erarbeiteten, anhand dessen die technische Sicherheit von E-Commerce-Webseiten automatisiert überprüft werden kann. Dazu wird wie folgt vorgegangen.

Nach einer Einleitung, die die Ausgangslage und die Motivation dieser Diplomarbeit zeigt, wird eine Aufbauanalyse von E-Commerce Webangeboten durchgeführt. Es werden gezielt die technischen Bausteine betrachtet, die für den Aufbau benötigt werden. Anschließend werden technische Schwachstellen (Fehlerquellen) von E-Commerce Webangeboten dargelegt. Der Schwerpunkt liegt dabei auf Programmierfehlern, Konfigurationsfehlern, Konzeptionsfehlern und Fehlern resultierend aus menschlichem Handeln. Parallel zu den Schwachstellen werden die jeweiligen Bedrohungspotentiale für einen E-Commerce Anbieter erörtert. Hierauf folgt eine kurze Vorstellung der generellen Risikopotentiale der verschiedenen Parteien, die bei einer E-Commerce Transaktion beteiligt sind.
In dem darauf folgenden Schritt werden etablierte Kriterienkataloge vorgestellt. Diese werden kritisch bewertet, inwieweit sie befähigt sind zur Erstellung des angestrebten Kriterienkataloges beizutragen. Es werden klassische Kataloge wie das Orange Book, Common Criteria aber auch aktuelle Kataloge wie der BSI Grundschutzkatalog begutachtet.
In dem nächsten Schritt werden die eigentlichen Kriterien in Listen einsortiert. Es wird differenziert analysiert "was" überprüft werden kann und "was" nicht. Daraufhin werden die Kriterien anhand von Schwachstellen in ein Ebenenmodell unterteilt. Das Ebenenmodell orientiert sich an Vorgaben des BSI und besteht aus der Protokollebene, der Dienstebene und der Anwendungsebene. Im Anschluss daran findet eine Gewichtung aller Kriterien anhand ihrer Automatisierbarkeit, ihrem Schadenspotenial und ihrer Verbreitung statt.
Im vorletzten Schritt werden Lösungen vorgestellt, auf welche Weise der Kriterienkatalog automatisiert werden kann. Um einen Marktüberblick zu bieten werden verschiedene Web Vulnerability Scanner von kommerziellen Anbietern in Hinblick auf ihren Funktionsumfang, ihre Preisgestaltung und ihrer Bedienung verglichen.
Der letzte Schritt dieser Diplomarbeit behandelt im Gegensatz zu den vorhergehenden Kapiteln weniger die theoretische Ebene. Er bietet stattdessen Anhand von Praxisszenarien dem Leser die Möglichkeit die Auswirkungen von Schwachstellen und Attacken auf reale Anwendungen nach zu vollziehen.

Danksagung

Mein Dank gilt an erster Stelle der Firma Trusted Shops GmbH, die es mir ermöglicht hat, die vorliegende Arbeit in dieser praxisnahen Form durchzuführen. Persönlich möchte ich Herrn Jean-Marc Noel und Herrn Hendrik Lennarz für die persönliche und gute Betreuung danken. Sie waren jederzeit höchst kompetente Ansprechpartner und standen mir mit Rat bei der Entwicklung dieser Diplomarbeit zur Seite.

Des Weiteren bedanke ich mich bei Herrn Prof. Dr. Rüdiger Grimm für die gute Zusammenarbeit und die für mich so interessante Aufgabenstellung. Außerdem bin ich für die Betreuung meiner Diplomarbeit von Seite der Universität durch Frau Anastasia Meletiadou mit zahlreichen wissenschaftlichen Ratschlägen sehr dankbar.

Vor allem möchte ich an dieser Stelle von ganzem Herzen meinen Eltern für ihre verständnisvolle Unterstützung und die mir eröffneten Möglichkeiten danken.

Abschließend möchte ich noch allen Kommilitonen, Freunden und insbesondere meinen Korrekturleserinnen bedanken, die mir bei der Erstellung dieser Diplomarbeit in vielerlei Hinsicht beigestanden haben.

Inhaltsverzeichnis

1. Einleitung 1
1.1 Ausgangslage und Motivation ... 2
1.2 Problemstellung und erste Erkenntnisse ... 3

2. Analyse von E-Commerce-Webseiten 7
2.1 Technische Bausteine einer E-Commerce-Webseite ... 8
2.1.1 Internet-Anbindung (Inhouse-, Webhosting) ... 9
2.1.2 Durchsatz und Kapazität des Internet-Auftritts ... 9
2.1.3 Aktive Inhalte ... 10
2.1.4 Dynamische Webseiten ... 11
2.1.5 Datenbankensysteme ... 12
2.1.6 Zahlungsverkehrsplattform ... 13
2.2 Technische Schwachstellen ... 14
2.2.1 Programmierfehler ... 15
2.2.2 Konfigurationsfehler ... 35
2.2.3 Konzeptionsfehler (Konstruktionsfehler) ... 38
2.2.4 Fehler resultierend aus menschlichem Verhalten ... 51
2.3 Risiko-Potentiale im E-Commerce-Handel ... 55
2.3.1 Kundensicht ... 55
2.3.2 Anbietersicht ... 57
2.3.3 Providersicht ... 60

3. Diskussion bestehender Kriterienkataloge 61
3.1 Orange Book (TCSEC) ... 62
3.2 White Book (ITSEC) ... 62
3.3 Common Criteria ... 63
3.4 IT-Infrastructure-Library (ITIL) ... 66
3.5 BSI IT-Grundschutz-Kataloge ... 67
3.5.1 BSI Penetrationstest ... 70

4. Ausbildung von Kriterien 75
4.1 Vorbedingungen für Kriterien ... 75
4.2 Strukturierung der Kriterien ... 79
4.2.1 Protokollebene ... 80
4.2.2 Systemkonfigurationsebene (Dienstebene) ... 84
4.2.3 Anwendungsebene ... 89
4.2.4 Qualitätsmerkmale ... 95
4.3 Gewichtung der Kriterien ... 97
4.3.1 Automatisierbarkeit ... 97
4.3.2 Schadenspotential ... 99
4.3.3 Verbreitung nach OWASP ... 99

5. Automatisierung des Kriterienkataloges 103
5.1 Web-Application Scanner ... 104
5.1.1 BSI (BOSS-Nessus) ... 107
5.1.2 Nikto (Whisker) ... 109
5.1.3 Acunetix (WVS) ... 110
5.1.4 SPI Dynamics (WebInspect) ... 113
5.1.5 eEye (Retina) ... 114
5.1.6 GFI LANguard (N.S.S.) ... 116
5.1.7 Watchfire (AppScan) ... 118
5.1.8 N-Stalker (N-Stealth) ... 120
5.1.9 Microsoft (MBSA) ... 122

6. Praxisszenarien 125
6.1 XSS-Attacken ... 125
6.1.1 Session Hacking / Identitätsdiebstahl ... 126
6.1.2 Defacement ... 128
6.2 SQL-Attacken ... 131
6.3 Schadhafte Dateiausführungen ... 132
6.4 Unsichere interne Objektreferenzen ... 133
6.5 Cross Site Request Forgery (CSRF) ... 134
6.6 Informationslecks & Fehlermeldungen ... 135

7. Zusammenfassung 137
7.1 Ausblick und Fazit ... 141

Literaturverzeichnis 143

Abbildungsverzeichnis

Abbildung 1: IT-Sicherheit in Unternehmen [IW] 1
Abbildung 2: Absicherung der Datenübertragung 3
Abbildung 3: Spoofing (Beispiel) 4
Abbildung 4: "Anti-Spoofing" Hinweis 5
Abbildung 5: Kommunikationsweg einer Webanwendung [AC] 7
Abbildung 6: Aufbau eines Datenbanksystems 12
Abbildung 7: Beständiges ↔ Unbeständiges XSS [RM] 19
Abbildung 8: XSS – Accountdiebstahl [RM] 20
Abbildung 9: Schadhafter SQL-Code (Beispiel) 26
Abbildung 10: Klassifikation von Malware [EMS] 29
Abbildung 11: IPv4 Header [PL] 43
Abbildung 12: DNS-Spoofing (Beispiel) [SR] 44
Abbildung 13: ARP-Spoofing (Beispiel) 46
Abbildung 14: Drei Wege Handshake unter TCP 49
Abbildung 15: TCP Header [PL] 49
Abbildung 16: KES-Sicherheitsstudie 2004 [KES] 51
Abbildung 17: Angriffe auf Web-Server [KES] 57
Abbildung 18: Kriterien zur Risikobewertung [KES] 58
Abbildung 19: Bekanntheitsgrad von Kriterienkatalogen [KES] 61
Abbildung 20: Aufbau IT-Grundschutz-Katalog 68
Abbildung 21: Aufbauvarianten eines Penetrationstests [BSI_PEN] 71
Abbildung 22: Beratungs-Dienstleistungen [KES] 73
Abbildung 23: Ebenenmodell 79
Abbildung 24: Ebenen-Kommunikation 84
Abbildung 25: Aufbau der Anwendungsebene [BSI_A] 89
Abbildung 26: Gewichtung der Kriterien 101
Abbildung 27: BSI-BOSS (Nessus) Stufe 0 bis 2 [BOSS_B] 107
Abbildung 28: BSI-BOSS (Nessus) Stufe 3 [BOSS_B] 108
Abbildung 29: Acunetix Web Vulnerability Sanner [AC] 111
Abbildung 30: Retina Sicherheits-Audit Zyklus [EYE] 115
Abbildung 31: GFI LANguard N.S.S. [GFI] 117
Abbildung 32: AppScan Audit [WFS] 119
Abbildung 33: Microsoft Baseline Security Analyzer – Screenshot 123
Abbildung 34: Microsoft Baseline Security Analyzer – Prüfbericht 123
Abbildung 35: XSS - Session Hacking [ATA] 126
Abbildung 36: XSS - Attacke auf ein Banking-Portal [RC] 128
Abbildung 37: XSS – Defacement [ATA] 129
Abbildung 38: Netscape Defacement [FSN] 130
Abbildung 39: E-Plus Webshop XSS-Defacement [SJ_PHP] 130
Abbildung 40: Cross-Site-Scripting ↔ SQL-Injection [RM] 131

Tabellenverzeichnis

Tabelle 1: Sonderzeichen – Code 17
Tabelle 2: Beispiel Cross-Site-Scripting 18
Tabelle 3: TCSEC Klassen 62
Tabelle 4: ITSEC Klassen 63
Tabelle 5: ITSEC Implementierungsklassen 63
Tabelle 6: Common Criteria EAL-Stufen 65
Tabelle 7: Common Criteria Klassen 65
Tabelle 8: ITIL Klassen 67
Tabelle 9: Gewichtung der Kriterien 100
Tabelle 10: Accunetix Preistabelle 112
Tabelle 11: WebInspect Preistabelle 114
Tabelle 12: Retina Preistabelle 116
Tabelle 13: GFI LANGuard Preistabelle 118
Tabelle 14: AppScan Preistabelle 120
Tabelle 15: N-Stalker Preistabelle 122

Codebeispiele

Codebeispiel 1: Http-Banner 121
Codebeispiel 2: Link für einen Cookiediebstahl 126
Codebeispiel 3: Präparierter Link für ein XSS-Defacement 129
Codebeispiel 4: Schwachstelle in der Datei download.php 133
Codebeispiel 5: Präparierte Flashdatei 134

1. Einleitung

Die Aufgabenstellung dieser Arbeit ist die Erstellung eines Kriterienkataloges, der helfen soll die technische Sicherheit von E-Commerce Webauftritten zu gewährleisten. Der Schwerpunkt bei der Auswahl der Kriterien liegt auf deren automatisierter Überprüfbarkeit. Unter einem E-Commerce Webauftritt ist die Internet-Plattform einer Firma zu verstehen, die Güter und/oder Dienstleistungen über das Internet anbietet und zu diesem Zwecke über eine Webseite verfügt. Um eine verlässliche und reproduzierbare Überprüfung der Sicherheit zu ermöglichen, wird in dieser Arbeit ein Kriterienkatalog erstellt, bzw. ein bestehender Kriterienkatalog erweitert.

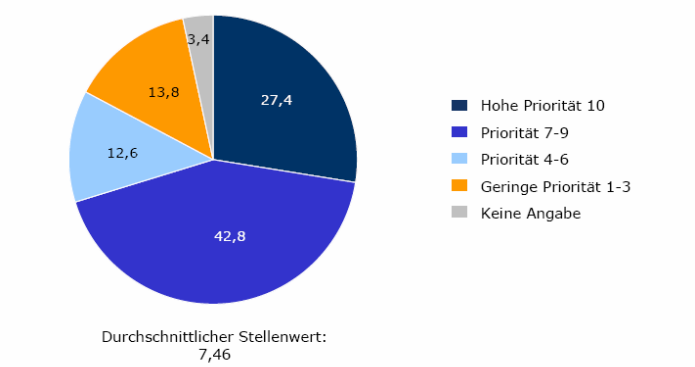

Abbildung 1: IT-Sicherheit in Unternehmen [IW]

Anhand der Abbildung 1 kann gezeigt werden, dass von der Seite der E-Commerce-Unternehmen ein Bedarf nach einem derartigen Kriterienkatalog besteht. Sie zeigt den Stellenwert der IT-Sicherheit in deutschen und schweizerischen Unternehmen im Jahr 2006. Für den überwiegenden Anteil der Befragten (ca. 82%) spielt die IT-Sicherheit in ihren Unternehmen eine hohe bis mittlere Priorität. Lediglich 13% der befragten Unternehmen gaben an, dass sie der IT-Sicherheit nur eine geringe Priorität widmen. Anzumerken ist hierbei, dass es sich bei den befragten Unternehmen um solche handelt, die entweder einen großen Anteil ihres Umsatzes über das Internet erwirtschaften oder in anderer Hinsicht auf das Internet angewiesen sind. Diese Abhängigkeit erklärt das große Interesse der E-Commerce Unternehmen an der technischen Sicherheit ihrer Systeme, Anwendungen und Netzwerke. Der bestehende Bedarf zeigt, dass es eine grundsätzliche Motivation bei E-Commerce Unternehmen gibt, ihr System anhand eines Kriterienkataloges auf Schwachstellen hin überprüfen zu lassen. Da nur die wenigsten Unternehmen eine Überprüfung ihres Systems in eigener Regie durchführen werden können, sind sie hierfür auf Unterstützung von dritter Seite angewiesen. Diese Unterstützung bietet ihnen zum Beispiel ein Dienstanbieter wie Trusted Shops.

1.1 Ausgangslage und Motivation

Die Motivation zur Erstellung eines Kriterienkataloges kann anhand des folgenden Szenarios gezeigt werden:

Für eine Firma, die ihre Dienstleistungen über das Internet anbietet, ist die technische Sicherheit ihres Webauftrittes in zweierlei Hinsicht wichtig:

- Eine Firma kann nur über einen technisch einwandfrei abgesicherten Webauftritt das Vertrauen ihrer Kunden gewinnen. Gerade im anonymen Geschäftsbereich des Internets ist es wichtig, dass Kunden sowohl Vertrauen zu dem Webauftritt, als auch der eigentlich dahinter stehenden Firma, gewinnen. Kunden werden eher dort einkaufen, wo sie sich sicher fühlen. Firmen, die ihren Kunden eine angemessene Sicherheit garantieren können, erlangen infolgedessen einen Wettbewerbsvorteil ihren Konkurrenten gegenüber. Zusätzlich steigt das Image der Firmen.

- Böswillige Angriffe können einen direkten Schaden an der IT-Ausstattung einer Firma verursachen. Das kann in Form von Viren, Trojanern und ähnlichem, schadhaftem Code geschehen, der den Datenbestand und die Funktionsfähigkeit der Firma gefährdet. Hier liegt die Motivation für die Firma klar darin, ihre technische Ausstattung und deren Funktionalität zu schützen. Mit einer technischen Absicherung wird das generelle Risiko von Geschäftsausfällen für die Firma minimiert und ein möglicher Schaden wird begrenzt. Durch die gewonnene Sicherheit in der IT-Ausstattung werden des Weiteren die betrieblichen Prozesse indirekt beschleunigt, da es zu weniger Ausfällen im Unternehmensablauf kommt.

Eine Firma ist folglich im Geschäftsleben, sowohl aus eigener interner Sicht als auch der externen Sicht der Kunden darauf angewiesen, dass ihre IT-Ausstattung und ihr Webauftritt zuverlässig funktioniert. Das bedeutet, dass der Webauftritt stets für den Kunden erreichbar ist, funktional und allgemein den Sicherheitsstandards entsprechen sollte. Im Umkehrschluss wird eine Firma die ihren Webauftritt nicht absichert, mit der Zeit das Vertrauen ihrer Kunden verlieren oder dieses gar nicht erst gewinnen.

Nach außen präsentieren Firmen die Umsetzung ihrer technischen Sicherheit durch ein Zertifikat. Das Zertifikat wird in Form eines Buttons, den die Firmen in ihre Internetseiten integrieren können, dargestellt. Zertifizierungsstellen können kommerzielle Firmen wie Trusted Shops, die Telekom aber auch international agierende Firmen wie VeriSign sein. Daneben existieren staatliche Behörden wie das Bundesamt für Sicherheit in der

Informationstechnik (BSI). Sie alle überprüfen die Sicherheit einer Internetseite und geben das Resultat einer Überprüfung durch ein aussagekräftiges Zertifikat wieder. Das von Trusted Shops vergebene Zertifikat ist bislang nur auf die Überprüfung von rechtlichen Kriterien eines Webauftrittes hin ausgerichtet. Durch diese Arbeit soll der Ansatz gegeben werden, zukünftig auch technische Kriterien zu überprüfen. Das Zertifikat für die technischen Kriterien wird durch einen neuen Button dargestellt. Dies dient dem Zweck, dass ein Kunde zwischen dem technischen und rechtlichen Zertifikat unterscheiden kann.

1.2 Problemstellung und erste Erkenntnisse

Zuerst soll hier festgehalten werden, dass der zu erarbeitende Kriterienkatalog aus einer Liste von Schwachstellen bestehen wird. Jedes Kriterium wird in Form einer Schwachstelle oder einer Klasse von Schwachstellen dargestellt. Dies lässt sich durch ein einfache Beispiel beschreiben. Eine Webseite ist dann technisch sicher, wenn sie keine Schwachstellen besitzt, über die Angriffe durchgeführt werden können. Prüft der Kriterienkatalog mit einer Liste von bekannten Schwachstellen, können diese Schwachstellen nach einer Überprüfung geschlossen werden. Die Sicherheit der überprüften Webseite steigt folglich.
Ein Beispiel für eine technische Schwachstelle ist die unverschlüsselte Übertragung von schutzbedürftigen Eingaben der Kunden. Generell schützenswert sind persönliche Daten, die ein Benutzer zur Authentifizierung eingibt. In Abbildung Zwei wird eine Internetseite abgebildet, die den Benutzer auffordert seinen "Mitgliedsnamen" und das zugehörige "Passwort" zur Authentifizierung einzugeben. Diese schützenswerten Daten sollten immer durch das "https" Protokoll verschlüsselt werden, das in jedem Webbrowser integriert ist.

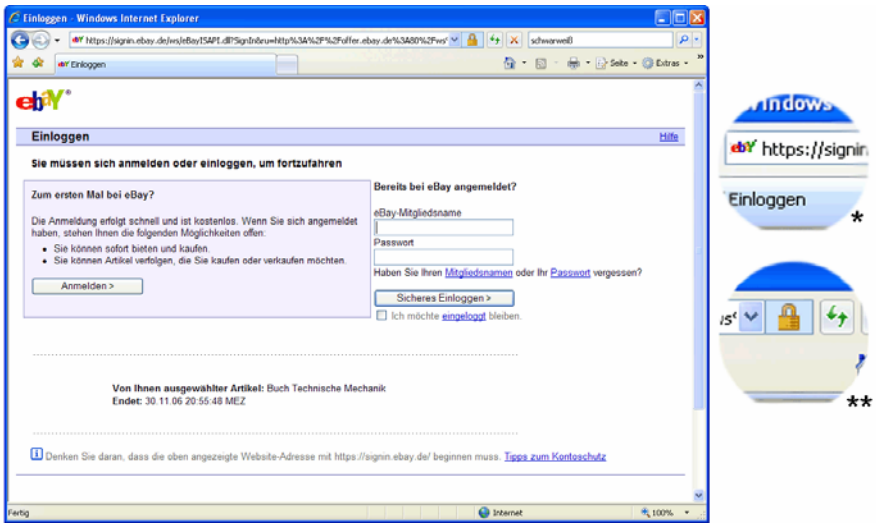

Abbildung 2: Absicherung der Datenübertragung

Für den Benutzer wird die gesicherte Datenübertragung durch zwei Merkmale erkenntlich gemacht: Zum einen werden sichere Internetseiten mittels "https:// " aufgerufen (Abbildung 2,*****), zum anderen wird zusätzlich rechts neben der Adressleiste ein geschlossenes Vorhängeschloss abgebildet (Abbildung 2,******).

Auf den ersten Blick scheint es ausreichend, wenn der Besucher der Internetseite auf die beiden oben angeführten Punkte, "https" und das "Schlosssymbol" achtet. Aber es gibt eine Möglichkeit, "spoofing" genannt, mit der die Originalseite täuschend echt nachgemacht werden kann. Ein Angreifer, der an persönliche Daten wie Passwörter und Kennungen gelangen will, kann diese "Fake-Seite" für seine Zwecke benutzen. Der Besucher einer Internetseite kann "spoofing", wenn überhaupt, nur durch einen Blick auf die Adressleiste erkennen.

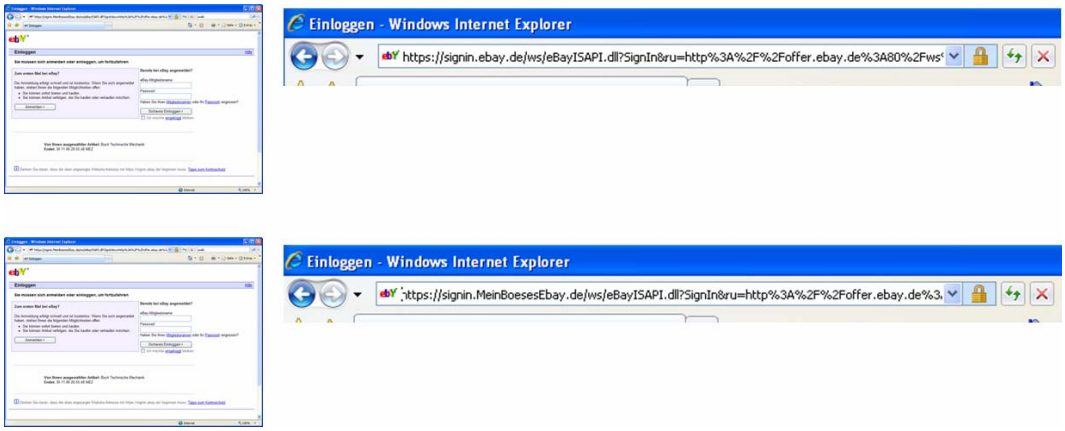

Abbildung 3: Spoofing (Beispiel)

In Abbildung 3 ist ein Beispiel für eine Spoofing-Attacke abgebildet. Die obere Internetseite unterscheidet sich in ihrem äußeren Erscheinungsbild nicht von ihrem unteren Gegenstück. Der einzige Unterschied liegt in der verwendeten Uniform Resource Locator (URL). Die Obere Webseite ist die Originalseite, die untere Webseite dient der Entlockung der Authentifizierungs-Daten eines Kunden.
Um den Betrug erkennen zu können, muss dem Besucher die korrekte Webadresse der Seite bekannt sein. Auf der Beispielswebseite nutzt der Anbieter einen an den Kunden gerichteten Vermerk, der die Erkennung der Authentizität der Seite erleichtern soll (https://signin.ebay.de/ - siehe Abbildung 4).
Beide Schwachstellen, zum einen die Verschlüsselung der Daten und zum anderen das Erkennen des Spoofings, sind daher in den zu schaffenden Kriterienkatalog aufzunehmen.

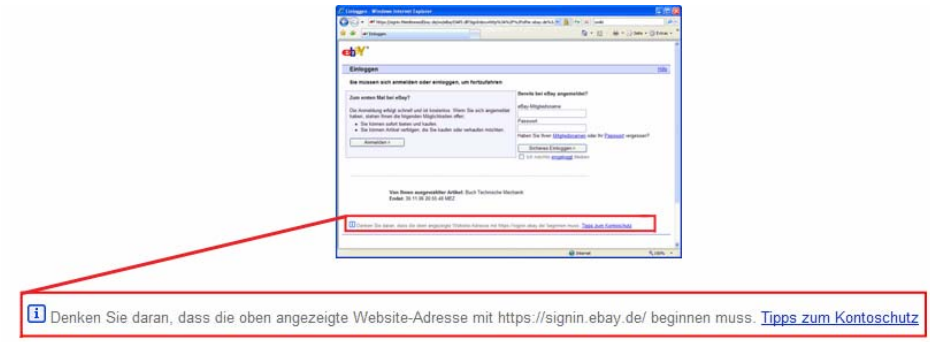

Abbildung 4: "Anti-Spoofing" Hinweis

Die Problematik bei der Erstellung eines Kriterienkataloges ist nun folgende:

- Welche Werte und Kriterien müssen bei E-Commerce-Webseiten berücksichtigt werden, damit eine effektive technische Sicherheit erreicht werden kann?

Im Anschluss an die Auswahl der Kriterien stellt sich die Frage, mit welchem Tool diese überprüft werden können. Hierfür kann entweder auf frei verfügbare Scanner oder kommerzielle Scanner zurückgegriffen werden, die für die eigenen Zwecke erweitert werden. Denkbar ist aber auch eine komplett neue Entwicklung eines Tools, dass die Überprüfung übernimmt. Bisher gibt es frei verfügbare Anätze für Kriterienkataloge von der Bundesagentur für Sicherheit im Internet (BSI-BOSS) oder das Nikto Projekt. Daneben existieren kommerziellen Lösungen von Firmen wie Acunetix oder SPI Dynamics. Die frei verfügbaren Lösungen haben den Nachteil, das sie oft nicht ausreichend in ihrem Funktionsumfang, oder noch nicht auf eine automatisierte Abarbeitung hin ausgerichtet sind. Die kommerziellen Lösungen sind sehr aufwendig in der Bedienung und liegen bei den Kosten für die Überprüfung einer einzelnen Webseite bei mindestens 500 €. Folglich richten sich die kommerziellen Lösungen momentan primär an große Firmen die sich eine solche Untersuchung finanziell leisten können. Die hohen Kosten, der große Aufwand verbunden mit einer Überprüfung und die unzureichende Funktionalität zeigen, dass gerade im großen Markt der kleinen E-Commerce-Webseiten, ein Bedarf nach einem einfach zu handhabenden und kostengünstigen Produkt besteht.

Das anschließende Kapitel befasst sich nach eine Analyse des Aufbaues einer E-Commerce-Webseite mit ersten Schwachstellen und Angriffspunkten einer Webseite. Aus diesen wird ein vorläufiger Kriterienkatalog gebildet, der im Laufe der Arbeit weiter verfeinert wird. Das Augenmerk liegt dabei insbesondere auf gewerblichen Internetauftritten da hier, im Gegensatz zum privaten Gebrauch, ein höheres Schadenspotential vorhanden ist. Bei der Auswahl der Kriterien wird der Schwerpunkt auf die Automatisierbarkeit, dem Schadenspotential und der Verbreitung der Schwachstellen gesetzt.

Die Aufgabestellung dieser Arbeit setzt sich folglich aus einer geeigneten Auswahl von Schwachstellen-Kriterien und deren späteren Automatisierbarkeit zusammen. Bei der Auswahl der Kriterien muss beachtet werden, dass sich der Schwerpunkt der Hacker-Angriffe in der letzten Zeit verlagert hat. Der Trend geht weg von den typischen und länger bekannten Angriffen gegen Systeme im Netz, zum Beispiel Buffer-Overflow-Attacken, hin zu Angriffen gegen Webanwendungen[LL].

2. Analyse von E-Commerce-Webseiten

E-Commerce-Unternehmen nutzen heutzutage die vielseitigen Möglichkeiten des World Wide Web zum kostengünstigen Verteilen und Austauschen von Informationen mit Kunden oder für Geschäftsvorgänge unterschiedlichster Art. Hierfür greifen sie auf Webanwendungen zurück, die ihre Webseiten mit Funktionen erweitern. Webanwendungen sind Computerprogramme, die Webseitenbesuchern die Eingabe von Daten in oder die Ausgabe von Daten aus einer Datenbank über das Internet mit Hilfe eines Webbrowsers ermöglichen. Die angeforderten Daten werden an den Besucher der Webseite von einem Webserver versendet. Beim Besucher der Webseite werden diese von der Webanwendung als aufbereitete dynamische Informationen im Browser angezeigt. Aktuell richten sich viel Angriffe auf diese Webanwendungen, da in ihnen die meisten kritischen Schwachstellen liegen.

Login-Seiten, Formulare für Support- und Produktanfragen, Einkaufswagen und sonstige bereitgestellte dynamische Inhalte sind mittlerweile elementare Bestandteile von Webseiten. Sie sorgen für eine reibungslose Kommunikation zwischen Unternehmen und ihren Kunden und sind allesamt Beispiele für Webanwendungen, die bereits als Serienprodukt erworben oder firmenspezifisch entwickelt wurden [AC].

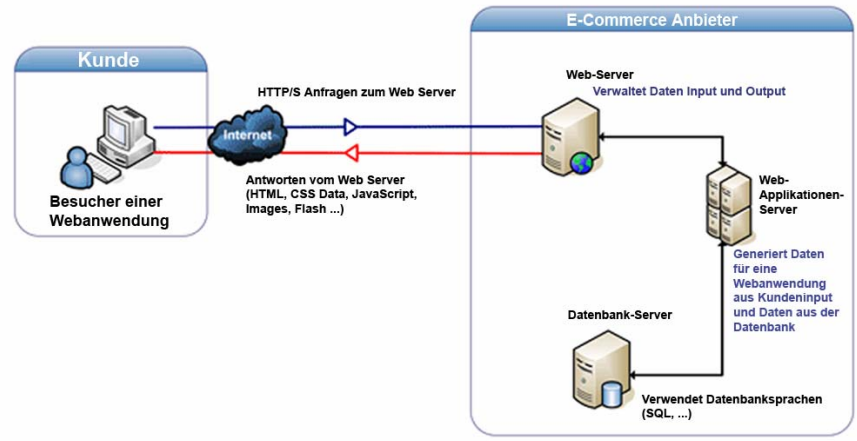

Abbildung 5: Kommunikationsweg einer Webanwendung [AC]

Zu Beginn der Analyse von E-Commerce-Webseiten stellt sich die Frage, welche "Werte" bei einem Webauftritt einer Firma schützenswert sind. Gesichert werden muss der Umsatz der Firma. Mit dem Umsatz verbunden sind zum einen der wertschöpfende Prozess und zum anderen die generellen Vermögenswerte der Firma. Der wertschöpfende Prozess liegt bei

einer Firma, die im E-Commerce-Bereich agiert, vornehmlich in ihrem Webauftritt. Dieser setzt sich zusammen aus der eigentlichen Webseite und dem zugehörigem technischen Grundgerüst. Das technische Grundgerüst eines Webauftrittes wird in kompakter Form im folgenden Unterkapitel aufgebaut. Aus diesem Grundgerüst können anschließend die einzelnen Bedrohungsszenarien auf die jeweiligen Bausteine des Webauftrittes gemappt werden.

2.1 Technische Bausteine einer E-Commerce-Webseite

Zur Realisierung eines E-Commerce-Webauftrittes sind eine Reihe von Bausteinen erforderlich. Das können rechtliche, organisatorische und technische Bausteine sein. Dieses Kapitel befasst sich nur mit den technischen Kriterien. Wird von einer Firma ein Webauftritt geplant, so sollte sie die folgenden Fragen beachten:

- Wo wird der Webauftritt gehostet?
- Welche Transaktionsraten und Datendurchsätze sind zu erwarten? Wie ist aufgrund dieser Anforderungen die Internetverbindung und der Webserver zu dimensionieren?
- Welchen Inhalt soll die Webpräsenz haben, sind zum Beispiel aktive Inhalte erforderlich?
- Erfordert ein flexibler Zugriff oder eine reversionssicher Datenhaltung ein Datenbanksystem?
- Wird ein Datenbanksystem benötigt?
- Soll eine Online-Bezahlplattform verwendet werden?

Nur wenn die aus den Fragen resultierenden Punkte konsequent beachtet werden kann eine Webpräsenz technisch und auch sicher funktionieren. Eine Vorraussetzung ist jedoch, dass diese Anforderungen einer Firma schon aktiv während der Planung und des Designs berücksichtigt werden. Die im folgenden Kapitel aufgeführten technischen Schwachstellen, verursacht durch Programmierfehler, Konfigurationsfehler, Konzeptionsfehler oder Fehler resultierend aus dem menschlichen Handeln, erlauben einer Firma nur eine spätere passive Abwehr.

2.1.1 Internet-Anbindung (Inhouse-, Webhosting)

Beginnt eine Firma ihre Webpräsenz aufzubauen, muss sie sich Gedanken darüber machen, welche "hosting"-Variante sie anwenden will. Der Begriff "hosting" bezeichnet, wo und auf welche Weise ihre Webseite mit dem Internet verbunden wird. Hierbei wird zwischen Inhouse- und Webhosting unterschieden. Bei der Inhouse-Variante werden die Daten der Internetseite auf Servern in der Firma abgelegt. Das bedeutet, dass die Daten räumlich in der eigenen Firma liegen und dort eigenständig von dieser verwaltet werden. Diese Variante ermöglicht, dass unmittelbar in der Firma auf die Daten zugegriffen werden kann. Die Kontrolle über die Daten bleibt der Firma voll erhalten. Weiterhin kann durch die räumliche Nähe schnell und unkompliziert auf die Datenbestände zugegriffen werden.

Unter Webhosting wird die Auslagerung einer Webpräsenz zu einem externen Dienstleister verstanden. Bei dieser Lösung kann die Firma vor allem davon profitieren, dass Know-how, Infrastruktur und erforderliche Sicherheitstechniken beim Dienstleister gebündelt zur Verfügung gestellt werden. Die Firma verliert aber zu einem gewissen Maße die direkte Kontrolle über ihre Daten und den Webauftritt. Problematisch ist bei Webhosting die Übertragung der Daten von der Firma zum Dienstleister. So kann es z.B. bei einer Aktualisierung des Datenbestandes zu zeitlichen Verzögerungen kommen. Des Weiteren ist jede Datenübertragung ein möglicher Gefährdungspunkt, bezüglich der Sicherheit. Über eine sichere online Verbindung können diese Schwachstellen aber weitestgehend beseitigt werden[DFW].

2.1.2 Durchsatz und Kapazität des Internet-Auftritts

Erwirtschaftet eine Firma ihren Umsatz vornehmlich über ihren Webauftritt, so muss sie dafür Sorge tragen, dass dieser Dienst für ihre Kunden stets verfügbar ist. Die Verfügbarkeit des Webauftrittes ist nicht nur durch vorsätzliche Angriffe und technische Ausfälle, sondern auch durch die normale Nutzung der Kunden gefährdet. Um den Dienst dem Kunden zur Verfügung zu stellen, benötigt die Firma eine Anbindung an das Internet, welche die Daten zum Kunden überträgt und einen Webserver auf dem die Daten abgelegt und verwaltet werden. Jede Internetverbindung verfügt über eine maximale Kapazität an zeitgleich übertragbaren Daten. Ebenfalls begrenzt ist der Datendurchsatz, den ein Webserver in einem Zeitintervall abarbeiten kann. Durch diese beiden Faktoren kann es bei intensiver Nutzung der Webpräsenz zu längeren Ladezeiten für den Benutzer kommen. Das langsame

Antwortverhalten des Servers kann schnell für einen Teil der Benutzer oder sogar alle Benutzer unakzeptabel werden. Der Dienst steht dann praktisch nicht mehr zur Verfügung. Solche Beschränkungen können auch für vorsätzliche Angriffe ausgenutzt werden. Eine angemessene Dimensionierung des Durchsatzes der Internetverbindung und der Kapazität des Webservers ist daher unentbehrlich für eine verlässliche Funktionalität des Webauftrittes.

Ist die Webpräsenz im Einsatz, muss die Last, die auf der Internetverbindung und den Webservern anliegt, dauerhaft überwacht werden. Oft ist die Auslastung nicht linear über den Tag gleich bleibend verteilt. Vielmehr treten Spitzenzeiten auf, in denen die Last auf die Netzanbindung und den Server stark zunimmt. Tritt eine solche Spitzenlast auf, kann mit der Zuschaltung weiterer Kapazitäten reagiert und die Situation entschärft werden. Die Kapazität der Internetverbindung, das heißt, die Gesamtbandbreite, kann mittels zweier Ansätze erhöht werden. Der erste Ansatz besteht darin, auf eine neue Anschlussart mit einer höheren Datenrate auszuweichen. Als alternativer Ansatz bietet sich an, dass eine zusätzliche Leitung (Backupleitung) zugeschaltet wird, um die Hauptleitung zu entlasten. Entsprechend wird bei einem Webserver vorgegangen. Entweder ersetzt man diesen durch eine leistungsfähigere Variante oder ein zweiter Server wird parallel geschaltet. Beide Lösungen haben jeweils Vor- und Nachteile. Ein paralleler Betrieb von zwei Servern oder Netzverbindungen bietet gegenüber der einzelnen Variante eine höhere Ausfallsicherheit, benötigt aber auch mehr administrative Pflege [DFW].

Das BSI bietet mit den IT-Grundschutz-Katalogen einen Leitfaden an, der einem E-Commerce-Anbieter bei der Auswahl und der Einrichtung eines Webservers, behilflich ist. Der Baustein *B5.4 Webserver* befasst sich mit diesem Thema[GSK_B1].

2.1.3 Aktive Inhalte

In den beiden vorhergehenden Unterkapiteln wurden die grundsätzlichen Anforderungen an die Hardwareausstattung eines Webauftrittes vorgestellt. Dieses Unterkapitel geht nun auf die geeignete Wahl der technischen Realisierung des Inhaltes eines Webauftrittes ein. Hierzu gehören neben statischen Informationen auch aktive Inhalte. Unter aktiven Inhalten sind Computerprogramme zu verstehen, die in Webseiten enthalten sind oder beim Betrachten einer Webseite automatisiert nachgeladen werden. Sie bieten dem Benutzer in der Regel einen Mehrwertdienst. Je nach Betätigungsgebiet einer E-Commerce-Firma kann diese in ihrem Webauftritt nur schwer auf aktive Inhalte verzichten. Die Gefährdung bei aktiven Inhalten liegt in ihrer unbemerkten Agierung im Hintergrund der Seite. Sie werden

vom Benutzer in der Regel nicht wahrgenommen. Ausgeführt werden diese Programme auf dem Computer des Internetnutzers. Entweder vom jeweils verwendeten Webbrowser, oder von dem darunter liegenden Betriebssystem. Grundsätzlich sollte zwischen aktiven Inhalten und dynamischen Webseiten unterschieden werden. Während aktive Inhalte Risiken für den Privatanwender bedeuten, bringen dynamische Webseiten Gefährdungen für den Betreiber mit sich. Dynamische Webseiten werden im nächsten Unterkapitel näher erläutert. Die wichtigsten Beispiele für aktive Inhalte, die in Webseiten häufig verwendet werden, sind die Technologien JavaScript, VBScript, Java und ActiveX.

2.1.4 Dynamische Webseiten

Die Technik der dynamischen Webseiten ist eine Weiterentwicklung die auf den statischen Webseiten aufbaut. Statische Webseiten entsprechen einfachen Dateien, die unveränderlich auf einem Webserver liegen. Dynamische Webseiten werden hingegen im Moment der Anforderung erzeugt. Das ist sinnvoll, wenn eine Webseite sehr aktuelle Informationen wie z.B. Börsenkurse oder das Wetter von Morgen enthalten soll. In diesem Fall führt der Webserver ein Programm aus, das vorher spezifisch für diese Aufgabe entwickelt wurde. Es trägt die Daten z.B. aus Datenbanken zusammen und erstellt das HTML-Dokument. Dieses wird dann vom Webserver an den Browser übertragen und beim Benutzer angezeigt. Dynamische Webseiten arbeiten in der Regel mit Datenbanken, zum Beispiel SQL oder PHP Datenbanken, zusammen.
Der Unterschied zu aktiven Inhalten, bekannt aus dem vorhergehenden Kapitel, liegt in der räumlichen Ausführung des generierten Codes. Die dynamische neue Webseite wird auf dem Webserver der Firma generiert. Dies geschieht aus den Eingabedaten, des Benutzers und den Daten des Webservers der Firma. Ist einem Benutzer eine Schwachstelle im System bekannt kann er böswillig Eingabedaten oder Codezeilen einschleusen. Hierdurch kann der Webserver oder der Datenbestand der Firma in Mitleidenschaft gezogen werden. Das Gefährdungspotential verschiebt sich folglich von der Benutzerseite bei aktiven Inhalten, zum Anbieter von Internetseiten bei der Verwendung von dynamischen Internetseiten [RP].

2.1.5 Datenbankensysteme

Soll in einem E-Commerce-Webauftritt ein größerer Daten- beziehungsweiße Warenbestand verwaltet werden, ist der Einsatz eines Datenbankensystems unumgänglich.
Ein Datenbankensystem ist ein System zur elektronischen Datenverwaltung, das sich aus zwei Elementen zusammensetzt:
Das erste Element besteht aus dem Datenbankmanagementsystem (DBMS). Dies ist eine Verwaltungssoftware, die die Administration der Daten übernimmt. Das zweite Element setzt sich aus der Menge der zu verwaltenden Daten zusammen. Die Menge der Daten kann als die eigentliche Datenbank verstanden werden. Das Datenbankmanagementsystem organisiert intern die strukturierte Speicherung der Daten gemäß eines vorgegebenen Datenbankmodells (z.B. das relationale Datenbankmodell). Zusätzlich kontrolliert es alle lesenden und schreibenden Zugriffe auf die Datenbank. Als Schnittstelle zu anderen Anwendungen benötigt das System eine Datenbanksprache (z.B. SQL). Die Datenbanksprache dient der Formulierung von Abfragen, dem Einfügen und Ändern von Daten und für die Ausführung von administrativen Befehlen. Die Datenbank enthält zusätzlich zu den eigentlichen Daten noch die Beschreibung der Daten, den sogenannten Datenkatalog [KE].

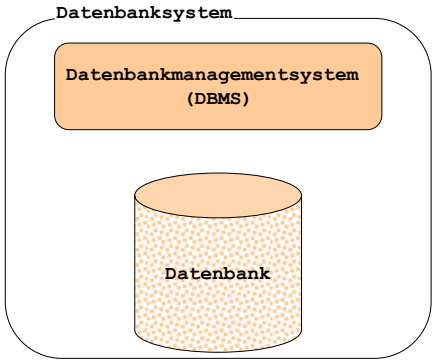

Abbildung 6: Aufbau eines Datenbanksystems

Der Vorteil bei der Verwendung eines Datenbanksystems liegt darin, dass verschiedenste Anwendungen mit der Datenbank kommunizieren können. Dies wird über die Schnittstelle, die das Datenbankmanagementsystem anbietet, ermöglicht. Für den Anbieter einer E-Commerce-Webseite stellt sich die Problematik, aus den verschiedenen Datenbanktechniken, die für seine Bedürfnisse passende, auszuwählen. Die bedeutendsten Datenbankmodelle sind die relationalen und die objektorientierten Datenbanksysteme. Beide Systeme haben für unterschiedliche Anwendungen Vorteile.

Ein relationales Datenbanksystem (RDBS) sollte bevorzugt werden, wenn mit einfachen und formatierten Datenbeständen gearbeitet wird. Sind kurze Antwortzeiten und/oder hohe Transaktionsraten gefordert, sind relationale Datenbanken ebenfalls zu bevorzugen. Sie können dadurch, dass die Daten formatiert und somit leichter indizierbar sind, schnellere Antwortzeiten garantieren. Objektorientierte Datenbanksysteme (ODBS) hingegen sollten bei komplexeren Strukturen eingesetzt werden. Beispiele hierfür sind Datenbanken die geographische Informationssysteme, Netzleitsysteme oder Multimedia-Anwendungen zu Grunde liegen haben.

Das Datenbanksystem einer E-Commerce-Webpräsenz entspricht einer Datenbank die den Warenbestand in katalogisierter Form enthält und gleichzeitig der Präsentation der Waren dient. Dementsprechend ist die Datenbank von dem Betreiber der Webpräsenz einzurichten und zu pflegen. Die Datenbank bildet die Basis für den elektronischen Katalog, der dem Kunden eine Übersicht über die Produktpalette verschafft. Die Beschreibung der Produkte muss so umfassend sein, dass sie eine Kaufentscheidung ermöglicht. Je nach Umfang des Produktkatalogs sollte der Betreiber eine Suchfunktion in seiner Webpräsenz integrieren. Durch sie wird dem Kunden eine leistungsfähige Recherche ermöglicht. Der Aufwand für die Indizierung der angebotenen Produkte nach Schlagwörtern muss allerdings berücksichtigt werden [SD].

2.1.6 Zahlungsverkehrsplattform

Die Zahlungsmodalitäten können als einfachste Lösungen unabhängig von der Webpräsenz, z.B. durch Lieferung ausschließlich gegen Rechnung, realisiert werden. Elektronische Zahlungssysteme wie Lastschriftverfahren, Kreditkartenzahlung und Micropayment bieten dem Kunden aber einen erheblichen Komfortgewinn. Die Kundenfreundlichste Lösung ist aber, wenn von dem Shop-Betreiber, eine Schnittstelle zu einem Serviceprovider für elektronische Zahlungssysteme eingerichtet wird. Diese Lösung ist in der Praxis auch die am weitesten verbreitet. Insbesondere wenn mehrere Zahlungsverfahren an einen Online-Shop angebunden werden sollen, lässt sich der Implementierungsaufwand durch Verwendung einer Zahlungsverkehrsplattform im Vergleich zu einer Eigenlösung deutlich reduzieren. Aufwändige Verfahren wie Adress- oder Bonitätsprüfungen können durch die Zahlungsverkehrsplattform automatisiert abgewickelt werden. Weitere Vorteile ergeben sich durch die einfache Anbindung weiterer Online-Shops und neuer Zahlungsverfahren [BKSW].

2.2 Technische Schwachstellen

In Kapitel 2.1 wurden die wichtigsten Bausteine eines E-Commerce-Webauftrittes vorgestellt. Anhand dieser Bausteine können im Folgenden die einzelnen Gefährdungspunkte einer E-Commerce-Webseite aufgezeigt werden. Gefährdungspunkte bei Webauftritten entstehen, vereinfacht ausgedrückt, immer dort wo Fehlerstellen in der eingesetzten Soft- oder Hardware auftreten. Jeder Gefährdungspunkt kann wieder als eine Schwachstelle verstanden werden. Schwachstellen repräsentieren letzten Endes die Kriterien, die später in den Kriterienkatalog aufgenommen werden. Diese Kriterien müssen bei einer Überprüfung einer Webseite beachtet werden, um eine technisch sichere Webseite ohne Schwachstellen zu garantieren. Denn ohne Schwachstellen (Fehlerstellen) gibt es für einen Angreifer keine Ansatzpunkte für einen Angriff auf eine Webpräsenz. Die Fehler die vorkommen können, werden für eine bessere Handhabung in vier Fehlerklassen aufgeteilt. In den folgenden vier Unterkapiteln werden diese vier Fehlerklassen näher erläutert [BSI_IS]:

- Schwachstellen aufgrund von Programmierfehler:

Die aktuell bekanntesten Programmierfehler eines IT-Systems sind die folgenden[OWASP]:

- ▫ Cross-Site-Reference-Forgery
- ▫ Cross-Site-Scripting
- ▫ Session Riding
- ▫ mutwillige Programmierfehler
- ▫ Buffer Overflow
- ▫ SQL-Injection

- Schwachstellen aufgrund von Konfigurationsfehler:

- ▫ Umgang mit Konfigurationen
- ▫ Homepage Defacement
- ▫ Directory Indexing

- Schwachstellen aufgrund von Konzeptionsfehler:

- ▫ Mailarchitektur
- ▫ CRLF Injection
- ▫ Logikfehler
- ▫ Minimalitätsprinzip von Informationen
- ▫ Spoofing

- Schwachstellen aufgrund von Fehlern verursacht vom Benutzer:

- ▫ Bedienungsfehler
- ▫ Umgang mit Passwörtern
- ▫ Administrative Fehler
- ▫ Leichtgläubigkeit von Benutzern

Es kann vorkommen, dass ein Fehler nicht eindeutig einer Klasse zugeordnet werden kann. In diesem Fall wird ein Vermerk an der betreffenden Textstelle vorgenommen. In den folgenden Unterkapiteln wird auf die einzelnen Fehlerklassen näher eingegangen.

2.2.1 Programmierfehler

Fehler bei der Programmierung der verwendeten Software sind eine häufige Ursache für Sicherheitslücken von Webauftritten. Dies können Fehler der firmeneigenen Software sein, aber auch Fehler in Programmen von Zulieferern. Firmeneigene Software ist Software, die in der eigentlichen Firma selbst produzierte wurde. Das können ganze Systeme aber auch Tools bis hin zu einfachen HTML Seiten sein. Unter Software von Zuliefern versteht man zum Beispiel das Windows System, das extern programmiert und ausgeliefert wird. Fehler können in beiden Arten von Software gleichermaßen auftreten. Eigene Software kann aber im Vorfeld besser durch umfangreiche Testverfahren auf Fehlern hin überprüft werden, da die Firma selbst am Design der Software Anteil nimmt.

Jede bestehende Sicherheitslücke eines Webauftritts bietet einen möglichen Angriffspunkt für böswillige Eindringlinge. Programmierfehler an sich entstehen aufgrund mangelhafter Sorgfalt, Ausbildung oder unzureichender Qualitätssicherung während der Erstellung von Software. Oft ist das Verhindern von Programmierfehlern eine Frage der damit verbundenen Kosten. Eine hundertprozentige Verifikation, d.h. Überprüfung des Codes auf Korrektheit und ein zugehöriger formaler Beweis der Korrektheit des Codes ist sehr teuer und zeitaufwendig. Ein formaler Beweis ist in der Praxis generell nur für kleinere Programmstücke möglich, da der Aufwand proportional zur Codegröße steigt.

Zur Verdeutlichung der Bedeutung von Programmierfehlern wird im Folgenden kurz das Microsoft Windows Betriebssystem betrachtet. Eine aktuelle Windowsversion besteht aus rund 35 Millionen Codezeilen. Wird von einer, dem ersten Anschein nach, geringen Fehlerrate von einem Promille der Codezeilen ausgegangen, bedeutet dies jedoch tatsächlich eine Menge von 35.000 fehlerhaften Codezeilen [OS_XP]. Eine Überprüfung jeder einzelnen Codezeile ist in der Praxis wirtschaftlich nicht zu realisieren. Eine Verifikation des Codes, das heißt eine mathematische Überprüfung auf Korrektheit, wird in der Regel nur für sicherheitsrelevanten Anwendungen, wie im Bank- oder Militär-Bereich angewandt. Fehler in Softwareprogrammen können folglich nur schwer ausgeschlossen werden. Aus diesem Grund ist es unabdingbar, auf neu erkannte Fehler schnellstmöglich zu reagieren. Dies kann entweder durch das Einspielen eines Patches, oder wenn möglich durch Abschalten des gefährdeten Programms geschehen. Eine problematische Situation ist (aktuell) allein durch die große Menge an Programmen und Tools gegeben, die auf einem Computer laufen. Jedes zusätzliche Programm bietet einen möglichen Angriffspunkt. Die Anzahl der möglichen Angriffspunkte kann aber mit geringem Aufwand verringert werden. Gerade im Einsatzbereich eines E-Commerce-Webauftrittes werden längst nicht alle Dienste benötigt, die standardmäßig auf einem Windows oder Linux System laufen. Zur

Gefahrenvorsorge sollte aufgrund dessen nur ein minimales System benutzt werden, auf dem nur die wirklich benötigten Dienste ausgeführt werden.

Viele Programmierfehler werden erst bekannt, wenn die eigentliche Software schon ausgeliefert und längere Zeit in Gebrauch ist. Dieser Umstand macht es erforderlich, dass der Betreiber einer E-Commerce-Webpräsenz einer dauerhaften Systempflege nachgeht. Zu diesem Zweck gibt es verschiedene Quellen im Internet. In sogenannten Schwachstellendatenbanken kann, bzw. sollte sich ein Systemadministrator regelmäßig informieren. Diese bestehen entweder in Form von klassischen Datenbanken, Newsgroups oder auch aus abonnierten Mailinglisten, die verschickt werden, sobald eine Schwachstelle aufgedeckt wird [SH]. Bekannte Anbieter von derartigen Datenbanken sind das Bundesamtes für Sicherheit in der Informationstechnik oder, als internationale Quelle, die landesspezifischen Computer Emergency Response Teams (CERTs).

In dem folgenden Kapitel werden einige Beispiele für Programmierfehler vorgestellt. Das Hauptaugenmerk der ausgewählten Programmierfehlern liegt auf den potentiellen kritischen Auswirkungen in Hinsicht auf E-Commerce-Webseiten. Aus diesen Programmierfehlern entstehen dann im nächsten Schritt die Kriterien für den angestrebten Kriterienkatalog.

Cross-Site-Scripting (XSS)

Die im Jahr 2006 am häufigsten für einen Angriff verwendete Schwachstelle in Webanwendungen bestand laut dem Open Web Application Security Project (OWASP) in der Ausnutzung von Cross-Site-Scripting (XSS). Cross-Site-Scripting-Attacken nutzen Sicherheitslücken in Webanwendungen auf Java- oder HTML-Basis aus. Diese Schwachstellen entstehen in der Regel durch Fehler in der Programmierung von Webanwendungen. Ein XSS-Angriff fügt Code-Informationen aus einem Kontext, in dem der Code als nicht vertrauenswürdig eingestuft würde, in einen anderen Kontext ein, in dem der Code als vertrauenswürdig eingestuft wird. Aus diesem vertrauenswürdigen Kontext kann dann ein Angriff gestartet werden [OWASP].

In der Praxis bedeutet dies z.B., dass durch einen Angreifer, Javacode in die Eingabebox einer dynamischen Webseite eingetragen wird. Wird dieser Code im Anschluss von der Webseite nicht als "einfacher" Text sondern als Javacode interpretiert, d.h. ausgeführt, besteht eine Sicherheitslücke. XSS Attacken können durch eine Filterung der eingehenden Parameter verhindert werden. Hierfür müssen alle eingehenden Parameter nach JavaScript typischen Zeichen wie "<*script*>", "</*script*>", "<"," >", " / " durchsucht werden. Die Lösung, "schädliche" Eingaben zu definieren *(Blacklist)*, um diese herauszufiltern bietet in

der Praxis keine vollständige Sicherheit. Viele HTML-Sonderzeichen können in HTML-Codes "umgeschlüsselt" werden, die dann von den Filterregeln nicht mehr erkannt werden.

HTML-Sonderzeichen	HTML-Code
<	<
&	"
"	&

Tabelle 1: Sonderzeichen – Code

Daher empfiehlt es sich, eine exakte Liste mit "guten" Eingaben *(Whitelist)* zu definieren und nur solche zu erlauben. Auf der folgenden Seite wird ein Beispiel für eine einfach gehaltene Cross-Site-Scripting-Attacke vorgestellt. Sie führt keine Schadensroutine aus, sondern zeigt, dass mit einfachen HTML-Befehlen eine Webseite im Erscheinungsbild verändert werden kann.

Beispiel für Cross-Site-Scripting [SEC]:

Die Schnittstelle, an der Cross-Site-Scripting-Attacken ansetzen, sind Eingabefelder in Webseiten. Die Vorraussetzung dabei ist, dass diese die Eingaben des Benutzers als Code interpretieren. Ein einfaches Beispiel für eine solche Schnittstelle kann mit einer Suchfunktion gezeigt werden.
 	
In ein Suchfeld wird "`München`" als Suchwert eingegeben.
Die Antwort auf die Suchanfrage lautet:
 	
`"Die Suche nach München hat leider keine Treffer geliefert. Bitte versuchen Sie es erneut".`
 	
Die Benutzereingabe, hier München, wird in der Antwort der Webseite wiederholt. Ein einfacher Test zeigt, ob eine XSS-Schwachstelle vorliegt.
Wir geben folgenden Suchstring, in den HTML-Code eingebettet wurde, ein,:
 	
`Münche<i>n`
 	
Das Ergebnis lautet:
 	
`Die Suche nach 'Münche`*`n' hat leider keine Treffer geliefert.`*
 	
Ab diesem Zeitpunkt kann von einer Anfälligkeit für XSS gesprochen werden. In der Antwort wurde statt des eingegebenen <i> der nachfolgende Text *kursiv* dargestellt. Die Eingabe wird nicht als Nutztext angezeigt, sondern als HTML-Code interpretiert (<i> steht für das Umschalten auf Kursivschrift). Ein Blick in den Quellcode gibt weiteren Aufschluss:
 	
`<P>Die Suche nach 'Münch<i>en' hat leider keine Treffer geliefert.`
 	
Der unterstrichene Textteil ist das, was der Anwender eingegeben und die Anwendung (hier das Suchprogramm) unverändert wieder ausgegeben hat. Der Browser stellt dies auch genauso dar, d.h. stellt kursiv, was auf das <i> folgt.
 	
Richtig wäre gewesen, die Zeichen < und > in die Sequenz umzuwandeln, die der Browser auch als 'größer' und 'kleiner' darstellt. Das wären als HTML-Entity die Codes `<` bzw. `>` gewesen. Die Ausgabe hätte dann richtig gelautet:
 	
Im Quellcode:
 	
`<P>Die Suche nach 'Münch<i>en' hat leider keine Treffer geliefert.`
 	
und im Browser:
 	
`Die Suche nach 'Münch<i>en' hat leider keine Treffer geliefert.`
 	
Eine andere Möglichkeit des Umgangs mit derartigen Eingaben wäre das Löschen von Zeichen wie < und > gewesen.

Tabelle 2: Beispiel Cross-Site-Scripting

XSS Attacken existieren in zwei Varianten:

1. Beständiges XSS:

Beständiges XSS bedeutet, dass der schadhafte Code auf einer Webseite abgelegt wird und bei jedem Aufruf der Seite ausgeführt wird. Das können zum Beispiel Foren-, Nachrichten- oder Blog-Webseiten sein. In dieser Variante wird folglich die Verbreitung von dem Server, auf dem die Webseite liegt, übernommen. Das Gefährdungspotential liegt hier höher als bei unbeständigem XSS, da ein Besuch der Webseite ausreicht um den Code auszuführen. Die Gefährdung bei dieser XSS-Variante ist hauptsächlich beim Besucher der Webseite, nicht beim Anbieter, zu sehen. Beständiges XSS kann zum Beispiel auch durch eine SQL-Injektion vorbereitet werden. Mehr hierzu wird unter dem Stichpunkt SQL-Injection in diesem Kapitel erleutert.

2. Unbeständiges XSS:

Unter unbeständigem XSS wird verstanden, dass der Benutzer eine speziell mit schadhaftem Code präparierte Link anklickt. In dem Moment in dem die URL besucht wird, wird der in dem Link enthaltene Code im Browser des Benutzers ausgeführt. Die Schwierigkeit für den Angreifer liegt in dieser Methode darin, dass der Angegriffene aktiv den Link anklicken muss. Dies kann der Angreifer erreichen indem er ihm eine E-Mail, die den präparierten Link enthält, schickt.

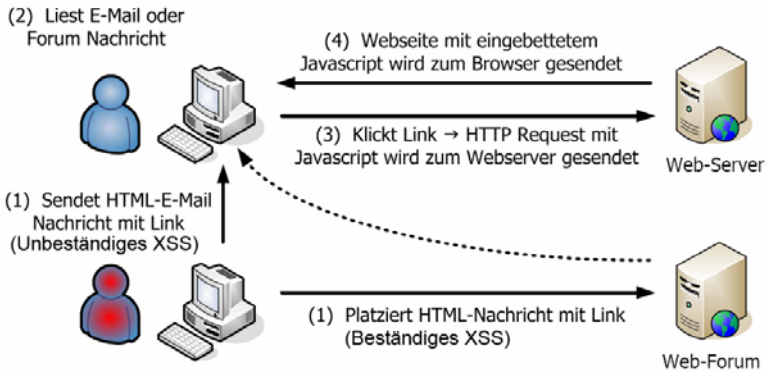

Abbildung 7: Beständiges ↔ Unbeständiges XSS [RM]

Zusätzlich gibt es eine Weitere Unterteilung in der zwischen Client- und Serverseitigen XSS-Attacken unterschieden wird:

Clientseitiges Cross-Site-Scripting schleuste Code auf Clientseite ein, so das dieser von ihm ausgeführt wird. Das bedeutet, dass der Code lokal auf dem angegriffenen Rechner durch den Webbrowser oder das E-Mail-Programm interpretiert und ausgeführt wird. Daher muss ein Angreifer seinem Opfer einen präparierten Hyperlink zukommen lassen. Diesen

Hyperlink kann er zum Beispiel in eine Webseite einbinden oder in einer E-Mail versenden. Es werden häufig URL-Spoofing-Techniken eingesetzt, um den Link unauffällig oder vertrauenswürdig erscheinen zu lassen. Ein klassisches Beispiel für clientseitiges Cross-Site-Scripting ist die Übergabe von Parametern an ein CGI-Skript einer Webseite. So können unter Umständen manipulierte Daten an den Benutzer gesendet werden [BD_A].

Serverseitiges Cross-Site-Scripting versucht, Code auf einem Webserver sowohl abzulegen als auch dort auszuführen. Dies ist z.B. durch PHPs "include"-Anweisungen möglich. Unter PHP ist es möglich, Dateien von anderen Rechnern einzubinden. Durch diese Schwachstelle können zum Beispiel unsichere Scripte von dem Rechner eines Angreifers in andere Kontexte eingebunden werden. Etliche Programmiersprachen wie Perl bieten die Möglichkeit, lokal Programme über eine Shell auszuführen. Wird ein lokales Programm mit benutzermanipulierbaren Parametern aufgerufen und die Parameter nicht entsprechend gefiltert, ist es möglich, weitere Programme aufzurufen. Auf diese Weise können etwa Dateien geändert oder sensible Daten ausgespäht werden [HF].

Gefährdungen für E-Commerce-Anbieter [SPI]:

- Cookies und andere "authentication tokens" können mittels XSS gestohlen werden. Das Ziel des Diebstahl ist es, die aktuelle Webanwendungs-Session eines Benutzer zu entführen und im eigenen Webbrowser zu klonen. Der Diebstahl ermöglicht es dem Angreifer vollen Zugriff auf einen fremden Account zu erlangen. In dem fremden System kann der Angreifer im Anschluss an den Cookie Diebstahl unter der falschen Identität frei agieren. In dem Fall eines E-Commerce-Shops könnte er somit unter fremdem Namen Waren bestellen und bezahlen. Für einen E-Commerce-Shop würde es einen enormen Imageverlust bedeuten, wenn bekannt würde, dass durch Fahrlässigkeit in der Programmierung, XSS Attacken auf Kunden möglich wären.

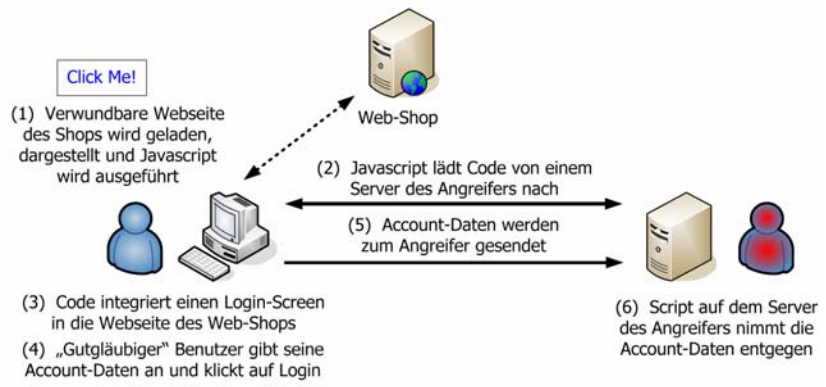

Abbildung 8: XSS – Accountdiebstahl [RM]

- XSS kann benutzt werden um Tastatureingaben abzufangen, die der User in den Browser eingibt. Das macht es sehr einfach Benutzernamen und Passwörter zu stehlen.

- Phisher benutzen XSS-Schwachstellen um realistischere Phishing-Attacken vorzubereiten.

- XSS kann dazu benutzt werden um den Bildschirminhalt zu verändern (Defacement), oder um Kommandos von einer dritten Partei zu empfangen und auszuführen.

Eine besonders aktuelle Angriffsvariante findet über die sich immer mehr verbreitende Web2.0 Entwicklung "Ajax" statt. Ajax bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Server und dem Browser, welches es ermöglicht, innerhalb einer HTML-Seite eine HTTP-Anfrage durchzuführen, ohne die Seite komplett neu laden zu müssen. Durch die Verwendung von Ajax kann eine traditionelle XSS-Attacke in einen ausgewachsenen, sich selbst verbreitenden Wurm, übergehen. So geschehen Ende 2005 bei My-Space. Ursache war ein kurzes Ajax-Skript, dass durch eine Cross-Site-Scripting-Lücke des Portals in der Lage war, sich selbst in generierte Einladungen einzubetten. Betrachte eine eingeladene Person die Einladungs-Nachricht, wurde das Script erneut zur Ausführung gebracht, um erneut alle Freunde der neuen Person einzuladen. Mit dieser Methode verbreitete sich der Wurm im Schneeballprinzip. Der My-Space-Wurm wird als einer der ersten reinen "Web 2.0-Würmer" bezeichnet, der sich ausschließlich auf Web-Applikationsebene bewegt [GB].

Session Riding (Session Fixierung)

In der Praxis findet Session Riding of im Anschluss oder im Verbund mit einem Cross-Site-Scripting-Angriff statt. Grundsätzlich besteht der Angriff darin, dass im ersten Schritt eine fremde Webanwendung (Session) entführt und diese im zweiten Schritt auf dem eigenen System geklont wird. Im Anschuss ist es dem Angreifer möglich Webanwendungen unter der entführten Identität auszuführen.

Eine Session für einen Benutzer wird von einem Webserver immer dann erzeugt, wenn sich ein Benutzer bei einer Webseite anmeldet. Dabei handelt es sich um eine Datei auf dem Webserver, in der aktuelle Status- und Benutzerinformationen gespeichert werden. Das können Identifikationsdaten aber auch der Inhalt eines virtuellen Warenkorbes sein. Diese Session erhält einen eindeutigen Identifier. Gleichzeitig speichert der Webbrowser lokal auf dem Rechner des Benutzers den Namen der Session in einem Cookie. Wenn der Benutzer jetzt eine Funktion der Webseite aufruft wie z. B. das Abschicken einer Bestellung, fragt der Server den Cookie ab und identifiziert so den Benutzer.

Der Angriff erfolgt mit Hilfe eines präparierten Links (URL). Damit der Angriff aktiviert wird, muss der angegriffene Benutzer diesen Link eigenständig ausführen. Um den präparierte Link vertrauenswürdig erscheinen zu lassen, kann dieser zum Beispiel in einer E-Mail oder auf einer mittels Cross-Site-Scripting manipulierten Webseite eingebettet sein. Beim Klick auf den Link sendet der Browser die komplette URL im Kontext des Benutzers an die Webapplikationen. Ist der Benutzer dort gerade angemeldet, so sendet der Browser auch automatisch die Cookie-Datei zur Authentifizierung mit, die der Angreifer somit abfangen kann. Durch den Besitz des Cookies verfügt der Angreifer nun über die Identität des Benutzers. Von diesem Moment an kann er eigene Daten an die Webapplikationen übergeben, mit der sein Opfer gerade in Verbindung steht [ST].

Gefährdungen für E-Commerce-Anbieter:

- Die Gefährdungsszenarien für einen E-Commerce-Anbieter können in etwa mit der Gefährdung die von Cross-Site-Scripting-Attacken ausgehen gleichgesetzt werden. Ist ein Angreifer durch eine Attacke in den Besitz der System-Rechten eines regulärer Benutzers gelangt, kann dieser Webanwendung unter dessen Identität benutzen und neu konfigurieren. Für einen E-Commerce-Shop würde dies Beispielsweise bedeuten, dass der Angreifer unter der Identität des Bestohlenen, Bestellungen aufgeben kann.

Cross-Site-Request-Forgery (XSRF/CSRF)

Cross-Site-Request-Forgery, das ins deutsche mit "Webseiten übergreifende Manipulation eines Aufrufes" übersetzt werden kann, ist die schadhafte Form eines Exploits einer Webseite. Ein Exploit ist ein Computerprogramm oder Script, welches spezifische Schwächen beziehungsweise Fehlfunktionen eines anderen Computerprogramms zur Erlangung von Privilegien ausnutzt.

Die Bezeichnung Cross-Site-Request-Forgery lässt eine Verwandtschaft mit dem Cross-Site-Scripting vermuten. Stellt aber gerade in der Art der Ausführung eines Angriffes fast das Gegenteil dar. Während das Cross-Site-Scripting das Vertrauen ausnutzt, dass ein Benutzer zu einer Webseite hat, nutzt die Cross-Site-Request-Forgery das Vertrauen aus, dass eine Webseite in einen Benutzer setzt. Der Angreifer bedient sich eines Opfers, das Zugriffsrechte auf das anzugreifende Ziel besitzt. Mit Hilfe von Spoofing- oder XSS-Attacken wird hierzu aus dem Webbrowser des Opfers ohne dessen Wissen und Einverständnis ein kompromittierter Http-Request an eine Webanwendung abgesetzt. Der Angreifer präpariert den Http-Request auf eine solche Weise, dass bei dessen Aufruf die Webanwendung die von dem Angreifer gewünschte Aktion ausführen wird.

Anders als bei anderen Angriffen auf Webanwendungen finden Cross-Site-Request-Forgery-Angriffe ausschließlich in dem Webbrowser des Opfers statt. Der Angreifer ist an der Interaktion mit der Webanwendung weder aktiv noch passiv beteiligt. Folglich eignet sich dieser Angriff auch nur zu der Manipulation von Daten in einer Webanwendung, nicht aber zum Aus- oder Mitlesen von Daten. Wenn das Ausspähen von Daten sein Ziel ist, kann ein Angreifer erst durch die Manipulation von Zugangsdaten sich gegebenenfalls in einem zweiten Schritt Zugriff auf eine Webanwendung verschaffen. Da diese Angriffe seltener stattfinden, stehen für ihre Abwehr auch weniger Ressourcen zur Verfügung. Auch lassen sie sich weniger leicht abwehren als einfache XSS-Angriffe und sind daher insgesamt gefährlicher [SAP].

Gefährdungen für E-Commerce-Anbieter:

- Durch das Ausführen von HTTP-Request-Attacken ist die Integrität von Daten auf einem Server eines Webshops gefährdet. Ist der Server korrekt konfiguriert ist nur das Auslesen, aber keine Veränderungen an dem Datenbestand möglich. Eine Schwachstelle besteht aber, wenn fälschlicherweise die PUT oder DELETE Befehle auf dem Server aktiviert wurden. In diesem Fall kann ein Angreifer Daten verändern und löschen.

- Ist ein Angreifer in der Lage Daten auf einem Server eines Webshops zu verändern kann er das Erscheinungsbild einer Webseite verändern. Hierdurch könnte er zum Beispiel Preise von Produkten ändern oder das Aussehen der Seite verleumderisch verändern.

- Manipuliert ein Angreifer die Zugangsdaten der Benutzer könnte der Fall eintreten, dass alle Benutzer ausgesperrt werden, was einem Stillstand eines Webshops gleichkommen würde.

Pufferüberläufe (Buffer Overflows)

Bis ins Jahr 2004 war eine der am häufigsten verwendeten Angriffsmethode gegen Webseiten, einen Pufferüberlauf auf einem angegriffenen System herbei zu führen und hierdurch Zugriff auf das System zu erlangen [OWASP]. Im Anschluss an den Angriff kann der Angreifer eigenen Programmcode auf dem System ablegen und ausführen. Das Gefährdungspotential für einen Betreiber einer E-Commerce-Webseite ist dementsprechend groß. Zum einem wird sein IT-System kurzfristig durch den Pufferüberlauf überlastet und außer Funktion gesetzt, für den Anbieter hat dies mögliche Geschäftausfälle und einen Imageverlust zur Folge. Desweiteren kann der Angreifer, im Moment der Systemüberlastung, eigene Programme in das System des Anbieters einschleusen. Gelingt

dies dem Angreifer, stehen ihm fast alle Möglichkeiten offen. Er könnte beispielsweise Dateien aus dem System auslesen, verändern oder löschen.

Pufferüberläufe sind, obwohl das Problem schon lange bekannt ist, immer noch eine häufige Ursachen für Schwachstellen in IT-Systemen. Ihren Höhepunkt hatten Angriffe die die Schwachstelle der Pufferüberläufe ausnutzten im Jahr 2003 als nahezu 40 Prozent der Meldungen, die das CERT herausgaben, sich auf Pufferüberläufe bezogen [CADV]. Pufferüberläufe treten überall da auf, wo Daten aus nicht vertrauenswürdigen Quellen, wie Tastatur, Netzwerk oder Benutzerdateien, in einen Speicherbereich mit statischer Größe ohne Längenprüfung geschrieben werden [BTSU]. Verursacht werden die Überläufe durch sogenannte Zeiger im Code. Zeiger sind Variablen, die auf bestimmte Stellen im Speicher gerichtet werden. In modernen Programmiersprachen wird die Verwendung von Zeigern oft eingeschränkt, da gerade durch deren Verwendung schwerwiegende Programmierfehler wie Pufferüberläufe entstehen können[ST]. Programmierfehler beim Umgang mit Zeigern können zum Beispiel Folgen haben, die zu Programmabstürzen, unbemerkter Beschädigung von Daten, oder den oben genannten Pufferüberläufen führen. Ein Pufferüberlauf wird dadurch ausgelöst, dass ein Programm in der Ausführung ständig mehr Speicher anfordert, der daraufhin anderen Programmen nicht mehr zur Verfügung steht. Im Extremfall geht das so weit, das der gesamte Hauptspeicher eines Betriebssystem voll geschrieben wird. In der Folge kann dann das Betriebssystem den laufenden Anwendungen keinen Speicher mehr zur Verfügung stellen [BH]. Im weiteren Verlauf dieses Kapitels wird aufgrund dessen auch vom Speicherüberlauf gesprochen. Ein Pufferüberlauf führt in der Praxis meist zu einem Überlaufen des Hauptspeichers. Jedes einzelne auf einem System laufende Programm verfügt für seine Ausführung über einen Pufferspeicher. Läuft dieser Pufferspeicher über, werden die Daten in den allgemeine Hauptspeicher des Systems ausgelagert. In Folge dessen wird dieser daraufhin ebenfalls voll geschrieben, was zu einem Speicherüberlauf führt.

Es gibt eine Reihe von Erweiterungsprogrammen, die unerlaubte Zugriffe auf den System-Kernel abfangen und unterbinden. Somit können Speicherüberläufe von böswilligen Angreifern nicht mehr ausgenutzt werden um Zugriff auf ein System zu erlangen [RZ]. Die Firma Cisco bietet zum Beispiel mit dem Cisco-Security-Agent eine Lösung für dieses Problem an. Durch den Einsatz des Security Agents kann die Sicherheit des Systems gesteigert werden. Allerdings gibt es generell keine hundertprozentige Sicherheit im IT Bereich. Nur wenige Monate im Einsatz zeigte sich, dass auch das Cisco Tool fehlerhaft programmiert worden war. Durch einen Fehler beim Abfangen von Speicherüberläufen konnten das Tool von einem Angreifer ausmanövriert werden. Wurde im Betrieb ein Speicherüberlauf entdeckt, forderte das Programm den Anwender in einem Dialogfenster auf, den Fehler zu bestätigen und das fehlerhafte Programm zu beenden. Bestätigte der

Benutzer den Dialog nicht innerhalb von 5 Minuten konnte ein zweiter Speicherüberlauf unbemerkt von den Sicherheitsmechanismen auftreten [BP].

Gefährdungen für E-Commerce-Anbieter:

- Wird durch eine DoS-Attacke der Absturz, eines für den Betrieb des E-Shops notwendigen Programms verursacht, fällt der E-Shop für einen Zeitraum aus. Es können in der Folge keine Waren mehr verkauft werden und das Image der Firma leidet unter diesem Systemausfall.

- Ein Dos Angriff kann zur Verfälschung von Anwendungsdaten oder zur Beschädigung von Datenstrukturen verwendet werden. Die Verfälschung von Anwendungsdaten hat einen negativen Einfluss auf das Image der Firma, da dies ihre Kunden direkt betrifft. Wird zum Beispiel ein Anschreiben aller Kunden mit der Aufforderung zu Überprüfung ihrer Daten aufgrund eines Hackerangriffes notwendig, so werden dies nur die wenigsten Kunden hinnehmen. Eine Beschädigung des Dateisystems kann des weiteren den Betrieb des E-Shops stören, da z.B. Preise von Produkten verändert worden sein können.

- Der ungünstigste Fall eines Dos Angriffes für ein E-Commerce Unternehmen würde eintreten, wenn es einem Angreifer gelänge Root-Zugang zum System zu erlangen. Dies würde dem Angreifer sämtliche Zugriffsrechte verleihen.

SQL-Injection

E-Commerce-Auftritte sind in vielen Bereichen auf die Verwendung von Datenbanken angewiesen. So werden zum Beispiel Datenbanken zur Verwaltung und der Speicherung von Daten- und Warenbeständen benötigt. SQL ist eine der geläufigsten Datenbanksprachen. Sie stellt eine Reihe von Befehlen zur Definition von Datenstrukturen, zur Manipulation von Datenbeständen (Zufügen, Bearbeiten und Löschen von Datensätzen) und zur Abfrage von Daten zur Verfügung [LH]. Benutzt ein E-Commerce-Auftritt eine SQL-Datenbank, so ist dies ein potentielles Einfallstor für Angriffe.

Eine SQL-Injection besteht darin, dass durch einen Angreifer eigene Befehlen in eine fremde SQL-Datenbank eingeschleust werden. Überprüft eine Web-Applikation Benutzereingaben nicht ausreichend, ist die Datenbank, unabhängig von der verwendeten Datenbanksoftware, verwundbar [BD_B].

Durch SQL-Angriffe können eine Vielzahl von Schäden auf einer E-Commerce-Webseite verursacht werden:

- Ausspähen von Daten der Datenbank
- Einschleusen von beliebigem Code in die Datenbank
- Veränderungen der Daten im Datenbestand
- Veränderungen der Konfiguration der Datenbank
- Erlangen von Administratorenrechten auf die Datenbank

	Erwarteter Aufruf
Aufruf	http://webserver/cgi-bin/find.cgi?ID=42
Erzeugtes SQL	SELECT author, subjekt, text FROM artikel WHERE ID=42
	SQL-Injektion
Aufruf	http://webserver/cgi-bin/find.cgi?ID=42;UPDATE+USER+SET+TYPE="admin"+WHERE+ID=23
Erzeugtes SQL	SELECT author, subjekt, text FROM artikel WHERE ID=42; UPDATE USER SET TYPE="admin" WHERE ID=23

Abbildung 9: Schadhafter SQL-Code (Beispiel)

Maßnahmen gegen SQL-Angriffen müssen in der betroffenen Webanwendung und nicht in der Software der Datenbank realisiert werden. Die Webanwendung sollte eine Prüfung der Eingabedaten auf Korrektheit und Unschädlichkeit durchführen. Die Programmierer von Anwendungen sollten sich verpflichtet fühlen, geeignete Schutzmaßnahmen, schon bei der Implementierung zu berücksichtigen. Ein Schutz des IT-Systems kann durch den Einsatz von Web-Application Firewalls, sogenannte WAF erreicht werden. Durch ihren Einsatz kann zu einem gewissen Grad verhindert werden, dass "SQL Injektion Schwachstellen" in Webanwendungen ausgenutzt werden.

Gefährdungen für E-Commerce-Anbieter:

o Ein Angreifer kann durch den Einsatz von SQL-Injections die Kontrolle über eine Datenbank erhalten, oder sogar Systembefehle ausführen. Für einen E-Commerce-Anbieter bedeutet dies, dass seine über eine Datenbank verwaltete Daten- und Warenbestände, verfälscht oder gelöscht werden können. Sind z.B. in der Datenbank Datensätze mit den Authentifizierungsdaten der Kunden gespeichert, können diese entwendet werden. Wurde bei der Speicherung der Datensätze auch noch eine unzureichende Verschlüsselung angewendet kann der Angreifer nun an Kreditkartenummern, Accountdaten oder anderen sensitiven Daten der Kunden gelangen.

Mutwillige Programmierfehler

Jede Firma, die über einen E-Commerce-Webauftritt verfügt, muss zu diesem Zweck Software einsetzen. Das kann eigene oder von Fremdanbietern bezogene Software sein. In beiden Fällen besteht die Gefahr, dass mutwillig von den Programmierern der Software, Fehler in diese eingebaut wurden. Als Programmierfehler bekannt sind zum Beispiel Hintertüren. Unter dem Begriff Hintertüren (engl. Backdoors) versteht man im Allgemeinen nicht dokumentierte Schwachstellen in Programmen. Diese Schwachstellen ermöglichen, dass auf Programme zugegriffen oder ein eigener Code eingeschleust werden kann. Backdoors können zum Beispiel offene Ports auf einem Computersystem sein. Die offenen Ports sind die Zugänge, über die auf einzelne Programme oder auf ganze Gruppen von Anwendungen zugegriffen werden können. Durch das Ausweichen auf Hintertüren können die eigentlichen Sicherheitsmechanismen eines Pogramms umgangen werden. Backdoors werden in der Regel von den Programmierern bewusst aus mehreren Gründen angelegt. Zum Einen können Hintertüren im positiven Sinne genutzt werden, um Wartungsarbeiten an schon ausgelieferten und im Gebrauch befindlichen Systemen oder Programmen auszuführen. Zum Anderen können Hintertüren aber auch für böswillige Spionage- und Sabotage-Attacken auf ein Systems genutzt werden. Da Hintertüren im allgemeinen nur den Programmierern der Anwendung bekannt sind, sind sie schwer zu entdecken (in der Regel nur durch Codeinspektion) und zu bekämpfen. Prinzipiell kann jeder, dem die Existenz einer Hintertür bekannt ist, diese ausnutzen und die Sicherheitsmechanismen eines Systems umgehen [DFS].

Für einen E-Commerce-Anbieter ist das Gefährdungspotential, dass von Hintertüren ausgeht, abhängig davon zu sehen, aus welchen Quellen er seine Software bezieht. Bezieht er seine Software von größeren Zulieferern, ist sein Risiko eher gering, da diese durch interne Qualitätsprüfungen, versteckte Hintertüren verhindern können. Von kleineren Firmen, die bei der Programmierung von Software weniger auf Qualität achten können, geht hingegen ein größeres Risiko aus [HMS].

In proprietärer Software, d.h. Software deren Quellcode rechtlich geschützt und damit nicht einsehbar ist, sind Hintertüren im Nachhinein nur schwer aufzudecken. Der Vorteil liegt diesbezüglich klar bei quelloffener Software aus der Open-Source-Bewegung, wie z.B. Linux. Hier kann der Quelltext eines potenziell schädlichen Programms nach derartigen Hintertüren leichter von geschulten Personen durchsucht werden.

Ein zweites Beispiel für mutwillige Programmierfehler, die sich in das System eines E-Commerce-Anbieters einschleichen können, sind "Logische Bomben". Logische Bomben sind Codefragmente, die sich in Betriebssystemen oder Anwendungsprogrammen

verstecken. Durch die Erfüllung von bestimmten Bedingungen werden sie ausgelöst. Dies kann zum Beispiel an einem speziellen Datum oder in Form einer speziellen Systemaktivität sein. Die Reaktionen, die die logische Bombe im System oder im jeweiligen Programm hervorruft, können sehr unterschiedlich sein. Im günstigsten Fall zeigen sie als Reaktion auf eine Tastenkombination einen Gruß eines Entwicklers an. Bekanntes Beispiel hiefür ist eine in MS-Access enthaltene Tastenkombination, die einen Würfel auf den Bildschirm erscheinen lässt, oder ein in MS-Word versteckte Spiel. Eine logische Bombe wartet folglich versteckt auf ihre Aktivierung.
Es gibt auch ungewollte logische Bomben, die durch Nachlässigkeit während der Programmierung entstehen. Ein bekanntes Beispiel für eine ungewollte logische Bombe war das Jahr-2000-Problem [BG].

Gefährdungen für E-Commerce-Anbieter:

- Um die Gefährdung, die von Logischen Bomben und sonstigen mutwilligen Programmierfehlern ausgeht, einschätzen zu können, muss zwischen der Wahrscheinlichkeit eines solchen Auftretens und der Auswirkung, wenn der Fall auftritt, unterschieden werden. Die Wahrscheinlichkeit, Opfer von solchen Programmierfehlern zu werden, kann durch Qualitätskontrollen minimiert werden. Tritt nichts desto trotz der Fall ein, ist der mögliche Schaden enorm, da der Angreifer interne Kenntnisse über die verwendeten Programme besitzt. Die Szenarien sind dann von Datendiebstahl, Veränderung der Datenbeständen, bis zum Systemausfall einzuordnen.

Malware 

Malware ist ein Begriff, unter dem schadhafte Programme zusammengefasst werden. Schadhafte Programme nutzen in der Regel Schwachstellen im Betriebssystemen oder anderen Programmen aus. Schwachstellen wiederum sind im Allgemeinen Programmierfehler, die bei der Erstellung von Software entstehen. Programme die heutzutage verwendet werden, verfügen über eine enorme Codemenge. Es kann demzufolge davon ausgegangen werden, dass nur eine geringe Anzahl der sich im Umlauf befindenden Software fehlerfrei und damit ohne Schwachstellen sind.

Nicht jede, aber viele Fehlerstellen bieten einer externen "bösartigen Software" einen Angriffspunkt. Diese externen Angriffe erfolgen aus dem Internet oder auch durch Software, die über Dienste wie E-Mail und FTP-Download auf den Rechner des Benutzers gelangen können und dort lokal ausgeführt werden.

Die Gesamtmenge aller schadhaften Software kann darin unterschieden werden, ob die Verbreitung auf ein einzelnes System beschränkt ist, oder ob eine Verbreitung über vernetzte Systeme stattfindet. Zusätzlich lässt sich Malware danach unterteilen, ob die bösartige Software selbst aktiv replizierend ist oder nicht. Selbst replizierende Software vermehrt sich im Moment ihrer Ausführung selbstständig. Nicht selbst replizierende Software kann dies nicht.

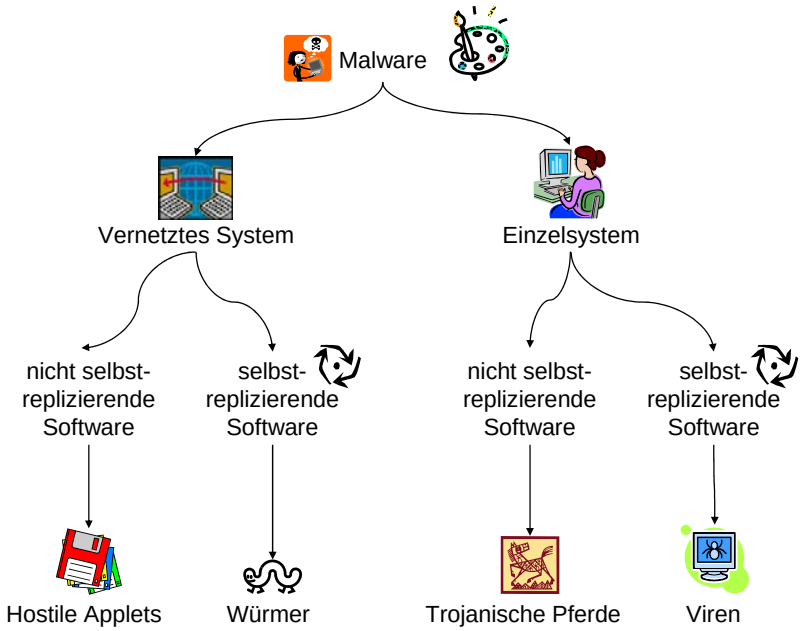

Abbildung 10: Klassifikation von Malware [EMS]

Viele bösartige Programme können nicht eindeutig einer der oben genannten Kategorien zugeordnet werden, da sie Elemente mehrere Kategorien in sich vereinen. So gibt es zum Beispiel viele Würmer, die auf befallenen Systemen trojanische Pferde oder Viren installieren und wie selbige agieren.

Computerviren

Ein Computervirus ist ein Programm, dessen Existenz und Verbreitung, in Anlehnung an sein biologisches Vorbild, von sogenannten "Wirtsprogrammen" abhängig ist. Wirtsprogramm bedeutet in diesem Zusammenhang, dass sich ein Virus in den Code eines bestehenden fremden Programms einfügt und in ihm "lebt". Ein Computervirus ist demgemäß ein sich selbst vermehrendes Computerprogramm. Hierfür schleust es sich in andere Computerprogramme ein und reproduziert sich mit deren Hilfe. Der Virus fügt sich dazu entweder an den Anfang oder das Ende eines Wirtsprogramms an und ändert dessen Funktion ab bzw. fügt seine Schadensroutine hinzu. In Folge dessen wird der Virus bei jeder

Aktivität des Wirtsprogramms mit ausgeführt und kann sich weiter verbreiten, indem er sich an noch nicht infizierte Programme anhängt.
Die Schadensroutine von Viren können von einfachen Textfenstern die geöffnet werden, bis hin zu der kompletten Löschung einer Festplatte reichen. Es gibt aber auch "nützliche" Viren, die zum Beispiel bekannte Fehler in Programmen schließen. Diese Viren als nützlich zu bezeichnen ist aber zweischneidig, da sie sich wie echte Viren unkontrolliert verbreiten und unkontrolliert durch den User und ohne Wissen des Users ihre Funktion ausführen [BG]. Ein Großteil der sich im Umlauf befindlichen Malware kann der Gruppe der Viren zugeordnet werden.

Aufgrund dessen haben sich verschiedene Klassifikationen mit dem Ziel durchgesetzt, um Viren besser einteilen zu können. Eine mögliche Klassifizierung findet über die vom Virus benötigte Plattform statt. Oft sind Viren speziell für eine Betriebssystemumgebung optimiert worden und können sich nur in dieser einen Umgebung reproduzieren. Weitere Kriterien für eine Klassifizierung sind die Art der Schadfunktion des Virus und der Speicherort im System. Zusätzlich werden Viren häufig nach ihrem Grad der Verbreitung klassifiziert [EMS].

- Bootviren:

Bootviren zählen zu den ältesten Viren. Sie infizieren in der Regel den Master Boot Record Sektor auf einer Festplatte und verhindern zukünftig das Booten eines Computers.

- Dateiviren, auch Linkviren genannt:

Linkviren infizieren ausführbare Dateien oder Programmbibliotheken innerhalb eines Betriebssystems.

- Makroviren:

Makroviren benötigen Anwendungen, die Dokumente mit eingebetteten Makros verarbeiten. Sie befallen Makros in nicht infizierten Dokumenten oder fügen entsprechende Makros ein, falls diese noch nicht vorhanden sind. Beispiele dafür sind Dateien mit den bekannten Word und Exel Dateiendungen .doc und .xls, in denen sich die Markroviren verbergen.

- Skriptviren:

Ein Skriptvirus ist ein Programm, welches nicht durch einen Compiler in Maschinensprache übersetzt werden muss. Vielmehr werden Scriptviren, wie jedes normale Skript auch, durch einen Interpreter Schritt für Schritt ausgeführt. Ein Skriptvirus wird häufig auf Webservern in Form von JavaScript abgelegt. Besucht jemand diese Webseite, wird der Virus geladen und mit dem Java-Interpreter ausgeführt [JZ].

In der Praxis fallen nicht alle Computerviren eindeutig in eine der oben genannten vier Kategorien. Häufig treten auch Mischformen auf. Das kann zum Beispiel eine Mischformen zwischen Boot- und Dateivirus sein, der sowohl Dateien als auch Bootsektoren infiziert. Ebenso gibt es Makroviren, die auch Dateien infizieren können.

Mit einem Maßnahmenkatalog kann vorbeugend gegen Viren vorgegangen werden. Der Einsatz der benötigten Mitteln nach einer Infektion ist wesentlich größer, wenn überhaupt noch eine Rettung möglich ist.

Gefährdungen für E-Commerce-Anbieter:

- Von Viren ausgehende Risiken, bestehen für einen E-Commerce-Anbieter an zwei Stellen. Die erste Risikostelle sind die Mitarbeiter. Wenn diese im Umgang mit Medien wie E-Mails oder einem sicherem Verhalten im Internet ungeschult sind, können Viren in das Firmensystem eindringen. Die Auswirkungen für das System sind dann abhängig vom Virus. Das kann von ungefährlichen Bildschirmveränderungen bis hin zu einem Systemabsturz oder Datendiebstahl reichen.

- Die zweite Risikostelle besteht in der verwendeten Systemarchitektur und Systemkonfiguration. Wird das System falsch konfiguriert, d.h. es werden Ports offen gelassen, oder es wird vergessen eine Firewall, einen Virenscanner, in die Systemarchitektur aufzunehmen, so können Viren in das Firmensystem eindringen.

► Viren und insbesondere die beiden vorgestellten Risikostellen sind generelle Probleme von IT-Systemen und keine E-Commerce-Seiten spezifische. Aufgrund dessen kann die Betrachtung von Viren für den zu erarbeitenden Kriterienkatalog vernachlässigt werden. Nichts desto trotz bilden die in dem nachfolgendem Maßnahmenkatalog aufgelisteten Schritte, eine gute Basis grundsätzlich die technische Sicherheit von E-Commere Systemen zu überprüfen.

Maßnahmenkatalog gegen Computerviren:

- Einschränkung der Benutzerrechte:

Die standardmäßig aktivierten Administratoren-Rechte sollten deaktiviert werden. Viele Virenautoren nutzen heutzutage die Schwachstelle aus, dass Microsoft Windows-Nutzer ihren PC standardmäßig als Systemadministrator betreiben.

- Installation eines Antivirenprogrammes:

Die wirkungsvollste Maßnahme gegen Computerviren besteht in der Installation eines Antivirenprogrammes. Dieses sollte, in regelmäßigen und möglichst kurzen Intervallen, aktualisiert werden. Antivirenprogramme besitzen eine Datenbank, in der die Signaturen aller bekannter Viren abgespeichert sind. Diese Signaturen werden während einer Durchsuchung des Systems mit den Daten auf der Festplatte verglichen.

- Schulung der User im Umgang mit Dateien oder Programmen aus unbekannter Quelle:

Anwender sollten niemals Dateien oder Programme aus unbekannter oder unsicherer Quelle öffnen. Gerade die Verwendung eines E-Mailsystems macht einen achtsamen Umgang mit Dateianhängen erforderlich. Viele Viren erreichen heutzutage ihren Wirt über E-Mails. Sinnvoll wäre es wenn, generell Dateianhänge in E-Mails verboten werden. Zumindest sollten E-Mailanhänge immer vor dem Öffnen mit einem Antivirenprogramm überprüft werden.

- Deaktivierung der Möglichkeit des Öffnens von Dateien aus dem Internet:

Einem Benutzer, der nicht über die administrative Rechte verfügt, sollte das automatische Herunterladen und Öffnen von Dateien aus dem Internet verweigert werden. Ebenfalls sollten bekannte Dateianhängen nicht, wie in Windows standardmäßig eingestellt, ausgeblendet werden. Hierdurch ist nicht immer erkennbar, um welchen Dateitypen es sich bei einer Datei in Wirklichkeit handelt.

Trojanische Pferde 

Als Trojanische Pferde werden Programme bezeichnet, die nach außen als nützliche Anwendungen getarnt sind. Im Hintergrund führen sie aber ohne Wissen des Anwenders eine andere, meist schadhafte Funktion aus. Trojanische Pferde, auch Trojaner genannt, können sich im Gegensatz zu Viren nicht selbstständig weiter reproduzieren. Des Weiteren sind sie nicht auf ein Wirtsprogramm angewiesen. Vielmehr ist der Trojaner ein bösartiger Code, der an ein festes, meist nützliches, Programm angebunden ist. In der Regel werden Trojaner für persönliche Zwecke von den Entwicklern einer Anwendung an diese angefügt, dies wird aber nicht von ihnen dokumentiert. Trojaner infiziere in der Regel keine anderen Programme außer dem Betriebssystem. Infektionswege für ein Trojanisches Pferd sind das Internet, E-Mails, aber auch Wechselspeicher. Die die immer häufiger anzutreffenden USB-Sticks bergen eine neue große Gefahr. Ebenfalls in Tauschbörsen kommen viele Trojaner vor, die sich als nützliche Programme ausgeben. In dem Betriebssystems eines Computers kann sich ein Trojanisches Pferd häufig unbemerkt von dem User entfalten und wichtige Daten zerstören oder Passwörter ausspionieren und an eine geheime Adresse weiterleiten.

Aufgrund dieses verdeckten Agierens werden Trojaner meist zum ausspionieren von bestimmten Personenkreisen eingesetzt.

Nicht unter den Begriff Trojaner fallen die sogenannten "Easter Eggs", die oft mit dem Begriff Trojaner verwechselt oder gleichgesetzt werden. Dies sind Programmroutinen, die ebenfalls meist nicht offiziell vom Programmhersteller dokumentiert werden, also auch versteckt sind, aber keine schädliche Funktion aufweisen. Dazu gehören beispielsweise Funktionen für die Fehlersuche, Protokollierung oder eine Liste der beteiligten Programmierer [JZ].

Personal Firewalls oder andere Programme zur Netzwerküberwachung bieten keinen Schutz vor der unbeabsichtigten Installation eines Trojanischen Pferdes. Sie können aber unter Umständen nach einer Infektion auf unautorisierte Netzwerkkommunikation aufmerksam machen.

Gefährdungen für E-Commerce-Anbieter:

- Über Trojanische Pferde kann ein Angreifer Zugang zu einem geschützten System erlangen. Hierfür muss der Angreifer erreichen, dass im E-Commerce-Unternehmen eine privilegierte Person, d.h. jemand der mit root-Rechten ausgestattet ist, sein Trojanisches Pferd installiert. Ein realer Fall aus dem Jahr 2005 zeigt, wie dies geschehen kann. Computer-CDs, die Geschäftsofferten und Projektvorschläge enthielten, wurden gezielt an Führungskräfte gesandt. Auf diesen CDs wurde ein für Antiviren-Programme nicht erkennbarer Trojaner platziert, der beim Aufruf der enthaltenen Dateien die betroffenen Rechner infizierte. Hierdurch konnte der Angreifer einen "Sniffer" in das fremde Netzwerke einschleusen. Gerade im Bereich der Wirtschaftsspionage ist dies eine oft verwendete Methode [SA]. Die Verbreitung von Trojanern als Spionagetool, gerade im Bereich mittelständischer Firmen, legt eine Aufnahme in den Kriterienkatalog nahe [ARD]. Auf der anderen Seite handelt es sich um ein generelles Architekturproblem und verfügt somit nicht über die notwendigen Merkmale für eine Aufnahme in die Kriterien.

Würmer

Würmer sind eine weitere Variante von Malware. Das Auftreten von Würmern hat in den letzten Jahren stetig zugenommen. Dies lässt sich mit der ausgeprägten Vernetzung der heutigen Computer erklären. Würmer benutzen für ihre Vermehrung sogenannte höhere Ressourcen. Dazu gehören beispielsweise Wirtsapplikationen, Netzwerkdienste oder eine Benutzerinteraktion. Die Infektion erfolgt oftmals über E-Mail oder offene Ports. Startet man eine angehängte Datei in einer E-Mail, wird der Wurm aktiviert und verschickt sich anschließend selbst an neue potenzielle Opfer weiter. Durch Sicherheitslücken in einigen E-Mail-Programmen können sich Würmer besonders schnell verbreiten. Unter Outlook und Outlook Express war es durch einen Fehler im Programmdesign möglich, dass sich verseuchte E-Mails ohne Wissen des Benutzers, an Personen aus dem Adressbuch versendeten. Dadurch kann sich der Wurm enorm schnell verbreiten und fortpflanzen.

Im Gegensatz zu Viren und Trojanischen Pferden infizieren Würmer jedoch keine fremden Anwendungen, um sich fortzupflanzen. Sie sind auf die selbstständige Verbreitung in Netzwerken ausgerichtet und stehlen vornehmlich Rechenzeit. Dadurch können sie aber innerhalb kürzester Zeit Hunderte von PCs infizieren und diese lahm legen, indem sie Rechenleistung beanspruchen.

Ein Wurm muss nicht unbedingt über eine spezielle Schadensroutine verfügen. Ein Wurm bindet allein durch seine stetige Aktivität Ressourcen. Damit verfolgt er das Ziel, sich sowohl auf den schon infizierten, als auch auf den noch nicht infizierten Systemen weiter zu verbreiten. Allein dadurch kann er enorme wirtschaftliche Schäden anrichten. Des Weiteren können Würmer die Belastung anderer Systeme im Netzwerk, wie z.B. Mailserver, Router und Firewalls erhöhen[JZ].

Gefährdungen für E-Commerce-Anbieter:

- Würmer verursachen keinen direkten Schaden an der IT-Infrastruktur eines E-Commerce-Anbieters. Da das Wurmprogramm aber sowohl auf den infizierten Systemen als auch auf den Systemen, die es zu infizieren versucht, Ressourcen zur Weiterverbreitung bindet, kann es allein dadurch wirtschaftliche Schäden anrichten. Des Weiteren können Würmer die Belastung anderer Systeme im Netzwerk wie Mailserver, Router und Firewalls erhöhen. In dieser Systemüberlastung ist die Gefährdung eines E-Commerce-Anbieters zu sehen. Es besteht die Gefahr, dass sein Shop oder Teile davon auf Kundenanfragen nicht mehr reagiert. Die Problematik ist aber generell eher in der Systemarchitektur mit Internetanbindung zu sehen, als an einer spezifischen E-Commerce-Webseite. Aufgrund

dessen ist die Betrachtung der Gefahr, die durch Würmer verursacht wird, für den zu erarbeitenden Kriterienkatalog eher zu vernachlässigen.

2.2.2 Konfigurationsfehler

Gefährdungspunkte von E-Commerce-Seiten haben ihren Ursprung oft in einer fehlerhaften oder ungenügenden Konfiguration. Das können Konfigurationsfehler sein, die das ganze System, aber auch nur einzelne Anwendungen oder spezielle Schnittstellen betreffen. Es sollten zwei Fälle von Konfigurationsfehler unterschieden werden. Im ersten Fall kann der Schaden das Unternehmen direkt treffen. Beispiele hierfür wären Hardwareausfälle oder Datenverlust. Der zweite Fall tritt ein, wenn durch Fehlkonfigurationen das Vertrauen des Kunden in das Unternehmen in Mitleidenschaft gezogen wird.

Sachgemäßer Umgang mit Konfigurationsdateien

Viele Bedrohungsszenarien habe ihren Ursprung in einer fehlerhaften Konfiguration von Systembausteinen. Das können Konfigurationsfehler der unteren Systembausteine (Protokollebene) und das eigentlichen Betriebssystem betreffen. Ebenso sind aber auch Fehler in der obersten Schicht, den Anwendungsprogammen, möglich. Die Ursache für eine fehlerhafte Konfigurationen kann meist in einer sehr umfangreichen und gleichzeitig schlecht dokumentierten Konfigurationsdatei gefunden werden. Es ist folglich die Aufgabe des Benutzer, ob er auf eigene Initiative hin Zeit und Aufwand in eine korrekte Konfiguration investiert oder nicht. In der Praxis ist ein Mittelweg realistisch, in dem der Benutzer das System nach bestem Wissen konfiguriert und diese Konfiguration auch gut dokumentiert.

Konfiguration des Systems nur mit unabkömmlichen Diensten

Ebenfalls in die Kategorie von Konfigurationsfehlern fallen laufende Dienste, die für einen regulären Betrieb nicht benötigt werden. Häufig werden unwissentlich Programme gestartet, die für den beabsichtigten Einsatzzweck des Rechners unnötig sind. Viele dieser nicht benötigten Anwendungen werden von Windows selbst oder von sogenannten "Mehrwertdiensten" gestartet. In die Kategorie der Mehrwertdienste fallen Tools wie zum Beispiel der Real-Player, das Schockwave-PlugIn, der Flash-Player oder der Quicktime-Player. Alle diese Tools legen im Moment ihrer Installation automatisch einen Eintrag in der Autostart Tabelle eines Rechners an und werden somit bei jedem Start mit geladen.

Korrekte Konfiguration von Diensten

Durch Fehler bei der Konfiguration von Diensten (z.B. SQL, PHP) kann die Integrität, der auf einem Webserver abgelegten Daten, leiden. Damit verbunden sinkt, wenn dies bekannt wird, das Vertrauen der Kunden in den Webauftritt des Anbieters. Fehlerhafte Konfigurationen von Datenbankserver, Upload-Schnittstellen, E-Mail-Clients und Ähnlichem bieten häufig Schwachstellen in Webanwendungen.

Die Gefährdungen die im Allgemeinen für E-Commerce-Anbieter von Konfigurationsfehlern ausgehen können wie folgt zusammengefasst werden:

o Aufgrund ihrer benutzerunfreundlichen Dokumentation, die viele Konfigurationsdateien von Diensten, Anwendungen oder Hardware bieten, kann es zu Schwachstellen in einem IT-System kommen. Die Gefährdungslage ist in Abhängigkeit von den Diensten, die auf dem System laufen, zu sehen. Handelt es sich dabei um Dienste, die eine Verbindung zum Netz aufbauen oder Verbindungen aus dem Netz annehmen können, ist die Gefährdung als konkret einzustufen. Besitzen diese Dienste zusätzlich auch noch Root-Rechte, können Daten ausgelesen, geändert und gelöscht werden.

Homepage-Defacement

Ist der Web-Server eines E-Commerce-Unternehmens nicht einwandfrei konfiguriert, bietet er eine Angriffsstelle für Hacker. So ist es meist ein Leichtes, die Webseiten durch andere Seiten, meist schmähenden oder pornographischen Inhalts, zu ersetzen. Vorwiegend wird zu diesem Zwecke die Startseite oder Portalseite inhaltlich verändert. Dieser Vorgang wird auch "Defacement" genannt, da die Hompage (das "Gesicht") des Unternehmens nach außen, entstellt wird. Defacement birgt ein schwerwiegendes Imageproblem für das betroffene Unternehmen und kann sogar einen recht heimtückischen Schaden darstellen, sofern das Defacement raffiniert genug erfolgt und nicht oder nur spät entdeckt wird. Beispielsweise ist es denkbar, dass auf einer Homepage die Produktpreise durch solche ersetzt werden, die 3mal höher als normal sind.
Gegen Defacement kann ein E-Commerce-Unternehmen nur dahingehend vorgehen, dass es den Provider seiner Webpräsenz zur Einhaltung von Informationssicherheitsrichtlinien verpflichtet. Dies beinhaltet insbesondere, regelmäßige Updates der verwendeten Betriebssysteme und Anwendungen durchzuführen [KM].

Gefährdungen für E-Commerce-Anbieter:

o Homepage-Defacement wird im Allgemeinen eingesetzt, um den Betreiber einer Webseite zu diffamieren. Ein direkter wirtschaftlicher Schaden entsteht in der Regel nicht. Dafür ist der Schaden, den das Image davonträgt, um so größer.

Directory Indexing

Directory-Indexing ist eine Funktion eines Webservers, die automatisch alle Dateien in einem Verzeichnis anzeigt, wenn die übliche Standarddatei "index.html" fehlt. Wenn durch einen Webserver der Inhalt eines Verzeichnisses angezeigt wird, kann dies Informationen enthalten, die nicht für die Öffentlichkeit bestimmt sind. Oft vertrauen Webadministratoren auf "Security Through Obscurity" (Sicherheit durch Unklarheit) in der Annahme, dass wenn es keine Hyperlinks zu diesen Dokumenten gibt, diese auch nicht gefunden werden bzw. niemand nach ihnen suchen kann. Durch Analyse der "robots.txt"-Datei oder dem Inhalt von Verzeichnislisten können Schwachstellen-Scanner aber sehr viele Informationen von Webserver erlangen.

Es gibt 3 verschiedene Szenarien, die es einem Angreifer ermöglichen ein unbeabsichtigtes Directory-Listing/Indexing zu erhalten:

1. Der Webserver ist irrtümlicherweise so konfiguriert, dass Directory-Indexing erlaubt ist.
2. Einige Teile des Webservers erlauben ein Directory-Indexing auch dann, wenn es in der Konfigurationsdatei deaktiviert ist oder eine entsprechende Index-Datei vorhanden ist.
3. Die Google-Cache-Datenbank kann veralterte Daten enthalten, die Verzeichnislistings von früheren Scans einer bestimmten Webseite enthalten.

Gefährdungen für E-Commerce-Anbieter:

o Obwohl Directory-Indexing an sich harmlos ist, kann es sich um ein Informationsleck handeln, die einem Angreifer mit Informationen versorgt, um weitere Angriffe gegen das System auszuführen [LJ].

2.2.3 Konzeptionsfehler (Konstruktionsfehler)

Die Architektur von Software und Hardwareumgebungen kann, trotz aller Sorgfalt, auch fehlerhaft sein. So ist es möglich, dass ein Anwendungsprogramm für den ihm vorgesehenen Einsatzbereich, nicht ausreichend konzipiert wurde. D.h. dass es quasi nicht alle notwendigen Funktionen abdeckt, die es für seinen Einsatz sollte. Wird zum Beispiel beim Design eines Webshops vergessen, eine wirksame Authentifizierung zum Login der Benutzer zu implementieren, erleichtert dieser Umstand Hackern, ihre wahre Identität zu verbergen.

Abhören und Verändern von E-Mails

Der E-Mail-Verkehr wird aktuell in den meisten E-Commerce-Unternehmen noch unverschlüsselt durchgeführt. Dieser Umstand macht es potentiellen Hackern leicht, an Netzwerkknotenpunkten, E-Mails abzuhören oder sogar Veränderungen an dem Verkehr vorzunehmen.
Für ein Unternehmen ist dieser Gefährdungspunkt in der Praxis nur schwierig zu entschärfen, da die E-Mail-Kommunikation und somit deren Sicherheit vollständig dem Provider der Webpräsenz überlassen wird. Folglich kann nur in seltenen Fällen Einfluss auf die technische Realisierung genommen werden.

Heutzutage werden immer öfter E-Mails für Auftragsabschlüsse und Bestellungen verwendet. Die vertraulichen Informationen, die diese E-Mails enthalten, legen die Notwendig dar, dass zwingend auf die Einhaltung der Integrität und der Vertraulichkeit geachtet werden muss. Dies sollte unabhängig von dem gewählten Provider oder Transportmedium geschehen. Sowohl die Integrität, als auch die Vertraulichkeit einer E-Mail kann durch eine geeignete Verschlüsselung erfüllt werden. Zu diesem Zwecke gibt es eine Reihe von Programmen, die unter Verwendung moderner Algorithmen, die Datensicherheit und Abhörsicherheit nach dem heutigen Stande der Technik garantieren können [KM].

Die aktuelle Erkenntnis, dass insbesondere mittelständige Unternehmen immer öfter Opfer von elektronisch ausgeführter Wirtschaftsspionage werden, zeigt, dass dringend ein Bedarf nach Schutz in diesem Bereich besteht. Eine Verschlüsselung der E-Mail-Kommunikation wird folglich immer mehr ein zentrales Element der Informationssicherheit sein [ARD].

Gefährdungen für E-Commerce-Anbieter:

- Das Abfangen von E-Mail-Nachrichten findet in der Regel aus Gründen der Wirtschaftsspionage statt und wird von dem Betroffenen nur in Ausnahmefällen bemerkt [BJ]. Gefährdet ist die Integrität der mittels E-Mail übermittelten Daten.

Verfälschen von E-Commerce-Daten

Für eine sichere IT-Architektur eines E-Commerce-Unternehmens gelten dieselben Anforderungen, wie sie für die oben kennen gelernte E-Mail-Kommunikation benötigt werden. Für alle Daten und Transaktionen, die innerhalb des Unternehmens anfallen, sollte die Integrität und die Vertraulichkeit beachtet werden. Zusätzlich muss, um wirtschaftlich agieren zu können, auch die stete Verfügbarkeit des IT-Systems gewährleistet werden. Ein E-Commerce System besteht zumeist aus einem Webserver, einem Datenbankserver, einer Zahlungsverkehrsplattform sowie einem dahinter liegenden ERP-System (Enterprise Ressource Planning), das den Webshop mit den Produktionssystemen und Produktionsplanungssystemen verbindet. Unter einem ERP-System wird ein System verstanden, das sowohl die Planung des Einsatzes und die Verwendung der Unternehmensressourcen übernimmt.

Grundsätzlich muss jedes dieser vier Systeme entsprechend gegen Angriffe aus dem Internet abgesichert werden. Zusätzlich sollte noch der Datenaustausch zwischen den einzelnen Systemen abgesichert werden. Hier sind vor allem zwei Anforderungen erfüllt werden:

1) Weder der Datenverkehr noch der Datenbestand darf durch Unbefugte abgehört werden können.
2) Weder der Datenverkehr noch der Datenbestand darf durch Unbefugte verändert werden können.

Beide Kriterien lassen sich durch den Einsatz einer Firewall und verschlüsselnder Protokolle realisieren. Ist ein fortwährender Datenaustausch zwischen Zweigstellen und der Zentrale eines Unternehmens notwendig, kann dieser mittels IPsec abgesichert werden. Das heißt, dass durch die Einrichtung eines Virual Private Networks (VPN), der Datenverkehr verschlüsselt und vor Zugriffen von Außen geschützt wird. Der Nachteil von VPNs liegt jedoch darin, dass sie nur zwischen fixen Endpunkten eingerichtet werden können. Demzufolge sind sie für öffentliche Anwendung, wie im Falle eines E-Commerce-Webshops, nur bedingt geeignet. Hier entsprächen die Kunden den jeweils unbekannten Endpunkten eines VPNs, die sich folglich nicht spezifisch mit einem zentralen Server des E-Commerce-Anbieters verbinden lassen würden [KM].

Logische Konzeptionsfehler eines E-Commerce-Webseite

Eine große Kraftbeanspruchung für die künstliche Intelligenz einer E-Commerce-Webseite liegt darin, auf automatisiertem Wege, eine eindeutig fehlerhafte Produktbestellung zu erkennen. Zur Verdeutlichung wird kurz ein reales Beispiel aus den USA vorgestellt:

> *"In den USA bestellen zwei Kinder beim Spielen auf der E-Commerce-Seite eines Anbieters über 10.000 mal dasselbe Spielzeug. Das ERP-System reagiert richtig auf den vom E-Commerce-System erzeugten Auftrag und produziert entsprechend viele Spielzeuge. Die Eltern lehnen jedoch die Forderung des Unternehmens mit dem Hinweis auf die Unfähigkeit der minderjährigen Kinder, Rechtsgeschäfte abzuschließen ab. Das Unternehmen zieht vor Gericht und, wie nicht anders zu erwarten, unterliegt. Das Unternehmen bleibt damit auf einem Haufen Spielsachen sitzen.*
> *Auch nach österreichischem oder deutschem Recht wäre keine andere Entscheidung zu erwarten gewesen, da minderjährige Kinder nun einmal keine für sie oder die Eltern bindenden Rechtsgeschäfte abschließen können."*

Dieser Fall verdeutlicht die Notwendigkeit für E-Commerce-Betreiber Plausibilitäts-überprüfungen von Eingaben einzuführen, die jede Bestellung auf ihre Plausibilität prüfen. Ist eine Automatisierbarkeit nicht möglich, muss ein manueller Check erfolgen, um Fehlbestellungen und betrügerische Bestellungen, ausschließen zu können.

Diese Problemstellung sollte schon während des Design eines IT-Systems gelöst werden. Der Fall zeigt auch, dass ein schlecht konzipiertes E-Commerce-System einem Unternehmen durchaus mehr Schaden als Nutzen bescheren kann.

Zusammengefasst sei angemerkt, dass auch das technische Design Rücksicht auf die rechtlichen Rahmenbedingungen, für den Betrieb eines E-Commerce-Servers nehmen sollte. Nur dann kann sichergestellt werden, dass sich der Einsatz des Systems auch entsprechend lohnt [KM].

CRLF-Injection

CRLF steht für "Carriage Return" und "Line Feed". CRLF Code besteht aus ASCII-Zeichen, die keinen sichtbaren Inhalt auf dem Bildschirm anzeigen. Ursprünglich sind sie dazu gedacht, Text zu formatieren und zusätzliche Funktionen über Tastenkombinationen bereitzustellen. Die Kombination aus "CR" und "LF" entspricht beispielsweise dem Drücken der Eingabetaste auf der Tastatur. Je nach verwendeter Anwendung wird diese dadurch angewiesen, in eine neue Zeile zu springen, oder es wird ein Befehl gesendet.

Mit diesen Sonderzeichen kann ein Angreifer die Informationen und Ausführungen von verschiedenen Protokollen manipulieren. Im Zusammenhang mit einer ungeprüfter Eingabe eines Benutzers eröffnen sich viele Möglichkeiten.

In einem Angriffen per CRLF-Injection schleusen Hacker CRLF-Anweisungen in ein System ein und versuchen hierdurch Manipulationen an diesem vorzunehmen. Begünstigt wird ein Eindringen nicht etwa durch eine Sicherheitslücke im Betriebssystem oder der Server-Software, sondern durch die fehlerhafte Konzeptionierung einer Webseite.

Findet ein Angreifer heraus, dass eine Webseite für CRLF-Injection anfällig ist, kann er die Schwachstelle nur soweit ausnutzen, wie es der Aufbau der Webanwendung erlaubt. Der Umfang des Missbrauchs hängt von der Schwere des Fehlers im System ab [WA].

Gefährdungen für E-Commerce-Anbieter:

- Viele Netzwerkprotokolle, HTTP eingeschlossen, setzen die Befehlkombination aus CR und LF ein, da alle Zeilen dieser Protokolle per CRLF getrennt werden. Schafft ein Hacker es nun, einen nicht von einem Filter kontrollierten, manipulierten HTTP-Header zu erstellen, kann er durch Umgehung der Anwendungsebene direkt mit dem Server kommunizieren. Hat der Hacker dies erreicht, kann er wieder alle Daten des Server mit dem GET-Kommando des http-Befehlssatzes auslesen. Wurde der Server unzureichend konfiguriert sind mit den HTTP-Request-Befehlen PUT und DELETE Veränderungen am Datenbestand eines Webshops möglich.

Minimalitätsprinzip für Informationen

In der Regel ist es empfehlenswert, dem Benutzer einer Webanwendung nur diejenigen Informationen bereitzustellen, die dieser für die Anwendung benötigt. Darüber hinaus gehende Informationen könnten unnötige Ansatzpunkte für eine Kompromittierung der Webanwendung liefern. Der Anbieter muss bei der Bereitstellung von Informationen zwischen der bestmöglichen Unterstützung des Benutzers und dem Schutz vor Angriffen abwägen.

Folgendes Beispiel zeigt eine Falsche und eine Richtige Methode einem Benutzer zur Eingabe seiner Daten aufzufordern:

Falsch:	"Bitte geben sie ihre Benutzerkennung und die 6-stellige PIN ein."
Richtig:	"Bitte geben sie ihre Benutzerkennung und ihre PIN ein."

Die Falsche Methode unterrichtet einen Angreifer darüber, das ein 6-stelliger Pin fehlt, was eine unnötige Preisgabe von Informationen darstellt.

Falls die Benutzerkennung richtig und das Passwort falsch eingegeben wurde:

Falsch:	"Das eingegebene Passwort ist falsch."
Richtig:	"Login nicht möglich. Benutzerkennung oder Passwort falsch"

Die Falsche Methode unterrichtet einen Angreifer darüber, dass die verwendete Benutzerkennung korrekt war. Die Richtige Methode versorgt ihn nicht mit dieser Information.

Falls eine Benutzerkennung nicht existiert:

Falsch:	"Der Benutzer existiert nicht."
Richtig:	"Login nicht möglich. Benutzerkennung oder Passwort falsch"

Über Hilfeseiten und Systemmeldungen können Angreifer an Informationen über den Aufbau von Webanwendungen gelangen. Diese sollten daher nur einem angemeldeten Benutzer angezeigt werden. Das bedeutet, dass Hilfe-Seiten, die Informationen über Zusammenhänge von geschützten Anwendungen enthalten, auch nur in geschützten Bereichen zugreifbar sein dürfen[BSI_A].

Gefährdungen für E-Commerce-Anbieter:

o Die Gefährdungslage eines Anbieters ergibt sich aus der Menge an Informationen, die dieser einem möglichem Angreifer bereitstellt. In den obigen Beispielen ist erkennbar, dass ein Angreifer im Ausschlussverfahren nacheinander einen Benutzernamen und das zugehörige Passwort erraten kann. Er benötigt für eine Bestätigung auf Korrektheit nur eines von beiden.

IP-Spoofing

Mittels IP-Spoofing kann von böswilligen Angreifern die Identität einer anderen Person oder eines fremden Systems angenommen werden. Der Angreifer nimmt zu diesem Zweck die IP-Adresse der anderen Person an, was im Internet mit einer Identität gleichgesetzt werden kann. Das Ziel durch diesen Identitätsdiebstahl von IP-Spoofing ist es, Zugang zu einem fremden System zu erlangen. Im Falle einer E-Commerce-Webseite könnte auf diese Weise zum Beispiel ein Angreifer an die Authentifizierungsdaten eines Kunden gelangen. Hierfür muss er nur kurzfristig die Identität der E-Commerce-Webseite annehmen. Dies geschieht über folgenden Weg. Der Angreifer versendet eigene IP-Pakete an die angegriffene Person, den Kunden der E-Commerce-Webseite. Die gefälschten IP-Pakete müssen, um die Täuschung zu ermöglichen, die vom System des Angegriffenen erwartete Adresse der E-Commerce-Webseite enthalten. Ansonsten würde das System dieses Paket nicht akzeptieren. Dies erreicht der Angreifer, indem er im IP-Paket gewisse Konfigurationsbereiche, die im Header des Pakets enthalten sind fälscht.

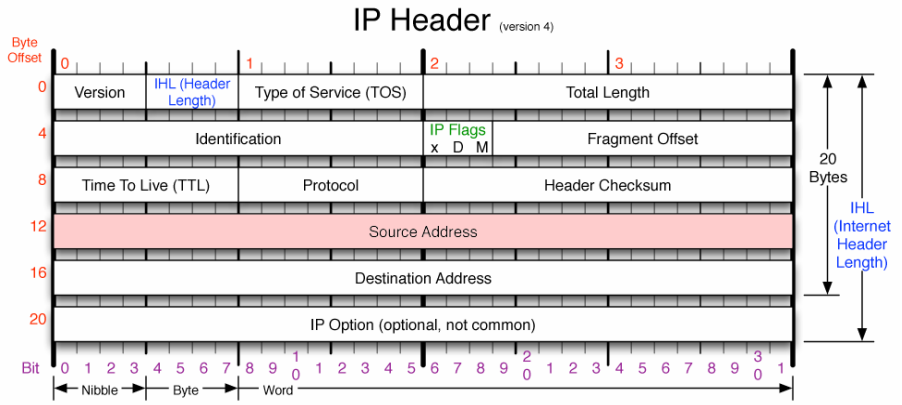

Abbildung 11: IPv4 Header [PL]

In Abbildung 11 ist ein IPv4-Header (Kopfteil) abgebildet. In diesem Codestück gibt der Angreifer im Feld "Source Address" die von dem angegriffenen System erwartete Adresse an. Der Angreifer ersetzt seine eigene IP-Adresse, die ursprünglich an dieser Stelle im IP-Header stehen würde, durch die IP-Adresse der E-Commerce-Webseite. Durch diesen Austausch wird das gefälschte IP-Paket von dem angegriffenen System akzeptiert.

IP-Spoofing nutzt die Tatsache aus, dass in vielen Protokollen, die auf TCP/IP basieren, die Authentifizierung der kommunizierenden Akteure nur über die IP-Adresse erfolgt. Diese Adresse ist aber, wie oben angeführt, leicht zu fälschen. Nutzt man darüber hinaus noch aus, dass die von den Rechnern zur Synchronisation benutzten Sequenznummern bei dem Aufbau einer TCP/IP-Verbindung ohne viel Aufwand zu erraten sind, ist es möglich, Pakete mit jeder beliebigen Absenderadresse zu verschicken. IP-Spoofing wird besonders oft innerhalb von Firmennetzwerken mit dem Ziel angewendet, dass sich bestimmte Mitarbeitergruppen automatisch für bestimmte Anwendungen autorisieren können.

Das BSI stellt eine Liste zur Verfügung, die Anwendungen auflistet, die anfällig für IP-Spoofing Attacken sind. Ein Systemadministrator kann mit Hilfe dieser Liste sein System auf IP-Spoofing anfällige Anwendungen hin überprüfen. Um IP-Spoofing vorzubeugen, sollten diese Dienste, wenn möglich, von der Firewall geblockt werden. Das Problem einer korrekten Authentifizierung bleibt aber weiterhin bestehen, da auch die im OSI-Layer 2 angesiedelten TCP- und MAC-Adressen gefälscht werden können[GSK_G1].

Gefährdungen für E-Commerce-Anbieter:

- Durch IP-Spoofing-Attacken verwundbar sind alle Dienste, die Authentifizierungen von Benutzer auf Basis von IP-Adressen durchführen. Der Sinn dieser Authentifizierungsmethode liegt darin, dass keine Eingabe von Passwörtern benötigt wird. Dies kann in Firmenstrukturen sinvoll sein, da firmeneigenen Rechner über bekannte IP-Adressen ihre Echtheit beglaubigen können.

DNS-Spoofing

Domain Name System (DNS)-Adressen werden im Internet eingesetzt, um komfortabler mit anderen Rechnern kommunizieren zu können. Für die Kommunikation zweier Rechner benötigt man deren IP-Adressen. Die IP-Adresse setzt sich aus vier Zahlen zwischen 0 und 255 zusammen. Da solche Nummern nicht sehr einprägsam sind, wird in der Praxis anstatt der IP-Adresse fast immer ein Name, die DNS-Adresse, verwendet. Mit dieser Technik kann jede Internetseite sowohl durch Eingabe ihres Namens, als auch über ihre IP-Adresse erreicht werden.

Für die Speicherung der Rechnernamen und der zugehörigen IP-Adressen werden sogenannte Nameserver (DNS-Server) eingesetzt. Für die Zuordnung zwischen Namen und IP-Adressen gibt es zwei Datenbanken: In der einen wird einem Webseitennamen seine IP-Adresse zugewiesen und in der anderen einer IP-Adresse der zugehörige Webseitenname. Diese Datenbanken müssen miteinander nicht konsistent sein. Von DNS-Spoofing ist die Rede, wenn es einem Angreifer gelingt, die Zuordnung zwischen einem Rechnernamen und der zugehörigen IP-Adresse zu fälschen, d.h. dass ein Name in eine falsche IP-Adresse bzw. umgekehrt umgewandelt wird. Indem der Angreifer die Einträge in den DNS-Tabellen ändert, kann er erreichen, dass Anfragen auf einen beliebigen Rechner umgeleitet werden [GSK_G2].

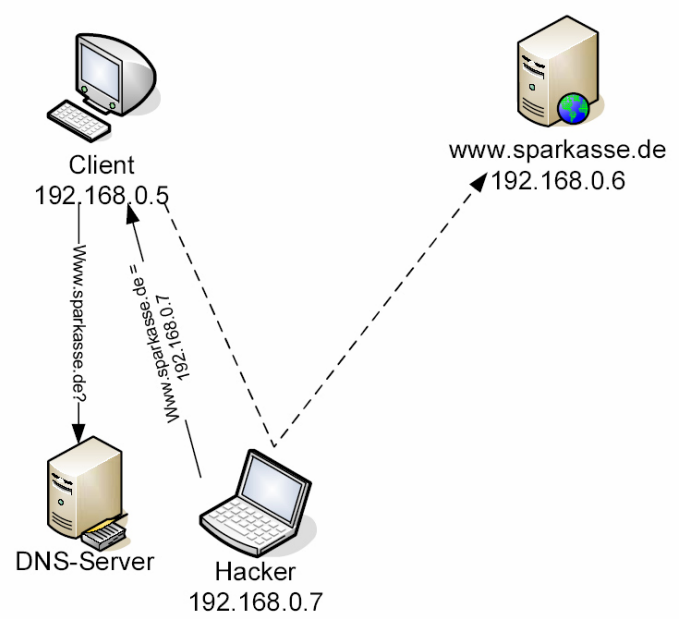

Abbildung 12: DNS-Spoofing (Beispiel) [SR]

Die Anfälligkeit für Angriffe mittels DNS-Spoofing hängt davon ab, wie das Netz des Angegriffenen konfiguriert ist. Da ein Rechner niemals alle Information über sämtliche weltweiten IP-Adressen und zugehörige Rechnernamen speichern kann, ist man immer auf den Austausch von Informationen angewiesen. Besteht der Verdacht, dass ein DNS Eintrag

verändert wurde, sollte ein zweiten DNS-Server kontaktiert und der Eintrag abgeglichen werden.

Gefährdungen für E-Commerce-Anbieter:

o DNS-Cache-Poisoning ist eine besonders leistungsfähige DNS-Spoofing-Variante die insbesondere E-Commerce-Unternehmen treffen kann. Von dem Angreifer werden gezielt falsche Informationen in den DNS-Cache eines DNS-Servers eingefügt. Zu diesem Zweck ist es nicht erforderlich, in den Server einzubrechen. Es werden lediglich fehlende Sicherheitsaspekte im DNS-Protokoll ausgenutzt, um den Server dazu zu bringen, zusätzliche Einträge in seinen Cache aufzunehmen. Ist der DNS Cache erst einmal verseucht, beantwortet der Server alle Anfragen für die injizierten Domains mit den gefälschten IP-Adressen. Ziel ist es, Kunden, die unwissentlich auf diese gefälschten Daten zugreifen, auf manipulierte Webseiten zu lenken. Da sich die meisten Internet-Protokolle auf die Korrektheit der Domainnamen verlassen, kann durch DNS-Spoofing zum Beispiel auch der Mailverkehr umgeleitet werden. [EC_A].

ARP-Spoofing

Das Address Resolution Protocoll (ARP) hat die Aufgabe, Medium Access Control Adressen (MAC-Adressen) IP-Adressen zu zuordnen. Jede Netzwerkkarte besitzt eine feste und weltweit eindeutige MAC-Adresse. Zu diesem Zweck wird vom Adress Resolution Protokoll in einem Router im lokalen Netzwerk eine Tabelle angelegt, in der jeder MAC-Adresse eine zugehörige IP-Adresse zugeordnet wird. Diese Tabelle wird im Laufe der Zeit aktualisiert und erweitert. Kommt ein Paket am Router an, sieht dieser in seiner Tabelle nach, ob ein Eintrag (MAC-Adresse) zur IP-Adresse vorhanden ist. Ist dies nicht der Fall, sendet der Router mittels Broadcast eine Anfrage an alle lokal angeschlossenen Rechner. Nur der Rechner, der mit dieser Anfrage angesprochen werden sollte, antwortet dem Router. Daraufhin aktualisiert der Router seinen ARP-Cache mit der dazugelernten IP/MAC-Verknüpfung.

ARP-Spoofing beruht auf dem gezielten Senden von gefälschten ARP-Paketen, mit dem Ziel, den Datenverkehr zwischen zwei Hosts in einem Computernetz abzuhören oder zu manipulieren. Man spricht von einer "man in the middle" Attacke. Der Angreifer sendet jeweils ein gefälschtes Paket an den eigentlichen Empfänger und ein Paket an den Sender. Hierdurch verändert der Angreifer den Weg des Paketes durch das Netzwerk. Zuvor kommunizierten die beiden angegriffenen Systeme noch direkt miteinander. Nun sitzt der Angreifer zwischen den Systemen und fängt deren Nachrichten ab, ohne dass diese es merken [GSK_M1].

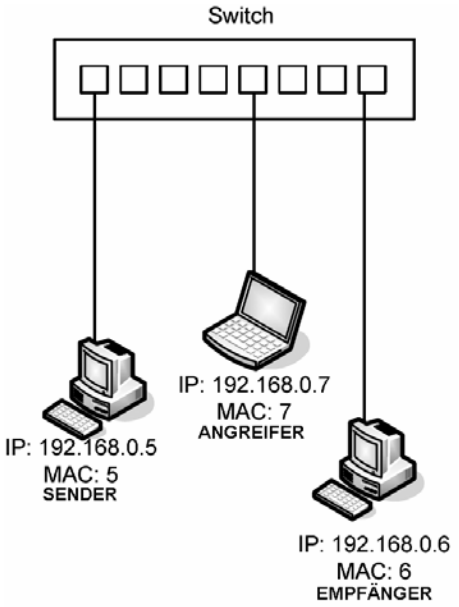

Abbildung 13: ARP-Spoofing (Beispiel)

In Abbildung 13 versucht der linke Rechner (192.168.0.5) eine Nachricht an den rechten Rechner (192.168.0.6) senden. Dafür sendet er eine ARP-Anfrage (Request) in das lokale Netzwerk. Der Angreifer reagiert auf diese Anfrage mit der Antwort, dass der gewünschte Rechner (192.168.0.6) im lokalen Netzwerk unter der MAC Adresse 7, seiner eigenen MAC-Adresse, zu erreichen wäre. Sendet jetzt der linke Rechner eine Nachricht an den rechten Rechner, so gibt er im Header der Nachricht die MAC-Adresse mit der Nummer 7 an. Der Switch sieht infolgedessen in seiner Tabelle nach an welchem Port die MAC-Adresse mit der Nummer 7 liegt. Sendet daraufhin die Nachricht an den Angreifer. Anschließend muss der Angreifer sein System nur noch so konfigurieren, dass alle Nachrichten an den rechten Rechner weitergeleitet werden. Für die andere Senderichtung wird äquivalent vorgegangen [SR].

ARP-Spoofing zu erkennen und zu verhindern, ist in der Praxis nicht ohne Weiteres, zu realisieren. Eine Möglichkeit ist es, das ARP-Protokoll vollständig im System zu deaktivieren und mit statischen Tabellen zur Umsetzung von IP-Adressen zu Hardware-Adressen zu arbeiten. Diese Lösung wäre in der Praxis sehr zeitraubend, weil ohne das ARP-Protokoll die ARP-Tabellen ständig von Hand aktualisiert werden müssten.

Eine bessere Lösung setzt am eigentlichen Grundproblem an. Prinzipiell wird zunächst jede ARP-Nachricht von fast allen Betriebssystemen akzeptiert. Hier wäre es sinnvoll, Tools einzusetzen, die das System bei der Verarbeitung von ARP-Nachrichten unterstützen. Diese Tools überwachen, wer Nachrichten wann schickt und welche Informationen die Nachrichten enthalten. Offensichtlich gefälschte ARP-Nachrichten lassen sich so erkennen und verwerfen. Dieser Ansatz wird zum Beispiel von ArpWatch und XArp implementiert.

Gefährdungen für E-Commerce-Anbieter:

- ARP-Spoofing ist eine einfache, aber effektive Art, Verbindungen in einem lokalen Netzwerken zu belauschen, umzuleiten, zu übernehmen oder zu verweigern. Das Verweigern von Verbindungen entspricht einer Denial-of-Service-Attacke (DoS). ARP-Spoofing kann aber nur in einem lokalen Netzwerk durchgeführt werden, das ARP-Nachrichten das lokale Netzwerk nicht verlassen. Ein ARP-Angriff ist für einen Angreifer einfacher zu managen, da er als zwischengeschalteter Proxy agiert und nicht blind ist, wie im verwandten IP-Spoofing. Folglich sind mögliche Schadensszenarien für einen E-Commerce-Anbieter wesentlich größer als bei IP-Spoofing [GS].

URL-Spoofing

Uniform Resource Locator (URL)-Spoofing wird im Internet angewendet, um dem Besucher einer Internetseite eine falsche Identität vorzutäuschen. Es werden zwei Arten von URL-Spoofing unterschieden.

Unter der einen Art versteht man das Vortäuschen von falschen Links, das heißt, wenn Links auf andere Seiten führen als sie dies vorgeben. Hiermit kann ein Opfer beispielsweise dazu verleitet werden, Dateien zu installieren oder Skripte auszuführen, die böswilligen Code enthalten.

Die zweite Art von URL-Spoofing besteht darin, einem Anwender, ohne das dieser es bemerkt, eine gefälschte Webseite anzubieten. Zu diesem Zwecke wird das Erscheinungsbild der zu fälschenden Seite in jedem Detail imitiert. In der näheren Vergangenheit wurden Internetauftritte von Banken oder Onlinehändlern so gut gefälscht, dass Opfer dazu verleitet wurden, persönliche Daten, wie zum Beispiel Pins, Tans oder Bankdaten, preiszugeben.

Eine technische Lösung für die Verhinderung von URL-Spoofing , wird allen durch den Einsatz des jeweils aktuellsten Internetbrowsers gegeben. Viele Schwachstellen im Bereich von Java und aktiven Inhalten bezüglich URL-Spoofing wurden inzwischen umgesetzt. Des Weiteren sollten die Benutzer im Umgang mit dem Internet geschult werden. Ein Katalog für das Verhalten im Internet kann hier schon ein großer Schutz sein.

Gefährdungen für E-Commerce-Anbieter:

o URL-Spoofing wird oft im Verbund mit Cross-Site-Scripting-Attacken eingesetzt, um einen Link vertrauenswürdiger oder unauffälliger erscheinen zu lassen. Hieraus ergibt sich für ein E-Commerce-Unternehmen eine ähnliche Gefährdungslage wie bei einem Cross-Site-Scripting Angriff. Das reicht vom Verlust der Datenintegrität, dem Identitätsdiebstahl, dem Abfangen von Tastatureingaben bis hin zu der Vorbereitung von Phishing-Attacken.

TCP-Spoofing

TCP-Spoofing basiert auf dem Abfangen und dem Fälschen von Transmission Control Protocol-Paketen (TCP-Paketen). Im Internet werden bei einer TCP/IP-Übertragung die gesendeten Nachrichten zuerst in die kleineren Internet Protokoll-Pakete (IP-Pakete) und diese dann in die größeren TCP-Pakete unterteilt. Das Fälschen von (TCP)-Paketen erfordert einen höheren Aufwand als das Fälschen von IP-Paketen. Die Schwierigkeit liegt darin, dass TCP-Pakete Sequenznummern besitzen. Die Anfangssequenznummer wird während des ersten Verbindungsaufbaus zwischen Client und Server ausgehandelt. Die Einführung von Sequenznummern macht aus TCP eine zuverlässige Übertragungsmethode. Da das unterhalb von TCP angelegte Internet Protokoll keine richtige Reihenfolge der übertragenen Paketen gewährleisten kann, erfolgt die Sortierung der Pakete im TCP über die Sequenznummern. Nach einer Übertragung von Paketen zu einem Empfänger werden die Sequenznummern angewendet, um die richtige Reihenfolge der übertragenen Paketen wieder herzustellen und um verlorene Pakete zu erkennen. Tritt ein Paket mit falscher Sequenznummer auf, wird es verworfen. Die Schwierigkeit für den Angreifer liegt in der Tatsache, dass er nicht nur die Absenderadresse fälschen, sondern dass er zusätzlich auch noch die zu jedem Zeitpunkt unterschiedliche Sequenznummern imitieren muss. Nur wenn er beides korrekt nachahmt, kann er eine Verbindung aufbauen und übernehmen.

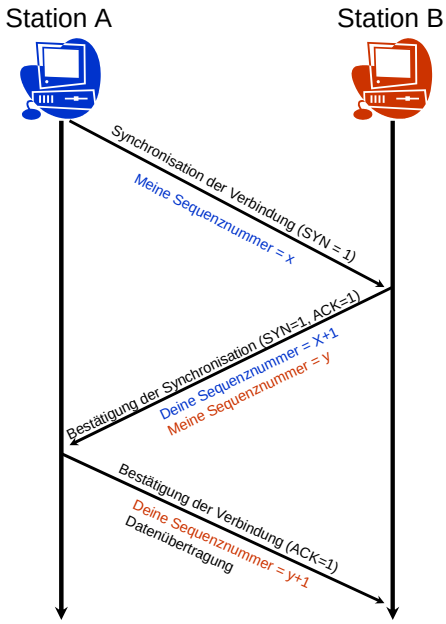

Abbildung 14: Drei Wege Handshake unter TCP

Der Aufbau einer Verbindung unter TCP erfolgt über einen drei-Wege-Handshake. Zu Beginn des Verbindungsaufbaus wird von der Station A (Client) an die Station B (Server) die Verbindungsanfrage (SYN-Flag) mit einer von ihm zufällig ausgewählten Sequenznummer übermittelt. Die Station B antwortet auf diese Anfrage mit einem SYN und ACK Flag. Die Antwort beinhaltet die um den Wert eins erhöhte Sequenznummer der Station A iund die eigene Sequenznummer. Die Station A beantwortet dies mit einem ACK. Ab jetzt kann die Übertragung von Daten stattfinden. Jedes TCP Paket wird mit einer fortlaufenden Sequenznummer, die im Header enthalten ist, gekennzeichnet.

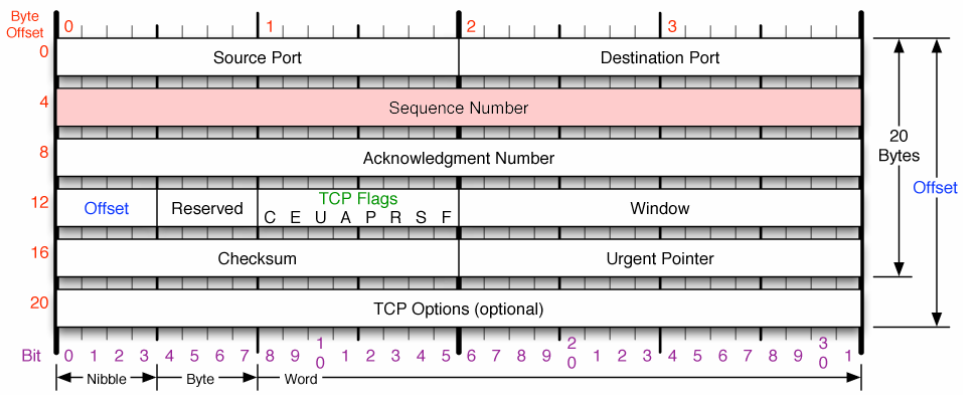

Abbildung 15: TCP Header [PL]

Ein Angriff auf ein System läuft in mehreren Phasen ab. In der ersten Phase muss der Angreifer den Rechner deaktivieren, dessen TCP-Adresse er übernehmen will. Dies kann zum Beispiel mit einem Denial-of-Service Angriff geschehen. Hierdurch kann der Rechner nicht mehr auf TCP-Pakete reagieren. Jetzt kann der Angreifer seinen Rechner so

konfigurieren, dass er die "entführte" Absenderadresse übernimmt. In dem nächsten Schritt sendet der Angreifer ein SYN-Paket an den Server. Der Server antwortet, wie im normalen drei-Wege-Handshake, jeweils mit einem SYN und einem ACK-Paket. Diese Pakete kommen jedoch weder auf dem Rechner des Angreifer noch bei dem durch den DOS-Angriff ausgeschalteten Rechner an. Infolgedessen stellt sich dem Angreifer die Problematik, dass er keine Informationen über die korrekte Sequenznummer besitzt. Ohne die korrekte Sequenznummer wird der Server keine Pakete vom Angreifer akzeptieren und selbige verwerfen. Es kommt keine Verbindung zwischen Angreifer und Server zu Stande. Da Sequenznummern aber immer nach einem festgelegten Schema erstellt werden, können diese von einem Angreifer nach dem folgenden Schema hergeleitet werden.

1) Wenn keine Verbindung aufgebaut ist, wird die Sequenznummer jede Sekunde um 128.000 erhöht.
2) Besteht eine Verbindung zwischen Client und Server, wird die Sequenznummer jeder Sekunde um 64.000 erhöht.

Mendax ist zum Beispiel ein solches Tool, das ein Angreifer zur Bestimmung von TCP-Sequenznummern verwendet werden kann. Baut der Angreifer mit einem drei Wege Handshake (unprivilegierte Verbindung) eine Verbindung mit dem Server auf, kann er mit diesen beiden Regeln versuchen, die erforderliche Sequenznummer zu berechnen. Ermittelt der Angreifer die Sequenznummer, kann er den Handshake vollenden und erlangt Zugriff auf das System [EC_B].

Gefährdungen für E-Commerce-Anbieter:

o TCP-Spoofing kann von Angreifern benutzt werden, um in Verbindungen von E-Commerce-Unternehmen einzudringen. Zusätzlich kann durch das Verfälschen der TCP-Pakete die Identität des Angreifers verdeckt bzw. eine andere vertrauenswürdige Identität angenommen werden.

2.2.4 Fehler resultierend aus menschlichem Verhalten

Menschliches Handeln kann und wird nie hundertprozentig frei von Fehlern sein. Diese möglichen Fehler im Vorfeld zu erkennen und zu verhindern ist Aufgabe der Software und deren Entwickler. Dies bedeutet, dass die Entwickler schon während der Anfertigung einer Software mögliche Fehler eines Benutzer bei der Bedienung ihrer Software berücksichtigen und abfangen müssen. Fehler, die aus menschlichem Handeln resultieren, können als Bedienungsfehler verstanden werden. Selbstverständlich sind menschliche Fehlhandlungen an sich keine technischen Schwachstellen, haben aber selbige zur Folge. Aufgrund dessen werden sie in diesem Unterkapitel des Kapitel 2.2 "Technische Schwachstellen" erwähnt.

	Bedeutung heute		Prognose		Schäden	
	Rang	Priorität	Rang	Priorität	Rang	ja, bei
Irrtum und Nachlässigkeit eigener Mitarbeiter	1	1,50	2	1,70	2	51%
Malware (Viren, Würmer, Troj. Pferde,...)	2	1,34	1	2,80	1	54%
unbefugte Kenntnisnahme, Informationsdiebstahl, Wirtschaftsspionage	3	0,60	4	1,14	8	9%
Software-Mängel/-Defekte	4	0,57	5	0,96	3	43%
Hacking (Vandalismus, Probing, Missbrauch,...)	5	0,48	3	1,26	5	9%
Hardware-Mängel/-Defekte	6	0,40	8	0,32	4	38%
unbeabsichtigte Fehler von Externen	7	0,30	9	0,26	7	15%
höhere Gewalt (Feuer, Wasser,...)	8	0,24	11	0,04	9	8%
Manipulation zum Zweck der Bereicherung	9	0,17	7	0,43	10	8%
Mängel der Dokumentation	10	0,15	10	0,20	6	17%
Sabotage (inkl. DoS)	11	0,12	6	0,55	11	8%
Sonstiges	12	0,03	12	0,00	12	3%

Basis: 161 Antworten (Bedeutung), Ø 124 (Prognose), Ø 128 (Schäden)

Abbildung 16: KES-Sicherheitsstudie 2004 [KES]

Bedienungsfehler

Die von Microsoft im Jahr 2004 in Auftrag gegebene KES-Sicherheitsstudie brachte zum Vorschein, dass in 51% der befragten Unternehmen Betriebsausfälle aufgrund von Irrtümern und Nachlässigkeiten ihrer Mitarbeiter bei der Bedienung der IT-Systeme auftraten. Der Unterschied zwischen Bedienungsfehlern und Konfigurationsfehlern liegt im Folgendem: Bedienungsfehler entstehen immer dann, wenn nicht nur einzelne Einstellungen vom Benutzer falsch gemacht werden, sondern IT-Systeme oder IT-Anwendungen von ihm falsch behandelt werden. Häufig werden schützenswerte Informationen potenziellen Angreifern so ungewollt durch Bedienungsfehler zur Verfügung gestellt. Das können zum

Beispiel Benutzernamen, Mailadressen, Passwörter, Rechnernamen, Programmversionen, Name und Version des verwendeten Betriebssystems, und vieles mehr sein.

Schwachstellen bei der Verwendung von Passwörtern

Schwachstellen entstehen oft durch ein unzureichende Verwendung von Passwörtern. Passwörter werden zur Authentifizierung von Benutzern oder für Zugriffsregelung auf Hardwarekomponenten eingesetzt. Ein Benutzer besitzt immer sowohl eine Kennung als auch ein zugehöriges Passwort. Erst durch die kombinierte Eingabe dieses Paares bekommt er Zugriff auf ein System, Programm oder einen Dienst. Die Sicherheit der Zugangs- und Zugriffsrechteverwaltung eines Systems ist entscheidend davon abhängig, ob das Passwort korrekt gebraucht wird. Dabei ist es empfehlenswert, eine Regelung zum Passwortgebrauch einzuführen und die Benutzer diesbezüglich zu unterweisen [GSK_M2].

Selbst die Nutzung von komplexen Authentifikationsverfahren hilft wenig, solange von den Benutzern nicht sorgfältig mit den benötigten Zugangsmitteln umgegangen wird. Passwörter werden von Benutzern oft aus Bequemlichkeit an andere Benutzer weitergegeben. Innerhalb von Arbeitsgruppen werden Passwörter geteilt, um jedem Mitarbeiter den Zugriff auf gemeinsam zu bearbeitende Dateien zu erleichtern. Der Zwang zur Passwortbenutzung wird oft als lästig empfunden und dadurch unterlaufen, dass Passwörter nie gewechselt werden, oder alle Mitarbeiter dasselbe Passwort benutzen. Durch die Vielzahl verschiedener Passwörter und PINs können sich Benutzer diese oftmals nicht alle merken. Daher werden Passwörter immer wieder vergessen, was teilweise zu hohem Aufwand führt, um mit dem System weiterarbeiten zu können. Authentikationstoken können ebenso verloren werden. Der Verlust von Passwörtern oder Token kann bei sehr sicheren IT-Systemen sogar dazu führen, dass ein Zugang zu alle Benutzerdaten verloren geht. Passwörter werden oft notiert, damit sie nicht vergessen werden. Dies ist solange kein Problem, wie sie sorgfältig, also vor unbefugtem Zugriff geschützt, aufbewahrt werden. Leider ist dies nicht immer der Fall. Ein klassisches Beispiel ist die Passwortaufbewahrung unter der Tastatur oder auf einem Klebezettel am Bildschirm. Auch Authentikationstoken finden sich gerne unter der Tastatur [GSK_G3].

In der Praxis haben sich Passwörter bewährt, die aus ganzen Sätzen gebildet werden. Sie bieten den Vorteil, dass sie leichter zu merken sind und gleichzeitig eine hohe Sicherheit bieten. Zeichenweise veränderter Sätze, wie "dIE bANANNE*3 durch 1/4 nIKOTIN" (32 Zeichen) oder die Verwendung der Anfangsbuchstaben eines Satzes ("Hd7Bsd7Z", gebildet aus "Hinter den sieben Bergen sind die sieben Zwerge"), sind Beispiele für sehr sichere Passwörter [JZ].

Gefährdungen für E-Commerce-Anbieter:

- Ein E-Commerce-Anbieter muss zum einem bei der Wahl der verwendeten Passwörter angemessen vorgehen. So dürfen zum Beispiel keine "Standardpasswörter", aus einer werkseitigen Konfiguration, benutzt werden. Das trifft insbesondere auf IT-Komponenten wie Router, Switche und Webserver zu. Gelangt ein Angreifer in den Besitz eines Passwortes einer IT-Komponente, kann er dessen Konfiguration verändern. Die Szenarien die sich daraus ergeben sind abhängig von der entführten Hardware. Wird die Konfiguartion eines Routers durch ein erratenes Passwort von einem Angreifer verändert, kann es zu einer Umleitung von Internetverbindungen und bei Webservern zur Veränderung der Datenbestände der Webseite kommen.

- Zum anderen müssen die Mitarbeiter im Umgang mit Passwörtern geschult werden. Die Unternehmenspolitik soll so transparent wie möglich sein, so dass jeder Mitarbeiter sich seiner Verantwortung gegenüber des eigenen Passwortes bewusst ist. Gelingt es einem Angreifer die Identität eines Mitarbeiters durch Kenntnis dessen Passwortes anzunehmen kann er mit dessen Systemrechten agieren. Die Schädigungen, die sich daraus ergeben können, reichen von Systemausfällen bis hin zu Datendiebstahl.

Administrative Fehler

Eine Reihe von Schwachstelle ergeben sich für einen E-Commerce-Anbieter aus Fehlern in der Administration des IT-Systems. So stellen für den regulären Betrieb nicht benötigte Netzzugangsmöglichkeiten eine Bedrohung dar. Die Aufgabe eines Administrators liegt darin, solche nicht benötigten Schnittstellen im System zu deaktivieren. Eine fehlerhafte Administration beeinträchtigt die Sicherheit eines IT-Systems, wenn dadurch IT-Sicherheitsmaßnahmen missachtet oder umgangen werden.

Ein weiteres Problem kann aus einer ungenügenden Verwaltung der Zugangskonten entstehen. Dies ist zum Beispiel dann der Fall, wenn Zugangskonten verwendet werden, die mehr Zugriffsrechte besitzen als für ihre individuelle Tätigkeit unbedingt nötig wäre. Hierdurch erhöht sich die Gefahr von Schäden, die von einem entführten Benutzerkonto oder durch Viren und Trojanischen Pferden verursacht werden können. Grundsätzlich müssen alle Zugangskonten, die nicht mehr benötigt werden, deaktiviert werden.

Die Konfiguration von Betriebssystemen oder Systemprogrammen im Zustand ihrer Auslieferung weist in den seltensten Fällen alle Merkmale einer sicheren Installation auf. Wird von einem Administrator keine angemessene Anpassung an die konkreten Sicherheitsbedürfnisse der E-Commerce-Firma vorgenommen, stellt dies eine Gefährdung

des Systems dar. Besondere Beachtung müssen Systemkomponenten finden, die im Falle einer fehlerhaften Administration negativen Einfluss auf den Schutz anderer Systeme haben. Desweiteren stellen Modifikation von Sicherheitseinstellungen und die Erweiterung von Zugriffsrechten eine potentielle Gefährdung der Gesamtsicherheit dar.

Ist im IT-System des E-Commerce-Anbieters ein Datenbankmanagementsystem (DBMS) enthalten, kann es hier zu weiteren administrativen Fehlern kommen. Diese können in Form von zu großzügig vergebene Benutzerrechte, in unregelmäßiger Datenbanküberwachung oder aber in einer mangelhaften Datensicherung bestehen.

Leichtgläubigkeit

Die Leichtgläubigkeit eines Menschen sollte bei der Betrachtung von Schwachstellen nicht vernachlässigt werden. Sie ist die Grundlage von "Social Engineering Angriffen" und der Verbreitung von Malware. So können Menschen leicht zu einem Klick auf eine infizierte und ausführbare Datei verleitet werden, wenn ihnen eine Liebesbotschaft oder der Gewinn eines Handys suggeriert wird.

2.3 Risiko-Potentiale im E-Commerce-Handel

Bisher wurde für jede Schwachstelle ein sehr konkretes und individuelles Gefährdungsschema für einen E-Commerce-Anbieter erstellet. Dieses Kapitel bietet nun ein kurzes Zwischenresümee, dass die verschiedenen Parteien bei einer E-Commerce-Transaktion betrachtet und differenziert darlegt wo und welche individuellen Bedrohungspotentiale vorliegen. Diese Bedrohungspotentiale werden im weiteren Verlauf der Diplomarbeit benötigt, da sie zu der späteren Gewichtung der Kriterien beitragen. Wie im realen Geschäftsleben, gibt es auch im Bereich des über das Internet abgewickelten Handels mehrere agierende Parteien. Auf der einen Seite gibt es die Anbieter von Dienstleistungen oder Gütern, die sogenannten Shopbetreiber. Auf der anderen Seite gibt es die Kunden, die auf die angebotenen Dienstleistungen oder Güter zugreifen. Zwischen diesen beiden Parteien sind die Provider angesiedelt. Provider stellen dem Anbieter von E-Commerce-Webseiten eine technische Infrastruktur zur Verfügung. Die E-Commerce-Anbieter greifen auf die Dienste und die Infrastruktur der Provider zurück, um ihre Webpräsenz aufzubauen. Theoretisch wäre es jedem E-Commerce-Anbieter möglich, seine Webpräsenz eigenständig aufzubauen. In den seltensten Fällen ist dies aber wirtschaftlich sinnvoll, da in der Regel die Provider über ein spezialisiertes Fachwissen und eine besser ausgebaute Infrastruktur verfügen. [JZ].

2.3.1 Kundensicht

Die Kunden einer E-Commerce-Transaktion können durch deren Nutzung in drei Bereichen Schaden erleiden. Der erste Gefährdungspunkt betrifft Datenschutzrichtlinien, der Zweite einen möglichen Schaden an ihrem Datenbestand und der dritte mögliche Schäden in finanzieller Hinsicht.

Punkte die den Datenschutz betreffen sind überall dort zu sehen, wo von einem Kunden persönliche Informationen angegeben werden. Das sind Authentifizierungsdaten und Informationen, die aus Cookie-Dateien entnommen werden können. Zu Beginn eines E-Commerce-Geschäftsvorganges muss von einem Kunden eine Registrierung auf der Webseite eines Shopbetreiber durchgeführt werden. Erst nach der Angabe von persönlichen Daten wie Name, Anschrift und Bankverbindung kann von einem Kunde einen Einkauf getätigt werden. Wichtig für den Kunden ist während dieses Vorganges die Frage, in wie weit von dem E-Commerce-Anbieter verantwortungsvoll mit seinen persönlichen Daten

umgegangen wird. Dies bezieht sich insbesondere darauf, in welcher Form seine Daten zum Shopbetreiber übertragen und wie diese Daten dort anschließend archiviert werden. Quasi in wie weit die Datenschutzrichtlinien eingehalten werden. Die einfachste Lösung für einen Kunden seine Registrierungsdaten zu übertragen, wäre das Versenden zum Shopbetreiber per E-Mail. Aufgrund von fehlendem Bedienkomforts und der grundsätzlich mangelhaften Sicherheit von E-Mails ist diese Variante kaum verbreitet. Eine zweite Möglichkeit besteht darin, dass der Kunde seine Daten in ein Webformular einträgt. Dieses Formular ist in der Bedienung für den Kunden sehr komfortabel, da es direkt auf der betreffenden Homepage des gekauften Artikels integriert werden kann. Nachdem der Kunde das Formular ausgefüllt hat, wird es mittels HTTP an den Shopbetreiber übertragen. Dieses Verfahren bietet ein hohes Maß an Komfort für den Kunden, ist aber aufgrund der unverschlüsselten HTTP-Übertragung nicht zu empfehlen. Die dritte Variante bietet die momentan praktikabelste und sicherste Lösung zur Datenübertragung. Die komplette Kommunikation zwischen dem Kunden und dem Anbieter wird mittels des HTTPS-Protokolls abgewickelt. Dieses Protokoll gewährleistet mit einer Verschlüsselungstiefe von 128Bit eine sichere Kommunikation. Für den Kunden ist diese Alternative folglich die wünschenswerteste. Hier kann er sicher sein, dass vertrauensvoll mit seinen Daten umgegangen wird. Es gestaltet sich problematisch für einen Kunden, an Informationen zu gelangen wie die Speicherung seiner Daten im Anschluss an die Transaktion von dem Betreiber des E-Shops vorgenommen wird. Die Übertragung der Daten zum Shopbetreiber kann der Kunde noch selbst aktiv absichern bzw. eine sichere Übertragung an ihren Merkmalen erkennen. Wie jedoch ein Shop mit seinen persönlichen Daten umgeht, nachdem diese zu ihm übertragen worden sind, ist für den Kunden kaum mehr nachvollziehbar. Hier stellt sich die Frage, wie sicher die Daten gelagert werden und wie der Shop mit den Daten in Hinsicht auf Customer Relationship Management (CRM) und generellen Data Mining Methoden umgeht.

Die Gefährdung für einen Kunden betreffend einem Schaden an seiner IT-Ausstattung, geht vor allen Dingen von den aktiven Inhalten in den verwendeten Webseiten des Shops aus. In Kapitel 2.1.3 wurden aktive Inhalte vorgestellt. Es handelt sich um kleine, in Webseiten integrierte Programme. Sie werden im Webbrowser des Besuchers der Internetseite ausgeführt, also lokal auf dessen Rechner. Aktive Inhalte werden von den Anbietern von Internetseiten angelegt, um dem Besucher einen Mehrdienst anzubieten. Gelingt es einem Angreifer, den Code der aktiven Inhalte auf den Webservern der Anbieter zu verändern, so können die Computer der Besucher der Webseite infiziert werden.
Die Gefährdung geht aber nicht nur davon aus, dass der Code von Angreifern verändert wird. Schon im Vorfeld können durch eine nachlässige Programmierung des Codes des aktiven Inhalts Fehler entstehen. Aus jedem fehlerhaftem Code kann wiederum eine Angriffsstelle entstehen.

Ein finanzieller Schaden ist in so weit möglich, dass durch eine Fahrlässigkeit des Shopbetreibers die Daten eines Kunden entwendet werden können. Gelangt ein Angreifer über diesen Weg an Bank- der Kreditkartendaten kann er mit der gestohlenen Identität Waren einkaufen.

2.3.2 Anbietersicht

Technische Sicherheit bedeutet für einen Anbieter zuallererst, dass sein Web-Shop ausfallsicher sein muss. Er kann nur Geld erwirtschaften, wenn die Kunden seinen Webauftritt nutzen können. Ein Anbieter von E-Commerce-Webseiten hat desweiteren die Problematik, im Gegensatz zu dem Endkunden, dass bei der Fällung von Entscheidungen, stets wirtschaftlich gehandelt werden sollte. So muss er für jedes technische Risiko zuerst die Kosten zu deren Behebung und dem daraus folgenden Nutzen gegenüberstellen. Mühsam für einen E-Commerce-Anbieter gestaltet sich die Erfassung aller Kosten, die zur Erreichung der IT-Sicherheit anfallen. Die Erfassung der Kosen wird erschwert durch den Umstand, dass sich die Sicherheitsanforderungen kontinuierlich ändern. Das macht eine andauernde Nachbesserung nötig. Zusätzlich können im Laufe der Zeit neue, bisher unbekannte Bedrohungen auftreten. Generell sind die möglichen Schadenszenarien, die ein Anbieter von E-Commerce-Webseiten ereilen können, von wesentlich größerer Tragweite als die möglichen Szenarien eines Kunden.

Wurde bereits versucht auf dem Web-Server...	ja
Seiten zu verändern (Vandalismus)	22%
Daten auszuspionieren	17%
Daten zu löschen	9%
Daten in betrügerischer Absicht zu manipulieren	7%
Angebote lahmzulegen (Denial of Service)	27%
nichts von alledem	56%

Basis: 124 Antworten

Abbildung 17: Angriffe auf Web-Server [KES]

Dieses hohe Risiko kann aus der KES-Studie kann entnommen werden. Nahezu die Hälfte der befragten Unternehmen sind schon das Opfer von Angriffen oder Angriffsversuchen gewesen. Die Bandbreite der Angriffe reicht von weniger gefährlichem Vandalismus, Defacement-Attacken bis hin zu dem "Abschießen" von ganzen Webangeboten. Dies würde einen E-Commerce-Shop besonders hart treffen.

Die Szenarien können in ihren Auswirkungen bezüglich eines E-Shops in acht Kriterien eingeordnet werden:

- **Imageverlust**

Ein E-Commerce-Anbieter ist durch SQL-Angriffe, Cross-Site-Scripting oder andere Attacken fortwährend der Gefahr ausgeliefert, dass seine Webpräsenz verändert werden kann. Durch schmähende oder verleumderische Verfremdung seiner Webpräsenz kann ihm ein Imageschaden zugefügt werden. Dabei wird zwischen ungerichtetem Vandalismus und Attacken mit betrügerischer Absicht unterschieden. Vandalismus zielt darauf ab, die Seite verleumderisch zu verfremden. Betrügerische Manipulationen zielen dagegen darauf ab, dem Unternehmen einen direkten wirtschaftlichen Schaden zu zufügen.

Das Image, das ein Anbieter in der Öffentlichkeit besitzt, ist ein wichtiges Kriterium anhand dessen Kunden ihre Kaufentscheidung fällen. Ein Anbieter von E-Commerce-Webseiten kann stark an Image verlieren, wenn bekannt wird, dass er technische Probleme hat. Das können sowohl technische Pannen sein, wie auch ein unsachgemäßer Umgang mit Daten.

- **Verstöße gegen Gesetze, Vorschriften oder Verträge**

Durch den unsachgemäßen Umgang mit Daten oder technischen Mitteln kann gegen bestehende Gesetze verstoßen werden. So kann zum Beispiel ein E-Shop gegen das Datenschutzgesetz oder das Urheberrecht verstoßen. Gleiches gilt auch im Umgang mit Verträgen. Regressforderungen von Kunden können auftreten, wenn der Shopbetreiber aufgrund von technischem Fehlverhalten gegen abgeschlossene Verträge verstößt.

Kriterien zur Risikobewertung	sehr wichtig	wichtig	un-wichtig	Vergl.-wert	Vgl.-W. 2002
Verstöße gegen Gesetze, Vorschriften oder Verträge	48%	44%	8%	1,40	1,47
Imageverlust	50%	36%	15%	1,35	1,51
Schaden bei Dritten/ Haftungsansprüche	43%	42%	16%	1,27	1,11
direkter finanzieller Schaden durch Manipulationen an finanzwirksamen Informationen	38%	49%	12%	1,26	1,36
Verzögerung von Arbeitsabläufen	33%	55%	12%	1,21	1,35
indirekte finanzielle Verluste (z. B. Auftragsverlust)	36%	42%	22%	1,14	0,98
direkter finanzieller Schaden an Hardware u. Ä.	12%	50%	37%	0,75	0,97
Verstöße gegen interne Regelungen	9%	54%	37%	0,72	0,85

Basis: Ø 135 Antworten

Abbildung 18: Kriterien zur Risikobewertung [KES]

- **Direkter wirtschaftlicher Schaden**

Durch Datenspionage kann einem E-Commerce-Unternehmen ein wirtschaftlicher Schaden zugefügt werden. Es können interne Informationen, wie Einkaufspreise von Waren, aber auch interne Wirtschaftsdaten, wie Quartalszahlen, die Liquidität und ähnliches ausspioniert werden. Desweiteren wird durch die Manipulation finanzwirksamer Informationen einem E-Commerce-Unternehmen ein direkter wirtschaftlicher Schaden zugefügt. Manipulationen können von böswilligen externen Angreifern, aber auch von internen Mitarbeitern ausgeführt werden. Hier heißt ab zu schätzen, dass auf der einen Seite an kritischen Stellen durch den richtigen Einsatz von technischen Mitteln dieses Risiko minimiert wird. Auf der anderen Seite kann, gerade wenn diese Mittel nicht fachgerecht eingesetzt werden, einem böswilligen Angreifer erst ein Angriff ermöglicht werden. Durch den unsachgemäßen Einsatz von technischen Mitteln kann es sogar vorkommen, dass eigene interne Mitarbeiter unwissentlich einen Schaden anrichten.

- **Haftungsansprüche**

Entsteht einem Kunden oder einem Vertragspartner ein Schaden oder ein Geschäftausfall durch die Nutzung des Webangebotes, so kann der Anbieter, wenn er fahrlässig gehandelt hat, hierfür haftbar gemacht werden. Um dies auszuschließen, ist ebenfalls auf eine technisch einwandfreie Infrastruktur zu achten. Gleiches gilt, wenn der Anbieter einem Vertrag nicht nachkommt. Tritt das Szenario ein, dass der Anbieter zum Beispiel aufgrund eines Ausfalls der Datenbank einem Kunden keine Ware liefern. Dann kann dieser in Folge nicht produzieren und es sind Haftungsansprüche zu befürchten wenn der Anbieter grobfahrlässig gehandelt hat.

- **Direkter finanzieller Schaden an der IT-Ausstattung**

Parallel zu finanziellen Schäden durch Manipulationen an Datenbeständen besteht für einen E-Commerce-Anbieter die Gefahr eines direkten finanziellen Schadens an der Hardware oder anderen technischen Einrichtungen. Fällt zum Beispiel aufgrund eines DoS-Angriffes ein Server wegen Überhitzung dauerhaft aus, bedeutet dies für das Unternehmen ein direkter finanzieller Schaden.

- **Verzögerung von Arbeitsabläufen**

Wird das IT-System eines Unternehmens durch eine Wurmattacke so stark ausgelastet, dass ein wirtschaftliches Arbeiten der Mitarbeitern nicht mehr möglich ist, kann es zu Verzögerungen von Arbeitsabläufen kommen. Ein anderes Beispiel wäre, wenn die Datenbank aufgrund eines Ausfalls mittels eines Backup wiederhergestellt werden muss und

es während dieser Zeit zu Verzögerungen kommt. Generell lässt sich festhalten, dass viele technische Schwachstellen Auswirkungen auf Produktionsabläufe haben können.

- **Indirekte finanzielle Verluste**

Verliert ein E-Commerce-Unternehmen aufgrund von technischen Ausfällen einen möglichen Auftrag eines Kunden, so bedeutet dies ein indirekter finanzieller Verlust. Finanzielle Verluste gehen ebenfalls indirekt aus einem Imageverlust und anderen, die Produktion verzögernden Ereignissen, hervor.

- **Verstöße gegen interne Regelungen**

Durch den unsachgemäßen Einsatz von IT-Mitteln können Mitarbeiter gegen interne Richtlinien verstoßen. Das kann ein nicht alle 30 Tage aktualisiertes Passwort, ein unsachgemäßer Umgang mit Daten, oder die eigenmächtige Installation von Software im System sein.

2.3.3 Providersicht

Das Risikopotential eines Providers ist stark von seinen Kunden, in diesem Fall den E-Commerce-Anbietern, und deren Verhalten abhängig. Generell verfügt ein Provider über bessere Kenntnisse, als die Anbieter, zur Absicherung von Webauftritten und bietet diese über Dienste seinen Kunden an. Problematisch wird es, wenn ein Kunde diese Möglichkeiten nicht oder nur unzureichend ausnutzt. So kann ein Kunde, der Opfer eines DoS-Angriff wurde, das ganze Netzwerk seines Providers in Mitleidenschaft ziehen. In der Praxis ist es üblich, dass die Webpräsenzen mehrerer Kunden, auf einem Webserver liegen. Wird dieser Webserver durch einen DoS-Angriff auf einen der Kunden ausgelastet, kann es passieren, dass auch die Webseiten der anderen Kunden nur noch eingeschränkt oder gar nicht mehr erreichbar sind. Ein Provider sollte folglich dafür sorgen, dass seine Kunden sich an Sicherheitsrichtlinien halten. Da dies von dem Provider nur schwer überprüft werden kann, werden derartige Sicherheitsüberprüfungen in der Regel durch Mehrwertdienste vom Provider angeboten.

3. Diskussion bestehender Kriterienkataloge

In den vorangegangen Kapiteln konnten die technischen Schwachstellen von E-Commerce-Auftritten aufgezeigt werden. Zusätzlich wurden den beteiligten Parteien jeweilige Bedrohungspotentiale zugeordnet. Der nächste Schritt bestand darin, aus diesen technischen Risiken einen geeigneten und automatisierbaren Kriterienkatalog zu erstellen. Diese Aufgabe wird durch die Bedrohungspotentiale der einzelnen Schwachstellen unterstützt. Die Schwachstellen ermöglichen eine sinnvolle Gewichtung und Einordnung der Kriterien. Als Ausgangs- und Vergleichsbasis für den zu erarbeitenden Kriterienkatalog können bestehende Kriterienkataloge herangezogen werden.

Kriterienkataloge werden vorwiegend von Unternehmen genutzt, um aussagekräftige Resultate über die Sicherheit ihrer IT-Systeme zu erwerben. Zusätzlich können Betreiber von Webseiten Zertifikate erlangen, bzw. auch selber als Zertifizierer agieren. Der zu prüfende Kunde möchte die Kriterien des Kataloges erfüllen und dadurch die Voraussetzungen zur Erlangung eines Zertifikates erreichen. Von diesem Zertifikat erhofft er sich einen Imagegewinn, verbunden mit einem wirtschaftlichen Vorteil. In der Praxis haben sich mehrere Kriterienkataloge etabliert. Herausgeber sind verschiedenste kommerzielle Firmen oder Organisationen. Zudem wird zwischen offiziellen und inofiziellen Stellen unterschieden.

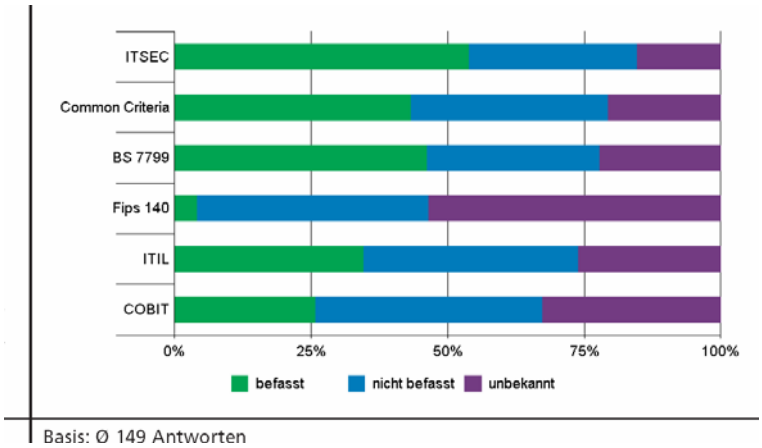

Abbildung 19: Bekanntheitsgrad von Kriterienkatalogen [KES]

3.1 Orange Book (TCSEC)

Als einer der ersten Kriterienkataloge wurde Anfang der 80er Jahre die "Trusted Computer Systems Evaluation Criteria" (TCSEC) in den USA entwickelt. Herausgeber des Standards war die US-Regierung. TCSEC ermöglicht die Bewertung und Zertifizierung der Sicherheit von Betriebssystemen, Netzwerkkomponenten und einzelner Anwendungen. Der TCSEC-Katalog, auch Orange Book genannt, stuft geprüfte Systeme grob in vier verschiedene Sicherheitsklassen ein. Die Klasse A bildet die höchste erreichbare Sicherheitsklasse und die Klasse D die niedrigste erreichbare Klasse. Die Prüfkriterien waren äußerst monoton auf die Vertraulichkeit der Datei und Benutzerverwaltung ausgerichtet.
Aufgrund seiner monotonen Ausrichtung auf das einzige Kriterium der Vertraulichkeit der Datenhandhabung und seinem inzwischen fortgeschrittenen Alter ist das Orange Book als Quelle für den zu erarbeitenden Kriterienkatalog eher ungeeignet. Für E-Commerce-Seiten bietet es nur die Möglichkeit die interne Handhabung bei der Datenhaltung (Speicherung, Backup, ...) zu überprüfen. Eine Überprüfung der Sicherheit der eigentlichen Webseite und deren Anwendungen ist nicht möglich. Gleichzeitig gewährt es jedoch aufgrund dieser einfachen Struktur einen guten ersten Einblick darauf, wie ein Kriterienkatalog aufgebaut werden kann [BSI_IT].

TCSEC	4 Sicherheitsklassen (A-B-C-D) Abstufungen nach Vertraulichkeit bei der Datei- & Benutzerverwaltung

Tabelle 3: TCSEC Klassen

3.2 White Book (ITSEC)

Zehn Jahre später, Anfang der 90er Jahre, wurde in Europa der "Information Technology Security Evaluation Criteria" (ITSEC) Standard veröffentlicht. Die deutsch Übersetzung des Titels bedeutet in etwa: Kriterien für die Bewertung der Sicherheit von Informationstechnologie und hat das Ziel IT-Systeme auf deren Vertraulichkeit, Integrität und Verfügbarkeit zu überprüfen. Inhaltlich ist das White Book stark an den älteren deutschen ITSK-Standard angelehnt. Geprüft werden die Art der Datenspeicherung und die verwendete Computerarchitektur. Verglichen mit dem Orange Book ist das White Book bei der Auswahl der Kriterien eine Generation fortschrittlicher und bietet eine differenziertere Einteilung der getesteten Systeme in Klassen.
Es existieren zwei Hauptklassen. Jeder dieser beiden Klassen wird wiederum in speziellere Ebenen unterteilt. Die erste Hauptklasse deckt Sicherheitsgrundfunktionen über 10

unterschiedlichen Ebenen ab. Die einzelnen Ebenen stehen für die jeweils umgesetzten Sicherheitsanforderungen. Die zweite Hauptklasse beurteilt die Qualität und die Vertrauenswürdigkeit. Die Ebenen gibt zwei Kriterien wieder. Zum einem in wie weit die korrekte Funktionsweise des verwendeten IT-Systems erreict wurde und gleichzeit die Wirksamkeit der eingesetzten Mechanismen. Für den zu erarbeitenden Kriterienkatalog ist der ITSEC-Standard inzwischen aufgrund seiner eingeschränkten Kriterienauswahl ungeeignet. Dieser wurde in der Praxis durch den Common Criteria Katalog ersetzt. Aufgrund dessen ist, wenn überhaupt, der Common Criteria Katalog als Quelle und Ausgangsbasis für den zu erarbeitenden Kriterienkatalog geeignet [LH_A].

ITSEC Klasse 1	Sicherheitsanforderungen (10 Unterklassen mit den jeweils umgesetzten Sicherheitsgrundfunktionen)
ITSEC Klasse 2	Qualität & Vertrauenswürdigkeit (Korrektheit & Funktionsweise des Systems, 6 Unterklassen)

Tabelle 4: ITSEC Klassen

Systeme, die nach den ITSEC-Kriterien auf die Richtigkeit ihrer Implementierung bewertet wurden, werden in sechs Klassen unterteilt:

E0	Unwirksam
E1	Informelle Spezifikation der Architektur, Funktionstest, gezielte Angriffe
E2	zusätzlich informelle Beschreibung des Feinentwurfs (Detailspezifikation)
E3	Analyse des Quellcodes bzw. des Hardwarelayouts
E4	Formales Sicherheitsmodell, semiformale Detailspezifikation
E5	Detailspezifikation muss nachvollziehbar auf Quellcode abbildbar sein
E6	zusätzlich formale Spezifikation und Analyse der Architektur (Verifikation)

Tabelle 5: ITSEC Implementierungsklassen

3.3 Common Criteria

Der vollständige Titel lautet "Common Criteria for Information Technology Security Evaluation". Common Criteria löste sowohl den veralteten europäischen ITSEC-, als auch den amerikanischen TCSEC-Standard ab. Übersetzt man den Titel ins Deutsche würde die Bezeichnung in etwa wie folgt lauten: Gemeinsame Kriterien für die Bewertung der Sicherheit von Informationstechnologie. In der Literatur werden meist die kürzeren Formen "Common Criteria" oder "CC" verwendet. Das Merkmal, dass den Common Criteria-Standard von anderen Standards hervorhebt ist seine Internationalität. An dem Common Criteria-Projekt sind bedeutende Industrieländer wie Deutschland, Frankreich,

Großbritannien, Kanada, die Niederlande und die USA beteiligt. Deutschland wird durch das Bundesamtes für Sicherheit in der Informationstechnik vertreten. Die internationalen Wurzeln des Common Criteria-Projektes vermeiden das Komponenten oder ganze Systeme in verschiedenen Ländern mehrfach zertifiziert werden müssen.

Der Standard besteht in seinen Grundzügen aus einer Vielzahl von Kriterien. Diese normierten Kriterien ermöglichen eine international vergleichbare Bewertung und Zertifizierung der Sicherheit von Computersystemen im Hinblick auf Datensicherheit und Datenschutz. Die aktuelle Version 3.1 des Common Criteria wurde im September 2006 von der Staatengemeinschaft im internationalen Common Criteria Anerkennungsabkommen verabschiedet. Der aktuelle Common Criteria Standard besteht aus drei Kapiteln:

Das erste Kapitel beginnt mit einer Einführung in der die Grundlagen der IT-Sicherheitsevaluation und der allgemeine Geltungsbereich der CC erläutert werden. In den Anhängen werden die grundsätzlichen Sicherheitsanforderungen eines Gegenstandes, auch Target Of Evaluation genannt (TOE), vorgestellt. Ferner beinhaltet das Kapitel verschiedene Sicherheitsmodelle für unterschiedliche Produktklassen. Jedes dieser Modelle beschreibt Schutzprofile, die für gewisse Klassen von Produkten abgestimmt wurden. Für besonders Schutz bedürftige Produkte, sogenannte "Security Targets" (ST) oder ähnliche Sonderfälle, enthält das erste Kapitel zusätzlich spezialisierte Sicherheitsvorgaben.

Das zweite Kapitel listet in einem umfangreichen Katalog die funktionalen Sicherheitsanforderungen an die verwendeten Bausteine einer IT-Infrastruktur auf ("Functional Requirements"). Das Kapitel stellt ein empfohlenes Angebot für die Beschreibung der Funktionalität eines Produktes bzw. Systems dar, von dem jedoch in begründeten Fällen abgewichen werden kann. Zusätzlich werden Zusammenhänge zwischen Bedrohungen, Sicherheitszielen und funktionalen Anforderungen aufgezeigt. Es ermöglicht eine Bewertung der Funktionalität und des Funktionsumfang des betrachteten Systems [BSI_CC].

Das dritte Kapitel des Common Criteria Kataloges listet die einzelnen Anforderungen an die Vertrauenswürdigkeit der eingesetzten Bausteine auf ("Assurance Requirements"). Durch die Umsetzung der Anforderungen wird parallel die Qualität des betrachteten Systems wiedergegeben. Die Qualität wird von zwei wesentlichen Indikatoren bestimmt. Zum einem die Wirksamkeit der verwendeten Methoden und zum anderem die der Korrektheit der Implementierung. Ist die Qualität des Systems hoch, kann ihm eine hohe Vertrauenswürdigkeit zugesprochen werden. Ist die Qualität gering, kann das System nur als eingeschränkt vertrauenswürdig bezeichnet werden. Ein Evaluationsergebnis wird in den

Common Criteria immer auf Basis einer Vertrauenswürdigkeitsstufe (EAL) dargestellt, eventuell ergänzt durch weitere Anforderungen.

EAL1	funktionell getestet
EAL2	strukturell getestet
EAL3	methodisch getestet und überprüft
EAL4	methodisch entwickelt, getestet und durchgesehen
EAL5	semiformal entworfen und getestet
EAL6	semiformal verifizierter Entwurf und getestet
EAL7	formal verifizierter Entwurf und getestet

Tabelle 6: Common Criteria EAL-Stufen

CC Teil 1	Grundsätzliche Sicherheitsanforderungen von/an Systembausteine(n)
CC Teil 2	Funktionale Sicherheitsanforderungen (Bewertung der Funktionalität & des Funktionsumfanges)
CC Teil 3	Qualitative Bewertung des Systems (Methoden & Implementierung) -> Vertrauen (hoch – niedrig) - 7 EAL Stufen

Tabelle 7: Common Criteria Klassen

Basis für die Vertrauenswürdigkeitsbewertung sind im CC-Katalog sieben Vertrauenswürdigkeitsstufen. In ITSEC wird das Vertrauen in das System in sechs Stufen die unterteilt. Die CC Stufen entsprechen inhaltlich in etwa den E-Stufen der ITSEC.

CC-EAL2 entspricht ITSEC-E1

...

CC-EAL7 entspricht ITCEC-E6

EAL1 ist in der Hierarchie unterhalb von E1 einzuordnen und wurde entwickelt, um den Zugang zur Evaluation zu erleichtern[BSI_CC].
Neben dem eigentlichen Kriterienkatalog gibt es zusätzlich noch eine Evaluationsmethodologie ("Common Methodology for Information Security Evaluation CEM"), durch die einem Anwender die Umsetzung des Kriterienkataloges erleichtert werden [Bu_CC].
Werden die Common Criteria und ITSEC verglichen, so erscheinen beide in ihrem Aufbau in vielen Punkten übereinzustimmen. Beide Kriterienkataloge erlauben eine getrennte Prüfung und Bewertung der Funktionalität und der Vertrauenswürdigkeit von IT-Systemen. Aufgrund ihrer nahen Verwandtschaft ist dies auch verständlich. Die CC sind jedoch erheblich umfangreicher. So bieten sie mit etwa 150 detaillierten Funktionalitäts-komponenten konkretere Funktionalitätskriterien als die ITSEC. Beiden gleich ist die Möglichkeit, dass weitere sicherheitsspezifische Funktionen evaluiert werden können, die nicht explizit in den Kriterien aufgelistet sind.

3.4 IT-Infrastructure-Library (ITIL)

Die IT-Infrastruktur-Library stammt aus Großbritannien. Sie beschreibt die für den Betrieb einer IT-Infrastruktur notwendigen Prozesse. Die vorgestellten Prozesse orientieren sich weniger an der verwendeten Technik, sondern haben ihren Schwerpunkt in der Betrachtung der durch den IT-Betrieb erbrachten Dienstleistungen. ITIL ist folglich kein technischer Kriterienkatalog im eigentlichen Sinne. Vielmehr ist ITIL eine Literatursammlung anhand derer ein IT-Service-Management aufgebaut werden kann.

Zum derzeitigen Zeitpunkt hat sich in IT-Betrieben der Einsatz von ITIL weit verbreitet. Die starke Verbreitung von ITIL führte dazu, dass sich eine gemeinsame Terminologie für das IT-Service-Management herausbilden konnte. Dies hat den Vorteil, dass die Kommunikation zwischen IT-Abteilungen, sowohl innerhalb eines Unternehmens als auch unternehmensübergreifend, vereinfacht wurde. Die starke Verbreitung von ITIL zeigt, dass sich unter den Verantwortlichen ein starkes Bewusstsein für das Thema entwickelt hat. ITIL hat grundsätzlich nicht den Anspruch, eine endgültige und umfassende Standardisierung zu schaffen. Die Literatursammlung verfolgt vielmehr einen sogenannten Best-Practice-Ansatz. Dabei werden in der Praxis erfolgreiche Modelle und Organisationsformen so beschrieben, dass jede Organisation sie beliebig adaptieren und auf ihre Bedürfnisse zuschneiden kann. ITIL beschreibt nicht wie etwas getan werden muss, sondern nur was getan werden sollte. Preis dieser Flexibilität ist allerdings die Ungenauigkeit der Beschreibungen, so dass bei der Umsetzung konkretisierende Sekundärliteratur oder das Wissen erfahrener Berater hinzugezogen werden muss [BSI_ITIL].

Die IT-Infrastruktur-Library in der zweiten Version aus dem Jahr 2003, besteht aus sieben Kernpublikationen und einem ergänzenden Teil. Für den zu erarbeitenden Kriterienkatalog ist das Kapitel 3, Security Management, von besonderem Interesse:

1. Service Support
2. Service Delivery
3. **Security Management**
4. ICT (Information and Communications Technology) Infrastructure Management
5. Application Management
6. Planning to Implement Service Management
7. The Business Perspective
8. Software Asset Management

Wird die IT-Infrastruktur-Library in Zusammenhang mit dieser Diplomarbeit und dem Ziel einen Kriterienkatalog zu erstellen, der die technische Sicherheit einer E-Commerce-

Webseite automatisiert nach festgelegten Kriterien untersucht, ist die dritte Kernpublikation, Security Management, von besonderem Interesse. Das Kapitel besteht aus einer Textsammlung, die sich mit der Einführung und Durchsetzung eines definierten Sicherheitsniveaus für ein IT-System beschäftigt. Dabei wird ausführlich auf die Teilgebiete Vertraulichkeit, Integrität und Verfügbarkeit eingegangen. Um sowohl die internen, als auch die kundenspezifischen Wünsche des benötigen Sicherheitslevels zu ermitteln, ist eine Risikoanalyse notwendig. Der minimale, interne Sicherheitsanspruch wird dabei als IT-Grundschutz bezeichnet. Darüberhinausgehende Ansprüche des Kunden an Sicherheitsbedürfnisse müssen individuell erarbeitet werden. Aufgrund der automatisierten Abarbeitung von vorher festgelegten Kriterien, wird der Kriterienkatalog im Vorfeld keine individuelle Risikoanalyse eines Unternehmens durchführen können. Somit ist ein Einbezug der ITIL-Kriterien in den zu erarbeitenden Kriterienkatalog nur eingeschränkt realisierbar. Miteinbezogen werden kann der interne, minimale Sicherheitsanspruch, der als IT-Grundschutz bezeichnet wird. Dieser IT-Grundschutz wird jedoch von dem im nächsten Unterkapitel vorgestellten BSI-Grundschutz-Katalog bzw. dem darauf folgenden BSI-Penetrationstest effektiver gewährleistet.

ITIL (Security Management)	Vertraulichkeit, Integrität und Verfügbarkeit

Tabelle 8: ITIL Klassen

3.5 BSI IT-Grundschutz-Kataloge

In den IT-Grundschutz-Katalogen werden standardisierte Sicherheitsmaßnahmen für typische IT-Systeme vorgestellt. Das Ziel dieser IT-Grundschutz-Empfehlungen ist es, durch geeignete Anwendung von organisatorischen, personellen, infrastrukturellen und technischen Sicherheitsmaßnahmen ein Sicherheitsniveau für IT-Systeme zu erreichen, das für den normalen Schutzbedarf angemessen und ausreichend ist und als Basis für hochschutzbedürftige IT-Systeme und IT-Anwendungen dienen kann. Die besondere Stärke ist dabei die strikte Systematik des Vorgehens. Mit dem Grundschutz-Tool (GSTOOL), steht zudem eine Software zur Verfügung, die den IT-Sicherheitsbeauftragten bei der Anwendung der Maßnahmen aus den IT-Grundschutzkatalogen umfassend unterstützt.

Das IT-Grundschutzhandbuch setzt sich im Baukastenprinzip aus einer Einleitung und drei umfangreichen Katalogen zusammen. Da sich das Handbuch ebenso an ungeübte Nutzer richtet, bietet die Einleitung Herangehensweisen an den IT-Grundschutz, Begriffs- und Rollendefinitionen sowie einen Glossar. Die einzelnen Bausteine spiegeln typische Bereiche des IT-Einsatzes, wie beispielsweise Client-Server-Netze, bauliche Einrichtungen oder, Kommunikations- und Applikationskomponenten wieder. In jedem Baustein wird zunächst

die zu erwartende Gefährdungslage beschrieben. Um dem Anwender eine Gewichtung der Bausteine zu ermöglichen, werden sowohl die typischen Gefährdungen als auch pauschalisierte Eintrittswahrscheinlichkeiten berücksichtigt. Hieraus kann der Anwender eine individuelle Gefährdungslage ableiten. Diese Gefährdung ist die Grundlage, auf der ein spezifisches Maßnahmenbündel aus den Bereichen Infrastruktur, Personal, Organisation, Hard- und Software, Kommunikation und Notfallvorsorge generiert wird [GSK_B2].

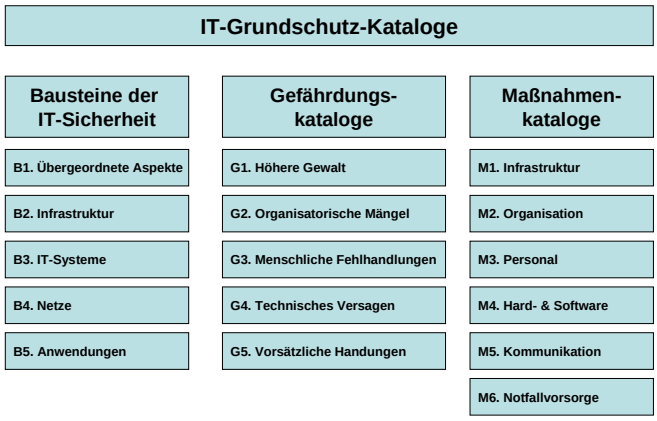

Abbildung 20: Aufbau IT-Grundschutz-Katalog

Jeder der drei Kataloge des IT-Grundschutz-Kataloges setzt sich aus einer Sammlung von Dokumenten zusammen. Diese Dokumente beschäftigt sich mit spezifischen Themen.
Ein Dokument, dass einem der drei Kataloge angehört, wird durch ein individuelles Kürzel gekennzeichnet. Zu diesem Zweck wird die folgende Strukturierung verwendet: Zunächst wird in einem Dokument die Kataloggruppe genannt. B steht für Baustein, G für Gefährdung und M für Maßnahme. Danach folgt die Nummer der Schicht, die dieses Katalogelement in seinem Katalog betrifft, anschließend die fortlaufende Nummer innerhalb der Schicht. Nachdem der prinzipielle Aufbau des IT-Grundschutz-Kataloges vorgestellt wurde, werden im Anschluss die einzelnen Inhalte der Kataloge betrachtet.

Bausteine

Das zentrale Element des IT-Grundschutz-Kataloges sind die Bausteine. Der Aufbau der einzelnen Bausteine ist im Prinzip ähnlich und folgt, wie auch die weiteren Kataloge, einem Schichtenmodell. Jeder Baustein beginnt mit einer kurzen Beschreibung der betrachteten Komponente, der Vorgehensweise, sowie einem Überblick über die Gefährdungslage und die zugehörige Maßnahmenempfehlungen. Durch die Einteilung in Schichten lassen sich die von der jeweiligen Schicht betroffenen Personengruppen klar eingrenzen. Die erste Schicht spricht das Management an (Übergeordnete Elemente). Haustechniker werden von der zweiten Schicht angesprochen (Infrastruktur). Die dritte Schicht wird von Systemadministratoren abgedeckt (IT-System-Schicht). Die vierte Schicht fällt in den

Aufgabenbereich der Netzwerkadministratoren (Netzwerkschicht) und die fünfte in den der Anwendungsadministratoren und der IT-Nutzer (Anwendungsschicht). Die Maßnahmen oder Gefährdungen, die in einem Baustein vorgestellt werden, können ebenso für andere, zum Teil auch komplett unterschiedliche Bausteine relevant sein. So entsteht eine Vernetzung der einzelnen Komponenten des Grundschutzhandbuchs.

```
Baustein(Komponente) → Vorgehensweise → Gefährdungslage → Maßnahmen
```

Gefährdungskataloge

In diesem Bereich enthält der IT-Grundschutz-Katalog die ausführlichen Beschreibungen der Gefährdungen, die in den einzelnen Bausteinen als Gefährdungslage genannt wurden. Ebenfalls, wie der bereits kennen gelernte Aufbau der Bausteine, folgt der Gefährdungskatalog dem allgemeinen Aufbau nach Schichten. Zur Erreichung eines Grundschutzes in einem IT-System ist nach Aussagen des BSI, dass in diesen Katalogen zusammengestelltes Wissen nicht unbedingt notwendig. Es fördert jedoch beim Leser das Verständnis für die Maßnahme sowie deren Wachsamkeit.
Die einzelne Gefahrenquelle ist in einem kurzen Text beschrieben. Anschließend werden Beispiele für Schadensfälle die durch diese Gefahrenquelle ausgelöst werden können, gegeben.

```
Gefährdung → Schadensszenario
```

Maßnahmenkataloge

In den Maßnahmenkatalogen werden die in den Bausteinen der IT-Grundschutz-Kataloge zitierten IT-Sicherheitsmaßnahmen ausführlich beschrieben. Hierfür werden die Maßnahmen in Kataloge zusammengefasst. So werden Maßnahmen, die für mehrere System-Komponenten angemessen sind, nur einmal zentral beschrieben.
In der jeweiligen Maßnahmenbeschreibung werden vorab Verantwortliche für die Einleitung und die Durchführung der Maßnahme genannt. Es folgt eine ausführliche Beschreibung der Maßnahme. Nach Abschluss der Durchführung stehen Kontrollfragen zur Verfügung, die die korrekte Umsetzung der Maßnahmen überprüfen.
Bei der Umsetzung der Maßnahme sollte zunächst überprüft werden, ob eine Anpassung dieser auf den jeweiligen Betrieb notwendig ist. Abschließend rät das BSI, Anpassungen zu dokumentieren, damit diese zu einem späteren Zeitpunkt nachvollzogen werden können. [GSK_B2].

```
Maßnahmen → Verantwortliche → Durchführung → Kontrolle  →
Dokumentation
```

GSTOOL

Das BSI stellt mit dem IT-Grundschutz Tool (GSTOOL) eine Software bereit, die Anwender bei der Erstellung, Verwaltung und Fortschreibung von IT-Sicherheitskonzepten entsprechend dem IT-Grundschutz effizient unterstützt. Das Tool erfasst im ersten Schritt sämtliche benötigten Informationen über das verwendete IT-System und das Netzwerk, sowie die verwendeten Anwendungen. Im zweiten Schritt stellt es dem Anwender ein umfangreiches Berichtssystem zur Verfügung, mittels dessen der Anwender strukturierte Auswertungen über sämtliche erfassten Daten durchführen und diese auf Papier oder elektronisch ausgeben kann. Das Berichtsystem umfasst eine Modellierung und Schichtung nach dem IT-Grundschutz-Katalog. Des Weiteren umfasst es eine Kostenauswertung und eine Schutzbedarfsanalyse.

Als Fazit lässt sich der IT-Grundschutzkatalog als eine geeignetes Medium bezeichnen, mit dessen Hilfe ein IT-System "Step by Step" abgesichert werden kann. Durch seine modular aufgebaute Struktur ist der Katalog in der Benutzung einfach gehalten und richtet sich ebenso an ungeübte Anwender. Sein enormer Umfang macht jedoch eine zeitlich längere Einarbeitung notwendig. Dafür besitzt er aber gerade in seinem Umfang seinen größten Vorteil, da er nahezu alle Module eines IT-Systems abdeckt. Durch den Umstand, dass der Katalog die Bausteine eines IT-System und die zugehörigen Bedrohungsszenarien zueinander in Beziehung setzt, lassen sich spezifische Prüfkriterien extrahieren. Diese können anschließend in den zu erarbeitenden Kriterienkatalog aufgenommen werden. Die Schwierigkeit liegt hier bei einer angemessenen Auswahl von Kriterien. Bei der Gestaltung dieser Arbeit findet der Katalog darin Verwendung das er die ausgewählten Kriterien bestätigt und bei der Erstellung von Schadenszenarien hilft

3.5.1 BSI Penetrationstest

“Unter einem Penetrationstest wird in der Informationstechnik die kontrollierte Prüfung der Sicherheit eines Netzwerk- oder Softwaresystems mit eben jenen Mitteln und Methoden, die ein Angreifer anwenden würde, um unautorisiert in das System einzudringen (Penetration) verstanden“ [WIK].

Ein Penetrationstest ermittelt folglich, wie Anfällig ein zu testendes System gegen böswillige Angriffe ist, und wie weit die eingesetzten Sicherheitsmaßnahmen funktionieren. Die hierbei identifizierten Schwachstellen können dann im Anschluss durch entsprechende Maßnahmen behoben werden. Besonders gefährdet sind grundsätzlich Systeme die über eine Verbindung zu öffentlichen Netzwerken verfügen. Der BSI Penetrationstest kann des weiteren als eine reale Anwendung des IT-Grundschutzkataloges verstanden werden. Das

Ziel des Kataloges ist es ein IT-System unter Zuhilfenahme der Grundschutz Bausteine, Gefährdungs- und Maßnahmen-kataloge auf seine Sicherheit hin zu überprüfen.

Beim Entwurf eines Penetrationstestes muss sich der Designer zunächst die Frage stellen welche Bedrohungsszenarien existieren. Die Bedrohungsszenarien setzen sich aus den möglichen Angreifern und deren Vorgehensweisen zusammen. Die klassischen Vorgehensweisen für einen Angriff auf ein IT-System finden entweder über das Netzwerk, über Social-Engeneering oder über die Umgehung der physischen Sicherheitsmaßnahmen statt. Im nächsten Schritt müssen die Kriterien erkannt werden die technisch überprüft werden können. Nicht alles kann mit technischen Mitteln überprüft werden. So ist der technisch versierte Umgang der Firmenmitarbeiter nur mittels Befragungen zu überprüfen. Die prüfbaren IT-Sicherheitsmaßnahmen können in logische (z.B. Passwörter) und in physische (z.B. Zutrittskontrollsysteme) Kriterien unterteilt werden. Parallel zu den möglichen Angriffspunkten sollten mögliche Sicherheitsmaßnahmen erarbeitet werden.

Mit einem Penetrationstest kann nur der aktuelle Sicherheitszustand zum Zeitpunkt des Testes ermitteln werden. Da sich die Techniken der potenziellen Angreifer schnell weiterentwickeln und beinahe täglich neue Schwachstellen in IT-Systemen gemeldet werden, kann keine Aussage über das Sicherheitsniveau in der Zukunft gemacht werden.

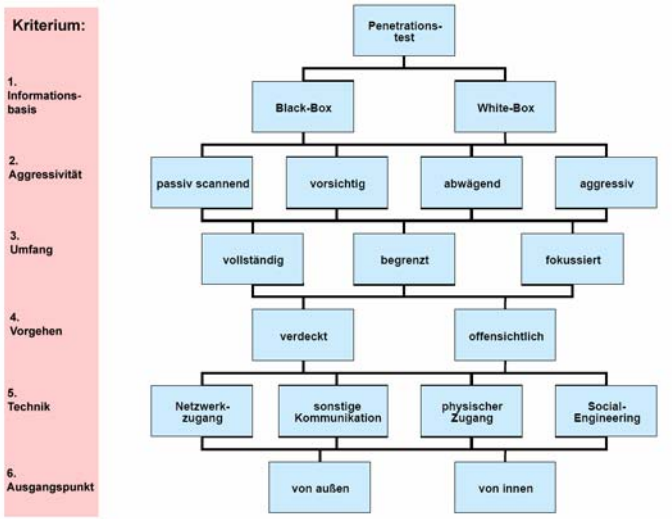

Abbildung 21: Aufbauvarianten eines Penetrationstests [BSI_PEN]

Ein Penetrationstest kann anhand verschiedener Kriterien individuell auf ein zu testendes System ausgerichtet werden. In Abbildung 21 sind auf der linken Seite sechs Kriterien aufgelistet. Nach diesen Kriterien kann man Penetrationstests unterscheiden. Zusätzlich sind auf der rechten Seite die unterschiedlichen Werte für die Kriterien grafisch zusammengefasst.

Die Ausführung des Penetrationstestes findet in folgenden Schritten statt:

1. Recherche nach Informationen über das Zielsystem:
Von welchem Wissensstand ausgehend wird das zu überprüfende System angegriffen? Wird eine Whitebox- oder ein Blackbox-Angriff simuliert? In einem Whitebox-Angriff besitzt der Angreifer Insiderwissen über das anzugreifende System, bei einem Blackbox Angriff hat der Angreifer keine internen Kenntnisse.

2. Scan der Zielsysteme auf angebotene Dienste:
Abhängig davon wie aggressiv der Penetrationstest durchgeführt wird, können angebotene Dienste auf dem Zielsystem aufgespürt werden. Bei geringer Aggressivität wird das zu testende System nur passiv gescannt. Vorteil ist, dass das System in seiner Funktionalität nicht beeinträchtigt wird. Es können aber nicht sämtliche Schwachstellen aufgedeckt werden. Je höher die Scan-Aggressivität ist, desto mehr Angriffsstellen können aufgedeckt werden. Hiermit steigt aber auch die Gefahr, dass das zu testende System, und sogar Sekundärhardware wie zum Beispiel Router ausfallen.

3. System- und Anwendungserkennung:
Wird erstmalig ein Penetrationstest durchgeführt ist es ratsam, das gesamte System mit allen Anwendungen zu überprüfen. Um bei späteren Tests den Aufwand gering zu halten, können wahlweiße nur noch bestimmte Bereiche gescannt werden.

4. Recherche nach Schwachstellen:
Es gibt primäre und sekundäre Schwachstellen. Primäre Schwachstellen sind generell alle diejenigen, die in direktem Zusammenhang mit dem Sicherheitssystem stehen. Sekundäre Schwachstellen sind beispielsweise das "IDS" (Intrusion Detection System), aber auch organisatorische und personelle Strukturen.

5. Ausnutzen der Schwachstellen:
Wurden in der Recherche Schwachstellen identifiziert, stellt sich die Frage welche Techniken gegen diese eingesetzt werden sollen. Bei einem klassischen Penetrationstest werden die Systeme nur über die Netwerkverbindung angegriffen. Zusätzlich könne aber ebenso direkte physische Angriffe und Social-Engineering Techniken eingesetzt werden.

Die Qualität und der Nutzen eines Penetrationstests sind im Wesentlichen davon abhängig, inwieweit dieser auf die individuelle Situation des Auftraggebers eingeht. Hier liegt in der Aufgabenstellung dieser Diplomarbeit die Schwierigkeit. Der zu erarbeitende Kriterienkatalog soll aus automatisierbaren Kriterien bestehen. Diese Automatisierung macht eine individuelle Abstimmung auf den einzelnen Auftraggeber nahezu unmöglich. Es

werden vielmehr lediglich standardisierte und bekannte Fehlerstellen überprüft werden können. Dies wirkt sich negativ auf die Qualität aus und eventuell werden nicht alle Ziele eines Penetrationstests erreicht werde können [BSI_PEN].
Abschließend lassen sich die Ziele eines Penetrationstest in vier Gruppen zusammenfassen:

1. Erhöhung der Sicherheit der technischen Systeme
2. Identifikation von Schwachstellen
3. Bestätigung der IT-Sicherheit durch einen externen Dritten
4. Erhöhung der Sicherheit der organisatorischen und personellen Infrastruktur

Genutzte Beratungs-Dienstleistungen	
Risikoanalysen und Konzeptentwicklung	55%
Penetrationstests	48%
Schwachstellenanalysen	44%
Strategie- und Managementberatung	35%
Kontrolle vorhandener Konzepte auf Eignung und Einhaltung	31%
Umsetzung von Konzepten und Maßnahmen	28%
Sonstiges	6%

Basis: 95 Beratungs-Kunden

Abbildung 22: Beratungs-Dienstleistungen [KES]

Als Fazit der Betrachtung des Penetrationstest lässt sich anführen, dass er generell ein förderliches Mittel ist, um die technische Sicherheit eines E-Commerce-Unternehmens zu überprüfen und Schwachstellen aufzudecken. Die KES-Studie aus dem Jahr 2004 zeigt das der Penetrationstest auf Platz zwei bei den Beratungsdienstleistungen liegt. Fast die Hälfte der befragten Unternehmen hatten zu diesem Zeitpunkt einen Penetrationstest auf ihrem System durchgeführt. Für den zu erarbeitenden Kriterienkatalog ist er jedoch in seiner Funktionalität zu mächtig. Für den Kriterienkatalog wird aufgrund seiner automatisierten und normierten Überprüfung zum Beispiel keine Voranalyse des zu testenden Systems möglich sein. Ebenfalls wird im Vorfeld kein umfassender Scan eines jeden zu untersuchenden Systems durchgeführt werden können. Stattdessen werden lediglich im Vorhinein definierte Bereiche gescannt werden können. Aus dem Penetrationstest lässt sich dennoch das grundsätzliche Vorgehen einer Systemprüfung, mit dem Ziel Prüfkriterien auszuarbeiten, übernehmen. Allerdings in wesentlich eingeschränkterem Umfang. Das Bundesamt für Sicherheit in der Informationstechnik bietet mit dem IT-Grundschutz-Katalog ein umfassendes Werk an, welches für die Auswahl der Prüfkriterien herangezogen werden kann.

4. Ausbildung von Kriterien

Für die Überprüfung von Webseiten unter dem Aspekt technischer Schwachstellen, wird ein Katalog verwendet, der für diese Aufgabe abgestimmte Prüfkriterien enthält. Als Prüfkriterien werden indirekt Schwachstellen verwendet, die bei einer Webseite auftreten können. Jedes Prüfkriterium ist folglich eine Schwachstelle oder eine Klasse von Schwachstellen. Beispiel hiefür ist die bekannte Cross-Site-Scripting-Schwachstelle. Wird eine Webseite bezüglich Cross-Site-Scripting-Schwachstellen untersucht, so kann anschließend beurteilt werden, ob die Webseite das Kriterium "Cross-Site-Scripting" erfüllt oder nicht. Erfüllt die Webseite das Kriterium im positiven Sinne, so können Cross-Site-Scripting-Attacken keine Schwachstellen in der Webseite finden, um einen Angriff zu starten.
Zu Beginn der Ausbildung der Kriterien wird ein kurzes Kapitel vorgestellt, das Aufzeigt "was" überprüft werden kann, und "was" nicht. Anschließend werden in tabellarischer Form Kriterien anhand von Schwachstellen aufgelistet. Dabei wird eine Unterteilung der Kriterien in Hinblick auf die Protokollebene, die Dienstebene, die Anwendungsebene und zusätzlich auf Qualitätsmerkmale vorgenommen. Die schon im ersten Kapitel erwähnte Problemstellung dieser Arbeit liegt des Weiteren in der Aufgabe, geeignete Kriterien auszuwählen die die Eigenschaft haben, automatisiert überprüft werden zu können. Das setzt voraus, dass die Kriterien streng normierbar sind. Nur auf diese Weise werden bei unterschiedlichen IT-Systemen Überprüfungen automatisiert durchgeführt werden können. In vielen Fällen ist aber eine individuelle Anpassung der Kriterien an das zu testende System notwendig.

4.1 Vorbedingungen für Kriterien

Nicht alle Kriterien kommen für einen Kriterienkatalog in Frage. Dieses Kapitel beschäftigt sich mit Vorbedingungen, die ein Kriterium erfüllen muss, um in den Katalog aufgenommen werden zu können. In Kapitel 2.2.6 wurde ein Praxisbeispiel für eine logische Schwachstelle einer E-Commerce-Webseite aus den USA vorgestellt. Die Schwachstelle bestand darin, dass eindeutig falsche Kundenbestellungen, nicht als solche erkannt und nicht verworfen wurden. Prinzipiell kann diese Schwachstelle gut als Ansatz für eine automatisierte Überprüfung herangezogen werden. Die Schwierigkeit liegt hier nicht in dem Aufwand einen Roboter zu programmieren der die Webseite des E-Commerce-Unternehmens mit eindeutig falschen Bestellungen testet. Zu diesem Zweck müssten die Eingabefelder einer Bestellseite lediglich mit geeigneten Werten (z.B. Werte größer 10.000) getestet werden.

Die Problematik besteht vielmehr darin, dass es durchaus auch Webshops geben kann, die Bestellmengen von größeren Stückzahlen akzeptieren müssen. Die Problematik kann gut mit dem folgenden Beispiel zweier Webshops gezeigt werden. Auf der einen Seite wird die Webseite eines Webshop der Eisenwaren, wie Nägel und Schrauben verkauft, und auf der anderen Seite einen Webshop der Ferienhäuser vermietet betrachtet. Der Shop der Eisenwaren, insbesondere Schrauben, verkauft wird Bestellungen in Stückzahlen von über 10.000 akzeptieren. Hingegen würde der Webshop der Ferienhäuser vermietet, gewiss keine Stückzahlen von 10.000 Ferienhäusern eines bestimmten Typs vermieten können.

Notwendigkeit von Blackboxtests

In der Praxis ist die Überprüfung eines IT-Systemes in der Form vorzustellen, dass dieser ohne interne Kenntnisse über das zu testende System durchgeführt werden muss. Die Auswahl der Kriterien zielt aufgrund dessen auf allgemein gültige Kriterien, die ohne Vorwissen über das zu testende System abgearbeitet werden können. Die Notwendigkeit eines Blackboxtestens erschließt sich aus dem oben angeführten Beispiel. Bei einem Anbieter mag eine Bestellgröße von mehr als 10.000 Einheiten unrealistisch sein, bei einem anderen nicht. Eine Entscheidung auf Korrektheit einer Überprüfung kann nur getroffen werden, wenn eine Interpretation des Ergebnisses möglich ist. Eine Interpretation kann aber nur erfolgen, wenn dem Prüfer im Vorfeld individuelle Erkenntnisse über den jeweils zu prüfenden Kunden vorliegen. Dies steht im Widerspruch zu einer automatisierten Überprüfung, bei der keine individuelle Anpassung gewünscht wird.

Beeinträchtigung des Testsystems

Gewisse Kriterien sind nicht geeignet in den Katalog aufgenommen zu werden. Dies ist immer dann der Fall wenn bei einer Überprüfung negative Auswirkungen auf die Performance des zu testenden Systems oder die Konsistenz der Daten befürchtet werden muss. Dies wäre zum Beispiel bei einem simulierten DoS-Angriff oder bei einem Scan aller durchführbaren SQL-Injections der Fall. Diese Überprüfungen von Schwachstellen könnte aufgrund ihrer negativen Auswirkungen auf ein IT-System bei einer automatisierten Überprüfung einem Kunden nur schwer zugemutet werden. Ähnliches gilt bei der Überprüfung von Webformularen, über die E-Mails generiert werden. Bei einer automatisierten Überprüfung dieser Eingabefelder können schnell einige tausend Test-Mails entstehen. Durch diese Masse an Mails kann der Mailclient des geprüften Unternehmens ausgelastet und sogar zum Absturz gebracht werden.

Installation von zusätzlicher Software auf Kundenseite

Ebenfalls aus der Auswahl fallen Kriterien, für deren Überprüfung zusätzliche Software auf dem IT-System des Kunden installiert werden müssen. Zum einem wird dies ein Kunde nur

in den seltensten Fällen erlauben. Zum anderen würde eine individuelle Installation von Software einer automatisierten Überprüfung im Widerspruch stehen.

Anwendbare Rückgabewerte

Ein E-Commerce-Anbieter verspricht sich von der Überprüfung seiner Webseiten zwei positive Einflüsse. Zum einem einen Imagegewinn durch das Erlangen eines Zertifikates, dass er auf seiner Webseite seinen Kunden präsentieren kann. Zum anderen möchte der Auftraggeber, wenn Schwachstellen aufgedeckt werden, anwendbare Rückmeldungen bekommen mit denen er die Schwachstellen schließen kann. Im Idealfall besteht die Rückmeldung aus einem klaren Hinweis auf die Schwachstelle, wenn möglich im Verbund mit einer Einstufung des Gefährdungsgrades.

Vorstellbar wäre z.B. eine folgende Rückgabemeldung auf die Überprüfung der verwendeten PHP-Version:

```
Schwachstelle Nr: 01      Betreff: PHP-Version
-------------------------------------------------------------------
Diagnose:    Sie benutzen die veralterte PHP-Version 5.1.0 (13.06.2004).

Schlüssel:   Upgrade sie ihr System auf die aktuelle PHP-Version 5.2.1
             (03.2007) um eine bestmöglich Systemsicherheit zu
             gewährleisten.
             Download: http://www.php.net
```

Nicht jedes Prüfkriterium erlaubt eine schlüssige Rückgabemeldung, die dem Auftraggeber zweckdienlich bei der Schließung einer Schwachstelle dienen würde. Eine solche Meldung wäre zum Beispiel eine pauschale Rückmeldung ohne Aussagewert, dass eine HTTP-Request-Schwachstelle auf dem Webserver entdeckt wurde, ohne diese Schwachstelle genauer zu spezifizieren.

Falsch:

```
Schwachstelle Nr: 02      Betreff: HTTP-Request
-------------------------------------------------------------------
Diagnose:    HTTP-Request Schwachstelle in ihrem IT-System.

Schlüssel:   Konfiguration des Systems
```

Aus dieser Fehlermeldung kann von einem Auftraggeber zu wenig an Informationen herausgenommen werden. Es kann von ihm nur erkannt werden, dass eine Request-Schwachstelle im System besteht. In der Meldung wird weder die Art der Schwachstelle näher erläutert, noch wo diese im System anzusiedeln ist. Eine Schließung der Schwachstelle ist somit für den Auftraggeber nur schwer möglich. Besser wäre der Ansatz die http-Request-Kriterien einzeln abzuprüfen (GET, PUT, DELETE). Dementsprechend würden die Rückmeldungen angepasst werden. Durch diese Selektierung ist es dem

Auftraggeber möglich die Schwachstelle einem Systembaustein zu zuordnen und somit auch zu schließen.

Richtig:

```
Schwachstelle Nr: 02      Betreff: HTTP-Request (PUT)
-------------------------------------------------------------------
Diagnose:    HTTP-Request Schwachstelle in ihrem IT-System.
             Das PUT-Kommando ihres Webservers ist aktiv und kann
             benutzt werden Dateien in ihr System einzuschleusen.

Schlüssel:   Konfiguration des Webservers ändern.
             Deaktivierung des PUT-Kommandos.
```

Infos über Schwachstellen auf Netzwerkebene sind für mittelständische Shopbetreiber nicht unmittelbar aussagekräftig. Dies beruht auf der Begebenheit, dass sie in der Regel ihre Serverstruktur zu Providern auslagern. Hierdurch sind ihre Systembausteine der Netzwerkebene von dem Verhalten des Providers abhängig. Folglich können die Shopbetreiber Probleme, die auf den unteren Systemebenen auftreten (Protokoll- und Netzwerk-Ebene), nicht eigenständig lösen.

Gesetzliche Bestimmungen

Prüfkriterien können, wenn sie nicht mit der Einwilligung des Auftraggebers durchgeführt werden, gegen geltende Gesetzte verstoßen. Wesentlich ist daher, dass im Vorfeld eine exakte Festlegung des beauftragten Rahmens der Überprüfung zwischen dem Auftraggeber und dem Auftragnehmer vereinbart wird. Ein passwortgeschützter WWW- oder FTP-Server stellt beispielsweise einen zugangskontrollierten Dienst dar. Werden die Prüfkriterien unter zur Hilfenahme eines Tools auf eine Weise überprüft, das ein vorhandener Schutzmechanismus umgangen wird, findet ein Verstoß gegen das Zugangskontrolldienstschutzgesetz (ZKDSG) statt [BSI_PEN].

Aktualität der Kriterien:

Ein generelles Problem bei der Erstellung eines Kriterienkataloges ergibt sich aus dem sich ständig vollziehenden Wandel des IT-Bereiches. Die Protokollebene mit den ICMP-, IP- und TCP-Protokollen mag hiervon weniger stark betroffen sein, da dieser Bereich nachhaltig normiert und über längere Zeiträume unverändert bleibt. Hingegen unterliegt der Bereich der verwendeten Betriebssysteme, Dienst- und Anwendungsprogrammen einem ständigen Wandel durch Erweiterungen und Updates. Ein Kriterienkatalog kann folglich nur alle aktuell bekannten Schwachstellen abdecken. Es kann vorkommen, dass eine Schwachstelle besteht, aber noch nicht aufgedeckt wurde. Folglich kann sie auch nicht von einem Kriterienkatalog überprüft werden. Die Aussage, dass ein geprüftes System frei von Fehlerstellen sei, wird ein Kriterienkatalog also zukünftig nicht bieten können.

4.2 Strukturierung der Kriterien

Zur Ausbildung von Prüfkriterien und einer Sicherheitskonzeptionen von Webanwendungen bietet sich eine Einteilung der Schwachstellen in Ebenen an. Diese erfolgt in Anlehnung an die BSI-Studie "*Sicherheit von Webanwendungen, Maßnahmenkatalog und Best Practices*" [BSI_A].

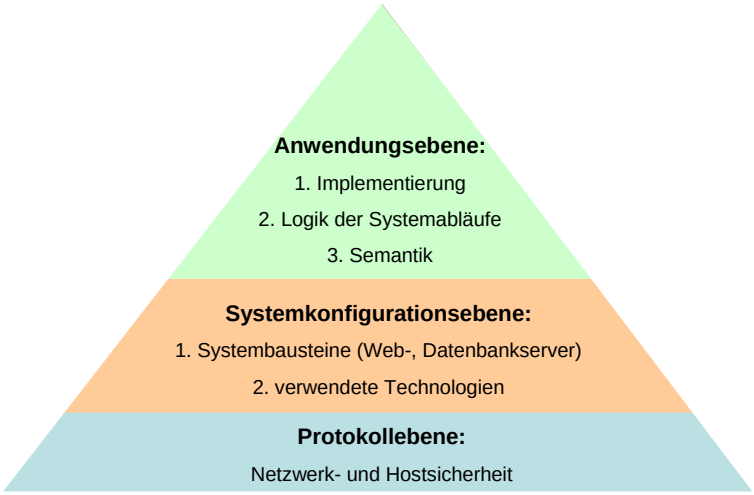

Abbildung 23: Ebenenmodell

Schwachstellen eines E-Commerce-Webauftrittes werden in drei Ebenen unterschieden. Die unterste Ebene stellt die Protokollebene mit Netzwerk- und Systemprotokollen dar. Darauf schließt die Systemkonfigurationsebene und als dritte und oberste Ebene die Anwendungsebene an. Die Systemebene, auch Dienstebene genannt, behandelt sowohl Konfigurationsfehler von Systembausteinen, als auch die verwendeten Technologien. Hierbei wird analysiert ob die für einen bestimmten Zweck eingesetzte Technologie die richtige Wahl ist. Die Anwendungsebene baut ihre Kriterien auf Implementierungs-, Logik- und Semantikfehlern auf. Die in Kapitel Zwei eingeführten Schwachstellen werden in diesem Kapitel in das Ebenenmodell eingefügt. Hierbei werden, falls notwendig, Kriterien zusammengefasst oder in mehrere aufgeteilt.

► Das Ziel der Unterteilung in ein Ebenenmodell liegt darin, einem E-Commerce-Unternehmen zu ermöglichen Schwachstellen jeweils zuständigen Unternehmensbereichen und Verantwortlichen zuzuordnen. Auf diese Weise können die Aufgaben, die für eine Sicherheitskonzeption und die Realisierung einer Webanwendung benötigt werden, den relevanten Unternehmensbereiche zugeordnet werden.

4.2.1 Protokollebene

Die Protokollebene repräsentiert die unterste Ebene im vorgestellten Ebenenmodell. Die Dienste und Anwendungen der System- und Anwendungsbene bauen auf der Protokollebene auf und greifen, wenn Dienste dieser Ebene benötigt werden, auf diese zurück. In der Protokollebene werden die Übertragungsprotokolle und Methoden betrachtet, die von den Hardwareherstellern vereinbart wurden. Sie befasst sich ferner mit der Absicherung des Netzwerks und des Host gegen Angriffe. Schwachstellen die von der Netzwerk- oder der Serverhardware und dem darauf laufendem Betriebssystem ausgehen, sind keine Schwachstellen die insbesondere bei E-Commerce-Webseiten auftreten. Bei diesen Schwachstellen handelt es sich viel mehr um generelle Schwachstellen von Webseiten, die ihren Ursprung in der Hardware- und Treibersoftwarearchitektur haben und unabhängig von der verwendeten Webanwendung auftreten. Die Umsetzung grundlegender Sicherheitsmaßnahmen auf dieser Ebene wird gleichwohl als zwingende Voraussetzung für sichere Webanwendungen betrachtet. Folglich werden die Kriterien, die Schwachstellen der Protokollebene betreffen, in den Kriterienkatalog aufgenommen[BSI_A].

Viele Angriffe werden auf der unteren Protokollebene durchgeführt. Der Grund liegt darin, dass auf dieser Ebene nur eine schwache Protokollierung stattfindet. Ein Angreifer muss folglich nur selten mit einer Entdeckung rechnen. Darüber hinaus werden bekannte Authentisierungsmechanismen erst auf der höheren Dienst oder Anwendungsschicht eingesetzt. Auf der unteren Protokollebene findet nur eine schwächere Authentisierung, z.B. mittels Paketnummern (TCP-Protokoll) oder IP-Adressen (IP-Protokoll) statt.

Die folgenden vier Protokolle werden in Netzwerken zur Kommunikation verwendet. Aufgrund der weiten Verbreitung bieten sie einem Angreifer Schwachstellen für einen Angriff und werden auf der Protokollebene in den Kriterienkatalog aufgenommen.

1. Address Resolution Protocol (ARP):
ARP ermöglicht eine Zuordnung von Netzwerkadressen zu Hardwareadressen. In den Kriterienkatalog wird das ARP-Spoofing aus dem zweiten Kapitel aufgenommen.
Insbesondere wenn der ARP-Push Befehl aktiv ist können die ARP-Tabellen verändert werden.

2. Internet Protokoll (IP):
IP ermöglicht es, dass Computer innerhalb eines Netzwerkes in logische Einheiten, sogenannte Subnetze, gruppiert werden können. Auf dieser Basis ist es möglich, Computer in größeren Netzwerken zu adressieren und Verbindungen zwischen ihnen aufzubauen. Es

gibt eine Reihe von Schwachstellen, die im Internet Protokoll ihren Ursprung haben (Fragmentierung). Zusätzlich wird IP-Spoofing als weitere Schwachstelle in den Kriterienkatalog aufgenommen. In dem zweiten Kapitel wurden das IP-Spoofing, der TCP/IP-Handshake und die TCP/IP-Sequenznummern als Schwachstellen für mögliche Angriffe genannt.

3. Transmission Control Protocol (TCP):
Das TCP wird im Verbund mit IP eingesetzt und besteht aus einer Vereinbarung der Hardwarehersteller darüber, auf welche Art und Weise Daten zwischen Computern ausgetauscht werden sollen. Die Schwachstellen von TCP und IP können zusammengefasst werden da die beiden Protokolle in der Praxis fast immer zusammen verwendet werden. Aufgrund dieser Praxis gehen viele Schwachstellen gleichermaßen von beiden Protokollen aus.

4. Internet Control Message Protocol (ICMP):
ICMP wird in Netzwerken zum Austauschen von Fehler- und Informationsmeldungen benutzt. Insbesondere Konfigurationsfehler bei der Verwendung von ICMP können zu Schwachstellen führen. Sie erlauben sogenannte ICMP-Tunnel-Angriffe über die Angreifer Zugriff auf einen Rechner erlangen können.

Auf den nachfolgenden zwei Seiten werden in einem Katalog Prüfkriterien aufgelistet, die auf der Protokollebene basieren. Jedem Kriterium (Schwachstelle) wird ein Titel gegeben. Es wird die verwendete Methode, ihre Ursache und die möglichen Schäden für ein E-Commerce-Unternehmen dargelegt. Zusätzlich wird eine Schätzung gemacht, in wie weit eine Gefährdung für ein E-Commerce-Unternehmen besteht (niedrig, mittel, hoch). Die Ergebnisse aus Bereich der Protokollebene des Kriterienkataloges richten sich an zuständige Netzwerk- und Systemadministratoren. Durch die Unterteilung des Kriterienkataloges in drei Ebenen ist es den verantwortlichen Personen in einem Unternehmen möglich, die aufgedeckten Schwachstellen zu beheben.

Als weiterführende Verknüpfung findet in der letzten Spalte der Tabelle eine Zuordnung der Kriterien zu dem BSI-Grundschutz-Katalog statt. Nicht alle Kriterien werden von dem BSI-Grundschutz-Katalog sowohl im Gefährdungs-, als auch im Maßnahmenkatalog behandelt. Kann eine Schwachstelle nicht dem Gefährdungskatalog zugeordnet werden ein, wird die betreffende Stelle im Maßnahmenkatalog angegeben oder umgekehrt.

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
ARP-Ebene	▫ ARP-Push Befehl ▫ Manipulation der ARP-Tabelle (nur innerhalb eines Netzsegments da nicht Routing-fähig)	Designfehler im Kommunikationsprotokoll	▫ Denial-of-Service ▫ Vertraulichkeitsverlust der transportierten Daten ▫ Integritätsverlust der transportierten Daten	■	G 5.112
TCP/IP-Ebene	▫ IP-Fragmentierung	Designfehler im Kommunikationsprotokoll	Zugriff auf Netzressourcen	■	
	▫ Reserved Bit Überprüfung eines fragmentierten SYN-Paketes	Designfehler im Kommunikationsprotokoll, Implementierungsfehler	Zugriff auf Netzressourcen	■	
	▫ SYN mit unerreichbaren Quelladressen	Designfehler im Kommunikationsprotokoll	Denial-of-Service	■	M 5.105
	▫ CPU-Angriffe mittels Telnet (nur Windows NT) ▫ Ausnutzen von anfälligen Ports	Implementierungsfehler	Denial-of-Service	■	M 5.39
ICMP-Ebene	▫ ICMP Redirect	Designfehler im Kommunikationsprotokoll	▫ Vertraulichkeitsverlust der übermittelten Daten ▫ Integritätsverlust der übermittelten Daten	■	G 5.50
	▫ ICMP Destination Unreachable	Designfehler im Kommunikationsprotokoll	Denial-of-Service	■	M 5.120
	▫ ICMP Echo Request (Ping of death)	Implementierungsfehler	Denial-of-Service	■	M 5.120
	▫ ICMP Zeitstempel	Designfehler im Kommunikationsprotokoll	Angreifer gewinnt Informationen	■	M 5.120
	▫ ICMP-Tunnel	Konfigurationsfehler, Designfehler in der Dienstspezifikation	▫ Zugriff auf Rechnerressourcen ▫ Datenvertraulichkeitsverlust ▫ Datenintegritätsverlust ▫ Verfügbarkeitsverlust ▫ Denial-of-Service	■	M 2.70

Tabelle: Kriterienkatalog auf Protokollebene (1) [HK_97], [BSI_06]

Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
Spoofing	▫ IP-Spoofing	Designfehler im Kommunikationsprotokoll	Zugriff auf Netzressourcen	■	G 5.48
	▫ TCP sequence number guessing	Designfehler im Kommunikationsprotokoll	Zugriff auf Netzressourcen	■	
	▫ DNS-Spoofing	Designfehler in der Dienstspezifikation	▫ Datenvertraulichkeitsverlust während des Transports ▫ Datenechtheitsverlust	■	G 5.78 G 4.39
	▫ RIP-Spoofing	Designfehler im Kommunikationsprotokoll	▫ Denial-of-Service ▫ Datenvertraulichkeitsverlust während des Transports ▫ Integritätsverlust der transportierten Daten	■	M 5.39
	▫ Source-Routing	Designfehler im Kommunikationsprotokoll	▫ Datenvertraulichkeitsverlust während des Transports ▫ Integritätsverlust der transportierten Daten	■	G 5.49
	▫ Session Hijacking	Designfehler im Kommunikationsprotokoll	▫ Zugriff auf Rechnerressourcen	■	G 5.89
	▫ Flooding	Designfehler, Konfigurationsfehler	▫ Denial-of-Service	■	G 5.112
	▫ Angriffe durch Kapseln und Tunnelangriffe	Konfigurationsfehler	▫ Zugriff auf Rechner- und Netzressourcen	■	

Tabelle: Kriterienkatalog auf Protokollebene (2) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog
[HK_97]

4.2.2 Systemkonfigurationsebene (Dienstebene)

Die Systemkonfigurationsebene, die auch kürzer als Dienstebene bezeichnet wird, befasst sich mit Schwachstellen, die sich aus den eingesetzten Systemkomponenten und deren verwendeter Technologie ergeben. Die Dienstebene ist zwischen der Protokoll- und der Anwendungsebene angeordnet. Eine Kommunikation findet immer nur zwischen den direkt benachbarten Ebenen statt.

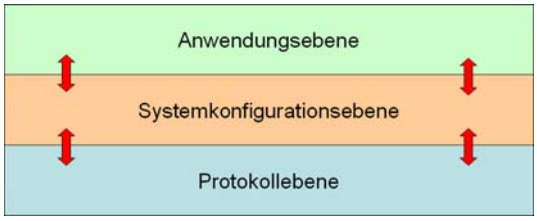

Abbildung 24: Ebenen-Kommunikation

Die Sicherheitskriterien der Systemebene ergeben sich aus den auf dieser Ebene verwendeten Bausteinen und deren Schwachstellen. Neben den Softwarebausteinen (Betriebssystem und Dienstprotokolle) gehören dazu auch Hardwarebausteine, wie zum Beispiel der Webserver und der Applikationsserver, aber auch Datenbank- und Backend-Systeme. Auf jedem dieser Server und Systeme läuft erneut ein Dienstprotokoll. Diese Komponenten müssen folglich bei der Sicherheitskonzeption einer Webanwendung mit einbezogen werden. Die Kriterien der Systemebene richten sich an die Netzwerk- und Systemadministratoren.

Die weiteren Sicherheitskriterien der Dienstebene werden aus der eingesetzten Technologie (Konzeptionsfehler, Konfigurationsfehler) ermittelt. Die eingesetzte Technologie muss ordnungsgemäß für den jeweiligen Einsatzzweck und Schutzbedarf der Webpräsenz ausgewählt, sowie korrekt genutzt (konfiguriert) werden. Ein Konzeptionsfehler tritt auf, wenn zum Beispiel bei einer Webanwendung, die sensible Daten unverschlüsselt über das Internet transferiert, nicht die richtige Technologie eingesetzt. Eine Webanwendung, bei der die Passwörter zwar verschlüsselt werden, dafür aber einen zu kurzen Schlüssel verwendet wird, wird die richtige Technologie falsch eingesetzt. Die Kriterien der Technologieebene richten sich an die zugehörige Fachstelle, bzw. den Entwickler. Ein Konfigurationsfehler besteht immer dann, wenn die richtige Technologie falsch eingesetzt (falsch konfiguriert) wird [BSI_A].

Zusammengefasst geht die maßgebliche Gefährdung eines E-Commerce-Unternehmens im Bereich der Dienstebene von Architekturfehlern in der Dienstspezifikation (Konzeptionsfehler) oder von fehlerhaft genutzten Diensten (Konfigurationsfehler) aus.

In den Kriterienkatalog fließen die in dem zweiten Kapitel vorgestellten Konfigurations- und Konzeptionierungsfehler ein. Konfigurationsfehler sind für Betreiber von E-Commere-Webseite insbesondere bei der Verwendung von Servern, der allgemeinen Systemkonfiguartionen, bei einer SQL- oder PHP-Datenbanken und dem Network File System kritisch und bieten Ansatzpunkte für Angriffe. Designfehler liegen oft in einer Dienstspezifikation und können von einem E-Commerce-Unternehmen nur indirekt behoben werden. So werden in dem DNS-Dienst, dem File Transfer Protocol immer wieder Schwachstellen für Angriffe verwendet.

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
DNS-Angriffe	▫ DNS-Cache Verunreinigung	Designfehler in der Dienstspezifikation	▫ Vertraulichkeitsverlust der im Cache liegenden Daten ▫ Authentizitätsverlust	■	G 5.78
	▫ Reserve Lookup	Konfigurationsfehler	▫ Datenechtheitsverlust	■	
Network Information System (NIS)	▫ NIS Authentisierung	Designfehler in der Dienstspezifikation	▫ Zugriff auf Rechnerressourcen	■	M 5.18
FTP	▫ Aktiver/Passiver FTP-Angriff	Designfehler in der Dienstspezifikation	▫ Datenechtheitsverlust, ▫ Datenvertraulichkeitsverlust auf Rechner	■	M 5.21 5.43 5.39
	▫ FTP-Brute-Force Angriff	Designfehler in der Anwendung, Konfigurationsfehler	▫ Zugriff auf Rechnerressourcen, ▫ Datenvertraulichkeitsverlust auf Rechner, ▫ Datenintegritätsverlust auf Rechner	■	
SQL-Injection	▫ Datenbanken ▫ einschleusen eigener Befehle ▫ SQL Datenbanken manipulieren	Konfigurationsfehler, fehlende Data Validation	▫ Datenvertraulichkeitsverlust auf Datenbank, ▫ Datenintegritätsverlust auf Datenbank ▫ verfälschen von Daten- und Warenbestände in der Datenbank	■	M 2.363
Nicht benötigte Dienste	▫ Ausnutzen von Schwachstellen auf Anwendungsprogrammen die für den Betrieb des E-Shops nicht von Nöten wären	Konfigurationsfehler	▫ Zugriff auf Rechnerressourcen,	■	
Network File System (NFS)	▫ erraten des "File-handle" ▫ mount/unmount Operation	Designfehler in der Dienstspezifikation, Konfigurationsfehler	▫ Datenintegritätsverlust, ▫ Verfügbarkeitsverlust, ▫ Datenvertraulichkeitsverlust	■	M 5.17

Tabelle: Kriterienkatalog auf Dienstebene (1) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
Einsatz von SSL/TLS/HTTPS	▫ abfangen von Cookies (sensitive Session ID) ▫ Referrer Information	Implementierungsfehler, Technologiefehler	▫ Authentizitätsverlust ▫ Datenvertraulichkeitsverlust während des Transports	■	M 5.66
Monitoring und Patching	▫ Eingesetzte Standard- und Systemsoftware ist auf "Known Vulnerabilities" zu überwachen, Logdateien sind zu monitoren	Bekanntwerden von Sicherheitsschwachstellen	▫ Abhängig von der Betroffenen Systemsoftware	■	M 2.273
Systemkonfiguration	▫ Benutzerkennung für Web-/Applicationsserver unter UNIX (1 Webserver ↔ 1 Passwort)	Konfigurationsfehler	▫ Datenvertraulichkeitsverlust	■	M 4.63
	▫ Dateiberechtigung, Zugriffsrechte	Konfigurationsfehler	▫ Datenvertraulichkeitsverlust	■	G 3.57
Serverkonfiguration	▫ Environment von Serverprozessen und Startup-Skripte	Konfigurationsfehler	▫ Datenvertraulichkeitsverlust	■	M 2.367
	▫ Webserver (Konfiguration): → Benutzerkennung, Zugriffsrechte, Prozesse, Verzeichnisse, Dateien, Fehlerausgaben und Protokollierung	Konfigurationsfehler	▫ Datenintegritätsverlust, ▫ Datenvertraulichkeitsverlust	■	M 4.239
	▫ Datenbankserver (Konfiguration): → Benutzerkennung, Zugriffsrechte, Tabellen, Stored Procedures und Tabelleninhalte	Konfigurationsfehler	▫ Datenvertraulichkeitsverlust auf Datenbank, ▫ Datenintegritätsverlust auf Datenbank ▫ verfälschen von Daten- und Warenbestände in der Datenbank	■	M 5.119

Tabelle: Kriterienkatalog auf Dienstebene (2) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
Serverkonfiguration	▫ PHP-Umgebung: → PHP-Schwachstellen ausschließen (Variablen sichern, Definition von Verzeichnissen für Includes und Uploads, Ausführbare externe Programme, Sessions absichern, Datenbankanbindung, Eingabedaten maskieren)	Konfigurationsfehler	▫ Datenintegritätsverlust, ▫ Datenvertraulichkeitsverlust	■	M 2.363
Web Application Security Tools	▫ WebShields (Firewall), ▫ WebScanner (Aufdecken von Schwachstellen)	Konfigurationsfehler	▫ Datenintegritätsverlust, ▫ Datenvertraulichkeitsverlust	■	B 3.301

Tabelle: Kriterienkatalog auf Dienstebene (3) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

4.2.3 Anwendungsebene

Die Anwendungsebene umfasst Sicherheitskriterien aus dem Gebiet der Anwendungs-Implementierung, der dabei zugrunde gelegten Logik und der verwandten Semantik.

Abbildung 25: Aufbau der Anwendungsebene [BSI_A]

Implementierungsfehler

Unter Implementierungsfehler werden unbeabsichtigte Programmierfehler, aber auch eine nicht vorhandene oder ungenügende Prüfung der Eingabedaten "Data Validation" verstanden. Die Gruppe von Implementierungsfehler beinhaltet ferner ein ungenügendes Testverfahren bei der Entwicklung von Anwendungen, und die Vernachlässigung der Qualitätssicherung zugunsten einer früheren Inbetriebnahme oder einem Mangel an Qualität der entwickelten Software aus Kostengründen. Zur Stützung der Sicherheitskriterien bietet der Kriterienkatalog einen Bezug zum Maßnahmen- und Gefährdungs-Kataloges des BSI IT-Grundschutz-Kataloges. So kann z.B. von einem Softwareentwickler im Maßnahmen Katalog M2.83 nachgelesen werden, wie ein bestmöglicher Test von Anwendungs-progammen durchgeführt wird. Programmierfehler wurden in Kapitel Zwei mit Maleware, Cross-Site-Scripting, Buffer Overlows und anderen vorgestellt. Hier fließen diese Programmierfehler wieder in den Kriterienkatalog ein, in dem sie in die Anwendungsebene einsortiert werden.

Viele Kriterien des Kriterienkataloges stammen aus dem Bereich der Anwendungsebene. Die Kriterien beruhen auf Implementierungs- bzw. Designfehlern von Webanwendungen. Gerade Webanwendungen sind aktuell das Ziel von Angriffen. In den letzten Jahren haben sich die Angriffe auf Webseiten gewandelt. Ende der 90er Jahre hatten Angriffe im Internet das Ziel Hardwarkomponenten wie Switche, Router oder ganze Server durch eine Überlastung zum erliegen zu bringen. Die Angriffe waren darauf ausgerichtet bei Webauftritten einen Systemausfall herbeizuführen. Heutzutage werden von Hackern in erster Linie Webanwendungen angegriffen, da diese üblicherweise von verschiedenen Firmen und mit durchaus unterschiedlicher Qualität erstellt werden. Die Webanwendungen

sind selbst für weniger erfahrene Hacker ein einfaches Ziel. Zusätzlich beruhen inzwischen viele Angriffe auf finanziellen Interessen. Ein Artikel der Frankfurter Allgemeinen Zeitung (FAZ) vom September 2006 drückte diese Veränderung passend in der Überschrift *„Hacker wollen Geld statt Ruhm und Ehre“* aus [FAZ].

Die Angriffe auf Webanwendungen können verschiedene Methoden und Quellen haben. Ein Angriff kann auf unlauteren Methoden von Konkurrenzunternehmen beruhen. Gerde im E-Commerce-Bereich ist es zum Beispiel üblich, dass eine Manipulation des Rankings der Webpräsenz auf bekannten Suchmaschinen wie Google vorgenommen wird. Hierdurch kann ein Unternehmen erreichen, dass es bei einer Suchanfrage früher als ein Konkurrenzunternehmen gelistet wird. Bekanntes Praxisbeispiel ist ein Fall von BMW und Google, geschehen im Frühjahr 2006. Google nahm alle BMW-Seiten aus seinem Index, nachdem bekannt geworden war, das BMW das Ranking manipuliert hatte [BMW].

Es können aber auch Angriffe von einzelnen Personen ausgehen, die an die Daten der Benutzer gelangen wollen. Das kann das Ausspionieren von E-Mail-Adressen sein, um diese Gewinnbringen weiter zu verkaufen. Aber auch der Diebstahl von Benutzerdaten mit denen sich der Hacker einen finanziellen Vorteil erwirtschaften kann (z.B. Kreditkartennummern, TANs, PINs, ...).

Logikfehler

Logikfehler führen zu Schwachstellen und somit zu Sicherheitskriterien wenn Abläufe innerhalb von Webanwendung logisch inkonsequent abgearbeitet werden. Aber auch in der Interaktion mit dem Benutzer sind Logikfehler möglich. Ist die Interaktion Gewissermaßen zu zweckorientiert implementiert worden, kann gegebenenfalls eine Schwachstelle die für einen Angriff missbraucht werden kann vorliegen. Wird z.B. nach dem dritten fehlerhaften Loginversuch eine weitere Eingabe des Passwortes und die entsprechende Benutzerkennung gesperrt, so kann ein Benutzer durch eine andere Person gezielt ausgesperrt werden. Diese missbräuchliche Vorgehensweise wird weiter erleichtert, wenn die Benutzerkennung einfach zu erraten ist. Logikfehler in Anwendungen können oft nur von einer Person erkannt werden, die Einblick in die internen Geschäftsprozesse eines Unternehmens hat. Aufgrund dessen richten sich die Kriterien des Kataloges dies bezüglich an Mitarbeiter der betreffenden Fachstelle im Unternehmen.

Semantikfehler

Die semantische Ebene umfasst Kriterien die über inhalts- und kommunikationsbezogene Aspekte verfügen. Über die Semantik wird der Vertrauenskontext für die Interaktion mit

einem Benutzer hergestellt. Wird in diesem Bereich nicht ein hohes Maß an Sorgfalt aufgewendet, so kann eine Webanwendung von Dritten missbraucht werden, um einen Benutzer zu täuschen. Dieser Bereich kann selten auf eine einzelne Anwendung beschränkt bleiben, vielmehr ist eine Webseite- oder Unternehmensübergreifende Betrachtung notwendig. Entsprechend schwierig gestaltet sich in diesem Gebiet die Auswahl von geeigneten Kriterien für eine Überprüfung. Am ehesten kann in diesen Bereich ein Fehlerverhalten eines Benutzers eingeordnet werden, da hier die Ursache oft in einer unzureichenden Semantik (Eindeutigkeit, Nachvollziehbarkeit, ...) des Webseitenbetreibers liegt. [BSI_A].

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
Sendmail	▫ SMTP unter Unix root-Rechte erlangen	Implementierungsfehler, Designfehler in der Anwendung	▫ Zugriff auf Rechnerressourcen	■	M 5.19
Mail Spoofing	▫ Direkte Kommunikation mit dem SMTP-Server (über Port) ▫ versenden von E-Mails über beliebigem Kennungen	Designschwäche des Kommunikationsprotokolls	▫ Datenechtheitsverlust	■	M 2.118
Verteilter, koordinierter Angriff	▫ Häufige Zugriffe auf das Zielsystem	Designschwäche des Kommunikation-Protokolls	▫ Denial-of-Service	■	
Erraten von Passwörtern	▫ Elektronische Wörterbuch-Attacke	Fehlverhalten der Benutzer	▫ Zugriff auf Rechnerressourcen	■	G 5.18
Viren	▫ Einschleusen von schadhaftem Code	Benutzerfehlverhalten, Konfigurationsfehler	▫ Datenintegritätsverlust auf Rechner ▫ Datenvertraulichkeitsverlust auf Rechner ▫ Verlust der Verfügbarkeit ▫ Denial-of-Service	■	G 5.23
Trojanische Pferde	▫ Einschleusen von schadhaftem Code	Benutzerfehlverhalten, (absichtliche) Programmierfehler	▫ Zugriff auf Rechner	■	G 5.21
Würmer	▫ Massenhaftes Einschleusen und Verbreiten von schadhaftem Code	Andere Sicherheitslücken	▫ Zugriff auf Rechnerressourcen ▫ Denial-of-Service ▫ Datenintegritätsverlust auf Rechner ▫ Datenvertraulichkeitsverlust auf Rechner	■	M 4.177
Java	▫ Ausführung von Programmen	Implementierungsfehler, Designfehler in der Anwendung	▫ Zugriff auf Rechnerressourcen	■	G 5.88

Tabelle: Kriterienkatalog auf Anwendungsebene (1) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
Cross Site Scripting (XSS)	▫ Ausnutzen von: HTML, Java, PHP, Perl ▫ Code von anderen Quellen in neuem Kontext einbinden ▫ beständiges ↔ unbeständiges	Implementierungsfehler, Designfehler in der Anwendung, fehlende Data Validation, Semantikfehler	▫ Verlust der Datenintegrität (Webseiten) ▫ Diebstahl von Cookies und authentication tokens → Identitätsdiebstahl ▫ Abfangen von Tastatureingaben ▫ Phishing-Attacken vorbereiten ▫ Bildschirminhalt verändern ▫ Web2.0 Entwicklung Ajax → Wurm	■	M 5.115 M 2.83
Cross Site Reference Forgery (XSRF/CSRF)	▫ Basiert auf Cross Site Scripting ▫ Bedient sich eines Opfers das berechtigten Systemzugriff hat ▫ HTTP-Request	Implementierungsfehler, Designfehler in der Anwendung, fehlende Data Validation, Semantikfehler	▫ Angreifer erlangt Zugriff auf den Server 1. nur Leserechte (GET) 2. auch Schreib- & Löschrechte (Fehler bei der Konfiguration des Webserver) ▫ Integrität von Daten ▫ Erscheinungsbild (Preise von Produkten ändern) ▫ Zugangsdaten	■	M 2.83
Session Riding (Session Fixierung)	▫ Basiert auf Cross Site Scripting ▫ Unterschieben von präparierte URL ▫ Session (Klonen) ▫ Cookie Diebstahl	Implementierungsfehler, Designfehler in der Anwendung, fehlende Data Validation, Semantikfehler	▫ Identitätsdiebstahl (Benutzer) ▫ Webanwendung benutzen und konfigurieren ▫ Angreifer kann mit den "Rechten" des Benutzers agieren	■	M 2.83 G 5.89

Tabelle: Kriterienkatalog auf Anwendungsebene (2) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

Titel	Kriterium (Schwachstelle)	Ursache	Mögliche Schäden	Gefährdung	BSI
"Logische" Überprüfung von Bestellungen	▫ Überflutung des E-Shops mit fiktiven Bestellungen (z.B. >10.000)	Implementierungsfehler, Designfehler in der Anwendung, Logikfehler	▫ Wirtschaftlicher Schaden ▫ Auslastung des E-Shops	■	M 2.83
Preisgabe von Informationen	▫ Analyse von Fehlermeldungen ▫ Systeminformationen ▫ Überversorgung eines Benutzers mit Systeminformationen	Implementierungsfehler, Designfehler in der Anwendung, Konfigurationsfehler	▫ Vorbereitung von Angriffen ▫ Verlust der Systemintegrität	■	M 2.83
Schwache Passwort – Wiederherstellungsfunktion	▫ Ermöglicht einem Angreifer die unerlaubte Erlangung eines Passwortes bzw. dessen Änderung ▫ Bsp: Passwort vergessen??	Implementierungsfehler, Designfehler in der Anwendung, Konfigurationsfehler	▫ Unerlaubter authorisierter Zugang zu einem System ▫ Ausschließen von Benutzern durch Passwortwechsel	■	M 2.83
Suchmaschinen Hacking	▫ Die Indizierung der Suchmaschine wird unterdrückt	Implementierungsfehler, Semantikfehler	▫ Webseiten werden nicht mehr gefunden	■	M 2.83
URL-Weiterleitungen	▫ Ausnutzen von automatischen Weiterleitungen (Redirects)	Semantikfehler, Logikfehler, Implementierung, fehlende Data Validation	▫ Webseite können nicht mehr erreicht werden, ▫ Weiterleitung auf schadhafte Webseiten	■	M 5.102

Tabelle: Kriterienkatalog auf Anwendungsebene (3) Gefährdung eines E-Commerce Unternehmen (■) = niedrig, (■) mittel, (■) hoch
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

4.2.4 Qualitätsmerkmale

Der zu erarbeitende Kriterienkatalog kann mit nur einem geringen Aufwand durch Kriterien erweitert werden die eine Webseite auf Qualitätsmerkmale überprüfen. Anhand bestimmter Merkmale kann eine E-Commerce-Webpräsenz auf Service- und Komfortmerkmale hin überprüft werden. Somit wird neben der reinen technischen Güte einer Webpräsenz auch die Einhaltung von Qualitätsmerkmalen gewährleistet. Dies bringt einem Kunden einen direkten Gewinn, da der Umgang mit der Webseite erleichtert wird. Ferner werden durch Qualitätsmerkmale mögliche Schwachstellen schon im Vorfeld verhindert. Als Qualitätsmerkmale werden Kriterien betrachtet die nicht technischem Ursprunges sind. Qualitätsmerkmale sind zum Beispiel:

- Die nicht lineare Vergabe von Datei- und Verzeichnisnamen. Dem ersten Anschein nach erhöht diese Maßnahme den Aufwand für die Pflege der Webpräsenz, da für den Webprogrammierer die Arbeit erschwert wird. Dieser Nachteil wird aber dahingehend wieder aufgehoben, dass durch die unstrukturierte Namensvergabe einem Angreifer nicht die Möglichkeit gegeben wird, Namen herzuleiten und sich einen Informationsvorteil zu verschaffen.

- Eine Überprüfung, dass eine Webseite alle notwendigen rechtlichen Kriterien erfüllt. Das bedeutet, dass an allen notwendigen Stellen einer Webpräsenz zugehörige Belehrungen und Bedingungen erwähnt werden. Dies kann mit nur einem geringen Aufwand automatisiert durch die Überprüfung bestimmter "Textphrasen" wie AGBs, und rechtlicher Belehrungen überprüft werden.

- Für die Qualität einer Webpräsenz ist wichtig, dass hinter jedem Links auch die Seite erscheint, die angekündigt wurde. Unter diesen Punkt fallen auch "tote Links“. Diese würden bei einem Kunden einen negativen Eindruck hinterlassen.

- Anmerkungen auf einer Webseite, wann das letzte Update der Webseite durchgeführt wurde, sind für Besucher wichtig. Anhand dieser Anmerkung kann der Besucher der Webseite erkennen ob seit seinem letzten Besuch einer Veränderung an der Webpräsenz vorgenommen wurden. Dies erspart ihm Zeit, da von ihm nicht eigenständig nach Updates gesucht werden muss.

- Der Quelltext der Webpräsenz wird dahingehend untersucht, ob Metadaten-Informationen für Suchmaschinen existieren. Ohne diese Metadaten würde die Webpräsenz nur unzureichend von den Suchmachschienen als Suchergebnis wiedergegeben werden.

Titel	Qualitäts-Kriterium	Ursache	Mögliche Schäden	Gefährdung	BSI
Datei- und Verzeichnisnamen	▫ Automatisierte Überprüfung auf nicht lineare Bezeichner von Dateien und Verzeichnissen	Erraten von Dateien o. Verzeichnissen	▫ Verlust der Systemintegriät	■	M 2.175
Rechtliche Kriterien	▫ Automatisierte Überprüfung mittels Trusted Shops Praxishandbuch-Kriterien	Einhaltung rechtlicher Bestimmungen	▫ Rechtliche Abmahnungen, finanzielle Strafen	■	
Tote Links	▫ Automatisierte Überprüfung der Webpräsenz auf tote Links	Inkonsistente Programmierung, Aktualisierung	▫ Imageverlust ▫ Umsatzverlust	■	
Vermerk über Update	▫ Automatisierte Überprüfung der Webpräsenz ob ein Vermerk über das letzte Update vorhanden ist	Qualitätsmerkmal gegenüber dem Kundem	▫ Imageverlust ▫ Umsatzverlust	■	
Metadaten	▫ Überprüfung der Metadaten einer Webpräsenz auf Suchmaschienenkriterien	Ranking der Webseite	▫ Webseite wird weniger häufig besucht	■	

Tabelle: Kriterienkatalog ausgerichtet auf Qualitätsmerkmale (1) (■) = Qualitätsmerkmal
[HK_97], [BSI_06] BSI: M = Maßnahmenkatalog, G = Gefährdungskatalog

4.3 Gewichtung der Kriterien

Eine Gewichtung der in dem vorhergehenden Kapitel 4.2 vorgestellten Kriterien erfolgt nach drei Merkmalen. Zuerst wird der Grad ihrer *Automatisierbarkeit* begutachtet. Jedes Kriterium besitzt spezifische Eigenschaften, die sich auf seine Automatisierbarkeit auswirken. Zusätzlich kann durch den Grad der Automatisierung einer Überprüfung die Aussagengüte leidet. Als zweiter Gesichtspunkt wird das jeweils bestehende *Gefährdungs- und Schadenspotential* für ein E-Commerce-Unternehmen betrachtet. Der dritte Gesichtspunkt besteht in einer Einstufung nach *OWASP* Top Ten. Es wird die Verbreitung und die Wahrscheinlichkeit Opfer einer derartigen Attacke zu werden, mit in die Gewichtung einbezogen. Ziel ist es, diejenigen Kriterien hervorzuheben, die sowohl technisch realisierbar als auch unerlässlich für eine technisch sichere Webseite sind. Bei der Auswahl der Kriterien muss mit in Betracht gezogen werden, dass nach einer Überprüfung eines Kunden, diesem ein Zertifikat oder zumindest eine Aussage über die Sicherheit seines Webauftrittes gemacht werden soll. Eine Aussage wie "Die geprüfte Webseite ist technische Sicher" ist unrealistisch, da dies nie erreichbar sein wird. Folglich wäre eine solche Aussage unzutreffend. Hingegen kann nach einer Untersuchung, in der nach den OWASP Top Ten Kriterien geprüft wurde, eine Aussage wie folgt gemacht werden: "Diese Webseite erfüllt die Technische Sicherheit nach den OWASP Top Ten Kriterien". Diese Aussage kann mit gutem Gewissen erfolgen, da die OWASP Top Ten Kriterien überwiegend aus sogenannte "Low-Hanging Fruits" Kriterien bestehen. Low-Hangig Fruits sind Sicherheitskriterien, die unter absehbarem Aufwand erkannt und behoben werden können.

4.3.1 Automatisierbarkeit

Ein Kriterium bzw. Schwachstelle, die automatisiert Überprüft werden soll, muss streng normierbar sein. Des Weiteren darf die Güte einer Überprüfung nicht mit dem Grad der Automatisierung leiden. Dies wäre zum Beispiel der Fall, wenn eine Überprüfung eines Kriteriums zwar automatisierbar ist, aber durch diesen Schritt nicht mehr alle Schwachstellen des Kriteriums aufgedeckt würden. Die gewählte Einteilung erfolgt nach weniger gut automatisierbaren Kriterien (+), gut automatisierbaren Kriterien (++) und sehr gut automatisierbaren Kriterien (+++).

Sicherheitskriterien der Protokollebene erfüllen die oben genannte Voraussetzung am besten und bieten Aufgrund dessen die Vorraussetzungen für eine stark automatisierte Überprüfung. Viele Schwachstellen der Protokollebene beruhen auf offen liegenden Ports (SYN-Flooding) oder Designfehlern im Protokoll. Offen liegende Ports lassen sich ohne viel Aufwand automatisch mit einem Port-Scanner überprüfen. Protokoll-Designfehler können

ebenfalls automatisiert überprüft werden, da sie im Wesentlichen unabhängig von semantischen Zusammenhängen in IT-Systemen sind. Zusätzlich zeichnen sich die Kriterien der Protokollebene bei einer Überprüfung durch eine hohe Trefferrate bei der Aufdeckung von Schwachstellen aus. Durch die normierte Struktur der Protokolle und ihrer längeren unveränderten Betriebsdauer sind sehr viele Fehlerquellen statisch und bekannt. Aufgrund dessen kann eine höhere Prozentzahl aller Schwachstellen aufgedeckt werden. Diese Eigenschaften führen bei Kriterien der Protokollebene zu einer Aufwertung der Automatisierbarkeit.

Kriterien, die der Dienst- oder der Anwendungsebene zugeordnet werden, müssen mit einem größerem Aufwand automatisiert werden. Die Struktur der beiden Ebenen unterliegt aufgrund kurzer Betriebsphasen der verwendeten Dienste und Anwendungen einem ständigen Wandel. Anders als in der Protokollebene werden hier Dienste und Softwarekomponenten in kurzen Zeitintervallen durch neue ersetzt oder durch neue Komponenten erweitert. Dieser Vorgang hilft bei der Schließung von bekannten Sicherheitslücken. Im gleichen Schritt können aber neue und bisher unbekannte Schwachstellen geschaffen werden. Diese neuen Schwachstellen lassen sich nur durch eine fortwährende Aktualisierung des Kriterienkataloges berücksichtigen. Kriterien der Dienst- und Anwendungsebene erhalten eine Aufwertung wenn sie sich durch bekannte Codezeilen oder andere Mechanismen überprüft werden können. Bei SQL-Injection-Schwachstellen ist zum Beispiel eine zuverlässige Überprüfung ohne viel Aufwand durch das Einfügen von bekannten schadhaften Codebeispielen möglich. Aufgrund der Vielzahl von Eingabemöglichkeiten werden aber nicht alle denkbaren Eingaben überprüft werden können. Folglich wird SQL-Injection bei der Automatisierung mit einem mittleren Wert von zwei Kreuzen bedacht. Hingegen kann das Network Information System (NIS) nur mit viel Aufwand und vor allen Dingen nur mit einer eingeschränkten Aussagegüte auf Schwachstellen getestet werden, erhält folglich nur eine Kreuz bei der Automatisierbarkeit.

Kriterien, die Schwachstelle von Logikfehler betrachten, werden nur schwer automatisiert werden können, da die Schwachstellen abhängig von internen Betriebsabläufen sind. Diese internen Abläufe können aber bei einer automatisierten Überprüfung nicht berücksichtigt werden. Kriterien dieser Art werden folglich abgewertet. Dies betrifft zum Beispiel das Kriterium zur Überprüfung von Benutzereingaben auf logische Korrektheit.

Kriterien die eine Webanwendung auf Qualitätsmerkmale hin untersuchen, lassen sich ohne viel Aufwand und in einer hohen Güte bei der Trefferquote automatisieren. Dies liegt an dem Umstand, dass diese Kriterien strukturierbar und weitestgehend unabhängig von der jeweiligen Webanwendung sind. Ein Qualitätsmerkmal, dass sich gut automatisieren lässt ist zum Beispiel eine Überprüfungen auf tote Links oder rechtliche Kriterien.

4.3.2 Schadenspotential

Die Bewertung des Schadenspotentials findet ebenfalls in drei Stufen statt. Hierbei werden zum Beispiel Schäden an dem äußeren Erscheinungsbild einer Webseite als weniger kritisch bewertet. Der Verlust der Datenintegrität oder im schlimmsten Fall der des Datenverlustes erhält eine höhere Bewertung. Die Basis für die Einteilung der Kriterien erfolgt sowohl nach den in den Kriterienkatalogen aufgezählten möglichen Schäden für einen E-Commerce-Anbieter, als auch nach der Wahrscheinlichkeit das Opfer einer solchen Attacke zu werden. So sind zum Beispiel Schäden von Angriffen auf der ARP-Ebene für einen E-Commerce-Anbieter sehr schadhaft. Gleichzeitig aber sind Angriffe dieser Art sehr unwahrscheinlich, da das ARP Protokoll nur im internen Netzwerk (Intranet) funktioniert. Ein Angriff kann folglich nur aus dem internen Netz kommen und wird folglich als "niedrig" eingestuft.

4.3.3 Verbreitung nach OWASP 

Das Open Web Application Security Project (OWASP) ist in seinem Aufbau an das Wikipedia Projekt angelehnt. Es besteht aus einer Sammlung von Katalogen, Konzepten und Maßnahmen für sichere Webanwendungen. Das hauptsächliche Themenfeld der veröffentlichten Artikel beschäftigt sich mit der Erforschung und der Bekämpfung von Fehlerquellen in unsicherer Software. Das Projekte veröffentlicht auf seiner Webseite eine "Top Ten-Liste" der aktuell brisantesten Sicherheitsmängel von Webanwendungen. Die Liste wird in regelmäßigen Zeitabständen aktualisiert und bietet aufgrund der zeitnahen Aussage eine gute Basis, um zu der Gewichtung der Sicherheitskriterien beizutragen. Kriterien die in der Liste enthalten sind werden mit 2 Punkten aufgewertet. Dies dient dazu, das aktuelle Schadenspotential eines Kriteriums zu unterstreichen. Da gewisse Kriterien nur über einen bestimmten Zeitraum ihre höchste Verbreitung besitzen, ist diese Gewichtung besonders wichtig [OWASP]. Weiterhin tragen die OWASP Kriterien dazu bei, dass im Anschluss an eine durchgeführte Überprüfung eine Aussage wie folgt gemacht werden kann: "Diese Webpräsenz wurde nach den OWASP Top Ten Kriterien überprüft". Aufgrund dessen, dass sich das OWASP Projekt im IT-Sicherheitsbereich stark etabliert hat, kann mit dieser Aussage das Untersuchungsergebnis fundamentiert werden.

OWASP Top Ten März 2007:

1. Cross-Site-Scripting
2. Injection Flaws
3. Insecure Remote File Include
4. Insecure Direct Object Reference
5. Cross-Site Request Forgery (CSRF)
6. Information Leakage and Improper Error Handling
7. Broken Authentication and Session Management
8. Insecure Cryptographic Storage
9. Insecure Communications
10. Failure to Restrict URL Access

Kriterien / Schwachstellen	Automatisier-barkeit	Schadenspotential (E-Commerce)	OWASP (+2 Punkte)	Σ (Punkte)
Protokollebene				
ARP-Ebene	++	—		3
TCP/IP-Ebene	+++	—		4
ICMP-Ebene	+++	—		5
Spoofing	+	—		3
Dienstebene				
DNS-Angriffe	++	—		3
Network Information System (NIS)	+	—		2
FTP	+	—		2
SQL-Injection	++	—	Rang 2	7
Nicht benötigte Dienste	+	—		2
Einsatz von SSL/TLS/HTTPS	++	—	Rang 9	6
Monitoring und Patching	++	—		4
Systemkonfiguration	++	—	Rang 8	6
Serverkonfiguration	++	—	Rang 8	6
Web Application Security Tools	++	—		3
Anwendungsebene				
Sendmail	+	—		2
Mail Spoofing	+	—		2
Verteilter, koordinierter Angriff	+	—		2
Erraten von Passwörtern	+	—		2
Viren	+	—		2
Trojanische Pferde	+	—		2
Würmer	++	—		4
Java/PHP (Script-Sprachen)	+	—	Rang 3	6
Cross-Site-Scripting (XSS)	++	—	Rang 1	7
Cross-Site Reference Forgery (XSRF/CSRF)	++	—	Rang 5	6
Session Riding (Session Fixierung)	++	—	Rang 7	6
"Logische" Überprüfung von Bestellungen	+	—		2
Preisgabe von Informationen	++	—	Rang 6	6
Schwache Passwort – Wiederherstellungsfunktion	++	—		3
Suchmaschinen Hacking	++	—		3
URL-Weiterleitungen	+	—	Rang 10	4
Qualitäts-Kriterien				
Datei- und Verzeichnisnamen	+++	—		4
Rechtliche Kriterien	+++	—		4
Tote Links	+++	—		4
Vermerk über Update	+++	—		4
Metadaten	+++	—		4

Tabelle 9: Gewichtung der Kriterien

+ = 1 Punkt, ++ = 2 Punkte, +++ = 3 Punkte
— = 1 Punkt, — = 2 Punkte, — = 3 Punkte OWASP = 2 Punkte

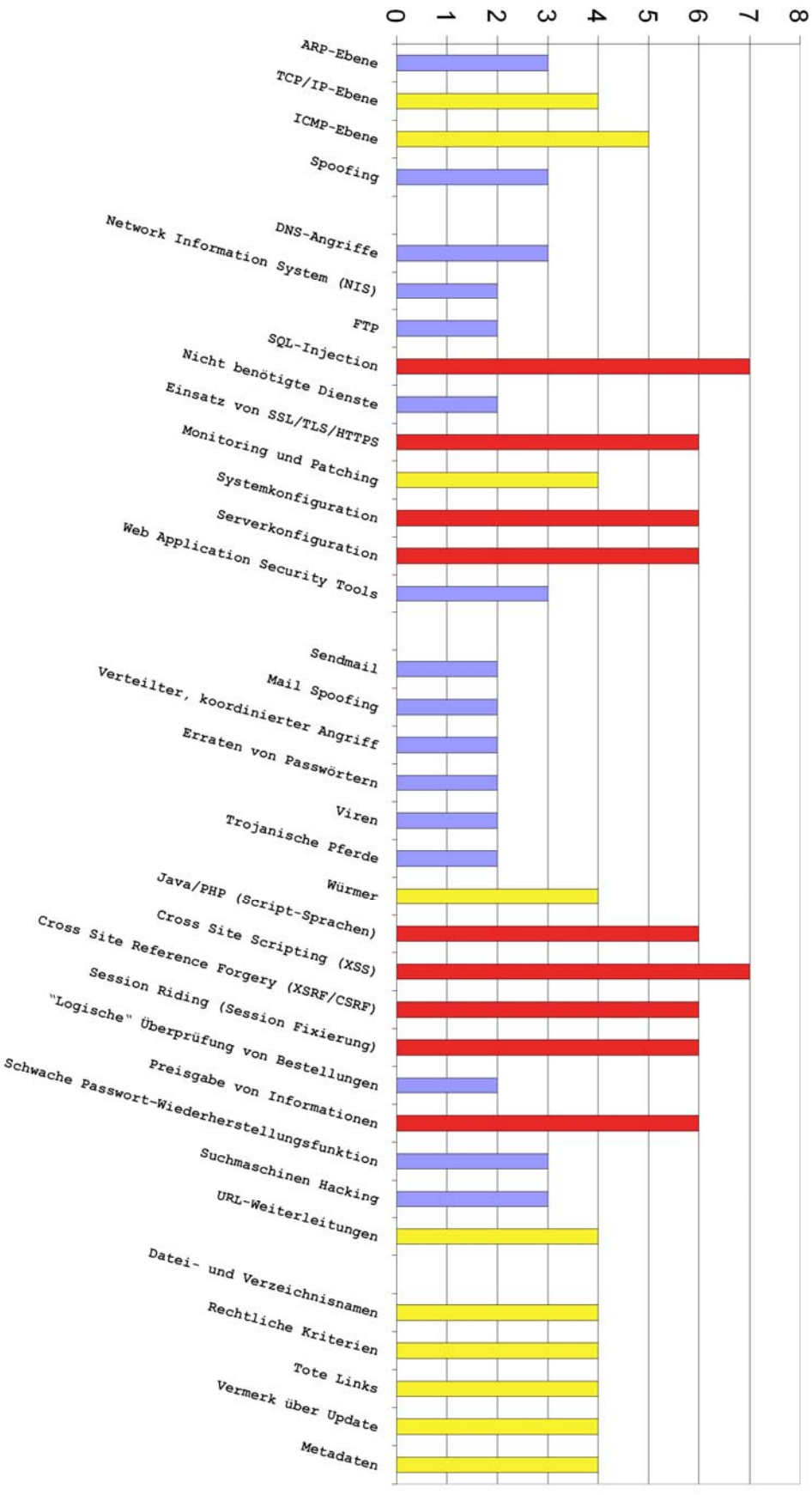

Abbildung 26: Gewichtung der Kriterien

5. Automatisierung des Kriterienkataloges

Die in Kapitel Vier erarbeiteten Kriterien sollen in Form einer automatisierten Durchsuchung einer Webseite, von einem Computerprogramm oder einem darauf spezialisiertem Tool, abgearbeitet werden. Die manuelle Durchsuchung einer Webseite ist Aufgrund des Umfangs des Kriterienkataloges und dem damit verbundenen Zeitaufwand nicht zu empfehlen, prinzipiell aber möglich. Für eine Automatisierung stehen zwei Szenarien für eine Realisierung zur Verfügung. Es kann entweder auf einen sogenannten Portscanner oder auf einen "(Application) Vulnerability Scanner" zurückgegriffen werden. Die Idee, die hinter einem Vulnerability Scanner steht, ist die Sicherheit von (Web-) Anwendungen automatisch zu überprüfen und gefundene Fehler in einem Report festzuhalten.

Portscanner

Portscanner untersuchen ein Zielsystem, bzw. darauf abgelegte Webanwendungen, auf der Netzwerkebene (Portebene) nach Schwachstellen. Eine E-Commerce-Webanwendung muss zu großen Anteilen frei über das World Wide Web erreichbar sein. Dies ergibt sich aus der Notwendigkeit Daten mit Kunden auszutauschen oder Informationen für Geschäftsvorgänge unterschiedlichster Art bereitzustellen. Daher wird der üblicherweise über Port 80 laufende Datenverkehr zwischen Web- und Datenbank-Servern sowie WWW-Anwendungen durch Sicherheitssoftware zugelassen. Aktuellen Datenbank-Systeme, wie zum Beispiel Microsoft SQL Server, Oracle und MySQL, können über weitere festgelegte Ports erreicht werden. Diese Ports müssen für den legitimen Datenverkehr geöffnet bleiben und stellen hierdurch eine wesentliche Schwachstelle dar [AC]. Böswillige Angreifer können über diese speziellen Ports versuchen, eine direkte Verbindung mit der jeweiligen Anwendung herzustellen, um die Sicherheitskontrollen eines Betriebssystems zu umgehen. Ein Portscanner ermöglich die Aufdeckung von derartigen Schwachstellen in einem System. Für das Ausmaß des zu erarbeitenden Kriterienkataloges ist der Funktionsumfang eines Portscanners aber nicht ausreichend. Es wird viel mehr ein Web-Application Scanner benötigt, da dieser ein breiteres Feld an Schwachstellen untersuchen kann.

Vulnerability Scanner (Web-Application Scanner)

Web-Application Scanner bieten, aufgrund ihres höheren Funktionsumfangs, die geeignete Basis für die Automatisierung des Kriterienkataloges. Die Scanner sind Anwendungen die Zielsysteme automatisiert auf bekannte Sicherheitslücken hin untersuchen. Der Scanner

bedient sich dabei einer Datenbank, die Informationen zu diversen bekannten Sicherheitsproblemen enthält. Das können zum Beispiel Sicherheitsprobleme bei verwendeten Diensten, der Handhabung von Dateien, Passwort- und Benutzerrichtlinien, offenen Ports und bei fehlenden Softwareupdates sein. Web-Application Scanner werten im Gegensatz zu Portscannern nicht nur statisch eine fest definierte Klasse von Diensten am Zielsystem aus. Sie prüfen stattdessen aktiv und dynamisch auf vorhandene Schwachstellen des verwendeten Betriebssystems, der angebotenen Anwendungen und der jeweiligen Hardware. Wie die Web-Application Scanner dies erreichen, wird in dem nächsten Unterkapitel vorgestellt [WIV].

5.1 Web-Application Scanner

Zum aktuellen Zeitpunkt haben sich eine Vielzahl von Web-Application Scanner auf dem Markt etabliert. Darunter vertreten sind Produkte von staatlichen und gewerblichen Stellen. Ein Produkt aus der Quelle des Staates kommt von dem Bundesamtes für Sicherheit in der Informationstechnik mit dem BSI-BOSS Tool. Daneben bestehen gewerbliche Produkte von Anbietern wie Acunetix(WVS), eEye (Retina) und Open Source Projekte wie CAL9000. Web-Application Scanner können passiv oder aktiv agieren. Passive Scanner schließen aus Eigenschaften, die im aktuellen Zustand des Systems vorhanden sind, auf die Anwesenheit einer Schwachstelle (Signaturen). Aktive Scanner werfen Schwachstellen durch deren Ausnutzung auf, wodurch sie Gefahr laufen dem zu testenden System einen Schaden zuzufügen. Würde sich zum Beispiel ein aktiver Scanner selbstständig verbreiten, so könnte er durch dieses Verhalten die Eigenschaften eines Wurms aufzeigen. Scanner können zudem Host-basiert oder Netzwerk-basiert sein. Host-basierte Scanner können nur das System untersuchen, auf dem sie laufen. Netzwerk-basierte Scanner sind hingegen in der Lage ein ganzes Netzwerk von Computern über die Netzwerkverbindungen auf Schwachstellen hin zu testen. Heutige Scanner sind überwiegend passiv. Lediglich die Anwesenheit mancher Denial-of-Service- oder Passwortschwächen werden durch aktive Tests geprüft. Solche Schwächen können nicht durch passive Tests nachgewiesen werden, da beispielsweise die Passwörter verschlüsselt auf dem Host gespeichert werden [WTS].

Web-Application Scanner können über zwei unterschiedliche Methoden Schwachstellen in einem System aufdecken. Das eine Feld nehmen die zuvor kennen gelernten Port-Scanner (Netzwerk-Scanner) ein, die vornehmlich Schwachstellen auf der Portebenen (Netzwerkebene) scannen. Das größere Feld nehmen die (Web-)Application Scanner ein, die gezielt Anwendungen auf Schwachstellen hin untersuchen.

Zwei wesentliche Probleme müssen bei der Qualitätssicherung eines Application-Scanners beachtet werden:

Zum Einen kann eine fehlende Vollständigkeit der Prüfkriterien vorliegen. Sogenannte passive Scanner verfügen über eine Datenbank mit statischen Signaturen, die mit dem aktuellen Zustand des zu testenden Systems abgeglichen werden. Demzufolge ist ein Application-Scanner wie ein Virenscanner ständig zu aktualisieren. Aufgrund der sich stets weiterentwickelnden IT-Systeme können Application-Scanner nicht für alle Schwachstellen eine Signatur besitzen. Die Signatur kann zudem fehlerhaft sein. So können unvollständige Ergebnisse eines Testdurchlaufs entstehen. Im Rahmen einer Diplomarbeit an der Universität Hamburg durchgeführten Untersuchung aus dem Jahr 2001 fand keiner der eingesetzten Scanner alle Schwachstellen. Einige Scanner meldeten nicht alle Schwachstellen, die in einem Testsystem präpariert wurden. Andere Scanner meldeten zu viele Schwachstelle [FJ].

Zum Anderen können Web-Application Scanner falsche Positivergebnisse von gefundenen Schwachstellen erzeugen (false positives). Dies ist dann der Fall, wenn ein Scanner eine Schwachstelle meldet, die eigentlich nicht in dem System vorhanden ist. Dieser Umstand ergibt sich aus den Signaturen, anhand derer die Schwachstellen erkannt werden. So lesen zum Beispiel die meisten Scanner nur das Banner einer Antwort-Nachricht einer Webanwendung aus, um Informationen über die verwendete Software und deren Version zu erhalten. Aus diesen Informationen schließen sie wiederum welche Schwachstellen auf die Anwendung zutreffen. Ob die Software wirklich installiert ist, prüft der Scanner nicht nach. Damit kann der Scanner durch ein manipuliertes Banner leicht getäuscht werden. Falsche Positivmeldungen führen zu einem fehlerhaften Ergebnisberichten und mindern das Vertrauen des Benutzers in das erzielte Ergebnis der Untersuchung. Hierbei muss herausgestellt werden, dass obwohl automatisierte Checks sicherlich nie eine hundertprozentige Erkennung von sicherheitsrelevanten Problemen an Servern haben werden, kann doch zumindest ein gewisses Sicherheitsbewusstsein bei den verantwortlichen Entscheidungsträgern hervorgerufen werden.

Das Vorgehen eines Web-Application Scanners für eine Untersuchung sieht wie folgt aus: Auf einem beliebigen Computer (Client), der über eine Internetverbindung verfügt, wird die Scanner-Software ausgeführt und auf die zu untersuchende Webanwendung angesetzt. Der Scannvorgang durchläuft hierfür 4 Phasen:

1. Aufbau-Analyse der Webseite:

Ein Web-Application Scanner bewegt sich durch die zu testende Webanwendung und "klickt" jeden vorkommenden Link und Button an. Dabei werden, soweit vorhanden, auch

Textfelder mit Testwerten ausgefüllt. Der Scanner erhält hierdurch ein umfassendes Wissen über den Aufbau der Webseite, den Funktionsweisen von Dialogfeldern und den Inhalte von Formularseiten.

2. Sicherheitsanalyse:

Im zweiten Schritt wird eine Sicherheitsanalyse der Webseite vorgenommen. Die gewonnenen Erkenntnisse werden in einer Datenbank hinterlegt. Erkenntnisse über die Webseite wären zum Beispiel: Typ und Version des Webservers, verwendete Technologien und Tools (SQL, JavaScript, PHP, ...), Verwendung von Cookies oder anderen Mechanismen zum Sessiontracking, Analyse der Form-Parameter einer Seite, insbesondere auch die Verwendung von HIDDEN-Variablen und eine Analyse von Kommentaren der Programmierer der Webseiten.

3. Penetrationstest:

Im Anschluss an die Analyse wird die Anwendung, unter Einbeziehung des in der vorangegangenen Phase gewonnenen Wissens, attackiert. Es wird ein realer Hacker-Angriff von dem Scanner simuliert. Das beinhaltet das Durchlaufen sämtlicher bekannter Schwachstellen der verwendeten Systeme, Testen auf liegengelassene Dateien (Backup-, Test-, Temporärdateien, usw.) und Parametermanipulationen, um die oben erwähnten Schwachstellen auszunutzen.

4. Reporting:

Die Ergebnisse des durchgeführten Scans werden in Report-Berichten zusammengefasst. Ein Report kann als eine einfache nummerierte Liste geschehen, die nur die schwerwiegendsten Sicherheitslöcher auflistet. Ein Report kann aber auch als eine detaillierte Beschreibungen ausfallen, in der auch Hinweise für die Behebung des jeweils aufgedeckten Problems gegeben werden[SECW].

Als Fazit kann festgehalten werden, dass die Automatisierung des Kriterienkatalogs über den Einsatz eines Web-Application Scanner realisiert werden kann. Open Source Projekte wie Nessus oder die Erweiterung des BSI-Tool (BOSS) können als Grundlage für den zu entwickelnden Web-Application Scanner gewählt werden. Auf dieser Grundlage basierend werden dann die eigenen Kriterien in das Tool integriert. Das folgende Kapitel stellt bestehende Scanner vor und vergleicht hierbei Open Source Projekte mit Tools von kommerziellen Anbietern. Der Schwerpunkt der Untersuchung liegt bei dem gebotenen Funktionsumfang (Anzahl von Prüfkriterien), der Erweiterbarkeit, der Güte der Reports, der Geschwindigkeit, der Handhabung und der Unterstützung zur Schließung der Schwachstellen. Zusätzlich werden bei kommerziellen Anbietern die anfallenden Kosten für das Produkt betrachtet.

5.1.1 BSI (BOSS-Nessus)

Die Open Source Software BOSS (BSI OSS Security Suite) baut im wesentlichen auf dem etablierten Vulnerability Scanner Nessus auf. Nessus ist ein bekannter Netzwerk- und Application Scanner für Linux- und Unixsysteme und arbeitet nach dem Klient-Server-Prinzip. Der Nessus Scanner wurde dahingehend erweitert, dass er mit der BOSS-Oberfläche und dem Security Local Auditing Daemon (SLAD) ausgestattet wurde. Der SLAD übernimmt die Steuerung der angebundenen lokalen Sicherheitssoftware [BOSS_A].

Ein Scan wird auf folgende Weise durchgeführt. Auf einem beliebigen Netzwerkrechner wird der Nessusserver (nessusd) gestartet. Anschließend kann der Scanner mittels eines Klient, entweder von einem lokalen oder einem entfernten Netzwerkrechner aus, mit dem Server verbunden werden. Mit dem Start des Servers werden automatisch die für einen Scan benötigten Plugins geladen. Mit diesen Plugins lassen sich diverse Sicherheitslücken eines Betriebssystems bzw. der Anwendungen, die auf dem zu scannenden Host laufen, finden. Eine spätere Erweiterung des Scanners mit neuen Prüfkriterien ist durch die Plugin-Architektur ohne viel Aufwand möglich. Die Plugins werden in der Nessus eigenen Skriptsprache "Nessus Attack Scripting Language" (NASL) erstellt. Derzeit existieren ca. 12.000 Plugins, die es ermöglichen auf verschiedenste Sicherheitslücken aller relevanten Betriebssysteme und Netzwerk-Produkte zu prüfen. Im zweiten Schritt kann daraufhin mit Hilfe des Klient eine Verbindung mit dem Server hergestellt und eine "Session" gestartet werden. In dieser Session werden die gewünschten Plugins, der Zielhost und andere Einstellungen eingetragen. Wurde der Scan auf einem Host ausgeführt, gibt der Nessus-Klient eine Übersicht über die offenen Ports und eventuell gefundene Sicherheitslücken aus.

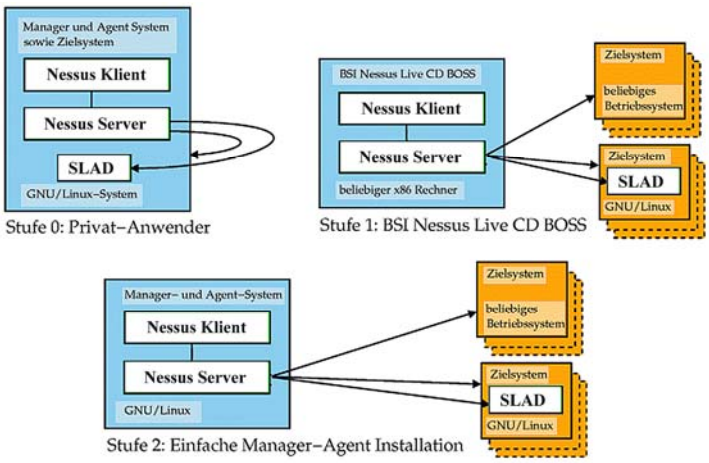

Abbildung 27: BSI-BOSS (Nessus) Stufe 0 bis 2 [BOSS_B]

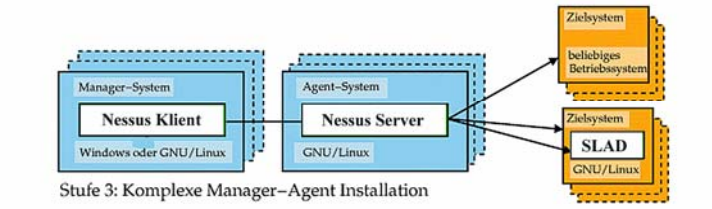

Abbildung 28: BSI-BOSS (Nessus) Stufe 3 [BOSS_B]

Die Benutzerfreundlichkeit wurde durch die BOSS-Oberfläche im Gegensatz zur Nessus-Oberfläche wesentlich erweitert. Durch den entwickelten SLAD verfügt Nessus jetzt über die Möglichkeit Ziel-Systeme auch intensiv von innen her auf Schwachstellen oder gar bereits erfolgreiche Angriffe zu prüfen. Durch die zentrale und vereinfachte Steuerung über den Klient wird die Auswertung und damit das organisationsweite Sicherheitsauditing fundamental vereinfacht. Aufgrund des Open Source Status der Nessussoftware und seiner Erweiterbarkeit bietet sich der Nessus Scanner als Plattform für den Kriterienkatalog der Firma Trusted Shops an.

Umfang der Prüfkriterien [DR]:

- Hintertüren (Backdoors)
- Denial of Service (DoS)
- Erlangung einer remote-Shell
- Windows: Benutzer-Verwaltung
- Network Information System
- Windows (generelle Konfiguration)
- Erlangung von Root-Rechten (remote)
- CGI Lücken
- Firewalls
- FTP
- Netware
- Port Scanner
- SNMP
- CISCO
- Nutzlose Dienste
- RPC
- Remote Dateizugriff
- SMTP Probleme

Zusammenfassung:

Das BSI-BOSS Tool bietet aufgrund seiner Erweiterbarkeit und seines Open-Source Status eine gute Basis für die Realisierung des Kriterienkatalogs aus dem vierten Kapitel. Das Tool arbeit, laut einer von Trusted Shops durchgeführten Untersuchung, stabil aber relativ langsam. Negativ fällt auf, dass das Tool hauptsächlich auf die Überprüfung von Schwachstellen auf der Netzwerkebene ausgerichtet ist. Somit ist es kein vollwertiger Web-Application Scanner. Zusätzlich sind viele der vorhandenen Plugins abhängig von der verwendeten Software. So gibt es zum Beispiel nicht ein universales Plugin mit dem sich Cross-Site-Scripting Schwachstellen aufdecken lassen. Vielmehr sind hierfür eine Reihe von Plugins vorgesehen, die alle individuell an die jeweils verwendete Softwareversion des Testsystems angepasst werden müssen. Dies macht eine automatisierte Überprüfung sehr aufwendig.

5.1.2 Nikto (Whisker)

Nikto ist ein Vulnerability-Scanner, der im Gegensatz zu Nessus nur für das Testen von Webservern eingesetzt werden kann. Nikto ist darauf spezialisiert bekannte Sicherheitslücken in Standard Webanwendungen, wie PHP, SQL, o.Ä. zu finden. Das Tool basiert auf LibWhisker. Whisker war früher ebenfalls ein Security-Scanner für Webanwendungen, wurde aber durch Nikto ersetzt. Nikto ist unter der Open Source Lizenz GPL (Gnu General Public Licence) veröffentlicht und ist somit kostenlos erhältlich und kann ohne Einschränkungen an persönliche Bedürfnisse angepasst werden.

Das Tool greift für einen Sicherheitsüberprüfung auf eine Datenbank zurück. Diese Datenbank enthält mehr als 3.100 schädliche Codefragmente und über 650 bekannte Schwachstellen von Webservern. Befindet sich auf dem zu testenden Webserver Anwendungen wie beispielsweise phpBB oder phpNuke ist Nikto das geeignete Tool für eine Überprüfung, da es für Standard Anwendungen optimiert ist. Nikto wurde in der Programmiersprache Perl geschrieben und ist es prinzipiell sowohl unter Unix Varianten als auch unter Windows oder OSX lauffähig. Es wird allerdings ein Perl Interpreter benötigt. Für Windows kann zum Beispiel ActiveState Perl verwendet werden. Zu betonen ist, dass Nikto jedoch nicht in der Lage ist, Sicherheitslücken in unbekannter (z.B. selbstentwickelter) Web-Software zu finden. Aufgrund dieses Missstandes ist er nur bedingt geeignet als Ausgangslage für den zu erarbeitenden Kriterienkatalog zu dienen [SEC].

Zusammenfassung:

Neben Nessus ist Nikto der einzige Open Source Vulnerability Scanner, der über eine ausbaufähige Basis verfügt. Positiv ist, dass er über eine ausgereifte "search engine", erweiterbare Prüfkriterien und eine grafische Benutzerschnittstelle verfügt. Negativ fällt allerdings auf, dass es sich um keinen klassischen Application-Scanner handelt. Folglich bietet er nur eine eingeschränkte Funktionalität. Das bedeutet, dass nicht alle Kriterien überprüft werden können [HH].

5.1.3 Acunetix (WVS)

Acunetix ist ein Anbieter von Web-Auditing-Software und bietet einen On-Demand-Dienst (Acunetix Site Audit) an. Mit dem Dienst können Unternehmen die Sicherheit ihrer Webseite und Webanwendungen von Acunetix überprüfen lassen. Für diesen Service benötigt ein Kunde keine weiteren eigenen firmeninternen Ressourcen. Zusätzliche Kosten für spezielle Hardware, Installation, Administration und Wartung der Audit-Software fallen nicht an. Der Service wird von Acunetix Mitarbeitern mit Hilfe des Acunetix Web Vulnerability Scannern (WVS) durchgeführt. Das Produkt richtet sich aufgrund der Preisgestaltung an größere Unternehmen, die einen automatisierten, nichtsdestotrotz aber auch individuell anpassbaren, Vulnerability Scan ihres Unternehmens wünschen. Obwohl Acunetix mit dem angebotenen Application Scanner vom Funktionsumfang grundsätzlich ein ähnliches Produkt wie das von Trusted Shop geplante anbietet, ist das dahinter liegende Konzept ein anderes. Acunetix spricht sowohl durch die gewählte Preisgestaltung, als auch durch den komplexen Aufbau der Scansoftware andere Kunden als Trusted Shops an. Dahingehend ist eine direkte Gegenüberstellung der Konzepte nur eingeschränkt möglich.

Leistungsumfang von Acunetix SiteAudit:

- Primär ermöglicht das Tool ein sofortige Beurteilung der Sicherheit einer Webseite, in dem es einen Umfassenden Bericht erstellt.
- Des weiteren bietet es eine Überprüfung einer Webseite, in Hinblick darauf, ob diese vor Web-Angriffen geschützt ist.
- Die Überprüfung der Webseite insbesondere auf aktuelle SQL-Injection-, Cross-Site-Scripting- und anderen Schwachstellen.
- Eine Überprüfung der Sicherheit von Webanwendungen wie Einkaufswagen, Formulare und dynamischen Inhalte von Webseiten.
- Einen vollständigen und dynamischen Scan der Webseiten und -Anwendungen (inkl. JavaScript/AJAX-Web2.0) auf bestehende Sicherheitslücken.

Umfang der Prüfkriterien (entnommen von der Herstellerwebseite)**:**

- Versions-Check (Webserver-Software, PHP-Check, …)
- CGI-Tester (unsichere http-Methoden of Webserver, PUT, DELETE, ...)
- Parameter-Manipulation-Kontrolle (XSS, SQL-Injection, Code Ausführung, ...)
- Multi-Request Parameter-Manipulation-Kontrolle (Blind SQL, ...)
- Eingabevalidierung
- Authentifizierungsangriffe
- Pufferüberläufe
- Google Hacking Database (GHDB)

- Datei-Checks (Skriptfehler, Backupdateien, Protokolle, ...)
- Verzeichnis-Checks (Aufspüren vertraulicher Verzeichnisse, ...)
- Textsuche (E-Mail Adresse, Offengelegter Code, ...)

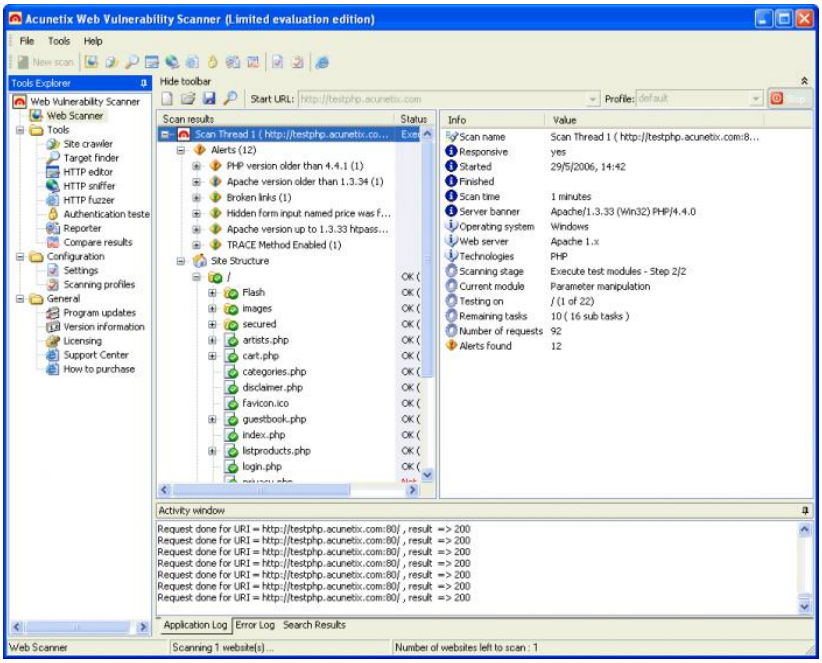

Abbildung 29: Acunetix Web Vulnerability Sanner [AC]

Durchführung eines Sicherheits-Audit [AC]:

1. Im ersten Schritt führt der Scanner eine Bestandsanalyse der gesamten Webpräsenz durch ("Crawling"). Dazu folgt er allen Links, die auf der Webseite und aus der Datei "robots.txt" entnommen werden können, sofern diese Datei vorhanden ist. In der Benutzerausgabe werden parallel zu der Struktur der Webseite detaillierte Informationen zu jeder gefundenen Datei angezeigt.

2. Im Anschluss an die Bestandsanalyse startet der Scanner automatisch eine Reihe von Schwachstellenüberprüfungen. Hierzu wird jede ermittelte Webseite mit Testangriffen überprüft, die realen Hacker-Angriffen nachempfunden sind. Der Scanner untersucht Seiten auf Eingabefelder und probiert in diesen Feldern unterschiedlichste Eingabekombinationen aus.

3. Wurden Schwachstellen identifiziert, zeigt der Scanner diese dem Benutzer über eine Schnittstelle für Warnmeldungen an. Jede dieser Warnungen liefert Informationen zur aufgedeckten Sicherheitslücke und bietet soweit möglich dem Benutzer eine Gebrauchsanweisung zu deren Behebung.

4. Ist ein Scan abgeschlossen, kann das Ergebnis zwecks einer späteren Analyse oder zum Vergleich mit anderen Scans in einer Datei gespeichert werden. Zudem wird ein zusätzliches Report-Tool angeboten, dass die Möglichkeit bietet, einen zusammenfassenden Scanbericht zu erstellen.

Das Produkt steht in 3 Versionen zur Verfügung:

Die Small Business Version eignet sich für Unternehmen, die ihre Webseiten auf einem einzelnen Server betreiben. Mit dieser Version kann nur die eigene Webseite gescannt werden, nicht jedoch Webseiten von Kunden oder anderen Dritten.

Die Enterprise Version eignet sich für größere Unternehmen, die mehr als eine eigene Webseite betreiben. Mit dieser Version können beliebig viele Webseite gescannt werden, nicht jedoch Webseiten von Kunden oder anderen Dritten.

Die Consultant Version eignet sich für Dienstleister, die Services in den Bereichen Penetration-Testing und Schwachstellenbewertung anbieten. Diese für Sicherheitsberater, Web-Entwickler und ISPs konzipierte Version erlaubt das Scannen einer uneingeschränkten Anzahl an Webseiten ihrer Kunden. Im Lieferumfang enthalten ist ein Vulnerability Editor zum Erstellen angepasster Bewertungstests und Sicherheitsberichte. Berichte können zudem mit dem Firmenlogo des Dienstleister versehen werden [AC].

Zusammenfassung:

Acunetix bietet kein reines Tool für die Überprüfung einer Webseite an. Vielmehr handelt es sich um ein Paket von Dienstleistungen. Dieses Paket übernimmt die Überprüfung der gesamten Webseite und der damit verbundenen Komponenten. Ein Kunde beauftragt Acunetix mit der Überprüfung seiner Webseite und bekommt nur die Ergebnisse der Untersuchung mitgeteilt. Die eigentliche Überprüfung findet bei Acunetix und nicht bei dem Kunden statt. Von der Preisgestaltung richtet sich das Tool an größere Firmen.

Acunetix Web Vulnerability Scanner [AC]	
Small Business Version	1.705 €
Enterprise Version	3.000 €
Consultant Version	5.700 €

Tabelle 10: Accunetix Preistabelle

5.1.4 SPI Dynamics (WebInspect)

Die Firma SPI Dynamics bietet mit der siebten Version des WebInspect-Tools einen kommerziellen Web-Application Scanner an. Dieser Scanner verfügt über einige innovative Neuerungen im Bereich der Web-Application Scanner. Der Scanner bewertet das Risikopotential gefährdeter Webanwendungen, indem bekannte und unbekannte Schwachstellen innerhalb der Webanwendung identifiziert werden. Die Eigenschaft die den Scanner auszeichnet, ist die Fähigkeit, bei der Durchführung einer Überprüfung dynamisch zu agieren. Eine Überprüfung wird nicht starr anhand von festen Angriffsmustern, die in einer Bibliothek abgespeichert sind, abgearbeitet. Vielmehr ist der Scanner in der Lage, bestimmte Attacken zu überspringen, wenn technisch ähnliche Scans im Vorfeld gezeigt haben, dass diese Attacken nicht zu dem Ziel führten. Diese Neuerung kann in der Praxis Auswirkungen auf die Dauer eines Vulnerabilty Scans haben. Bei komplexen Webseiten können Vulnerability Scans oft sehr lange dauern. So zeigt das Beispiel eines Kunde der Firma SPI Dynamics, der über eine Webanwendung mit mehreren Millionen Seiten verfügt, dass ein Test unter herkömmlicher Durchführung bis zu 75 Tagen dauern kann. Das intelligente Auslassen von nicht benötigten Überprüfungen würde diese Zeit wesentlich reduzieren, laut SPI Dynamics auf wenige Stunden.

Der Einsatz dieser neuen, von SPI Dynamics "Intelligent Engine" getauften Technik, ermöglicht des Weiteren, dass die Zahl der zu Unrecht ermittelten Fehler reduziert werden konnte. Neben der Geschwindigkeit eines Scans konnte somit auch die Aussagegüte verbessert werden. WebInspekt bietet ferner über das "Smart Update-Interface" die Möglichkeit eine Liste von aktuellen Sicherheitslücken von der Seite des Herstellers herunter zu laden. Hierdurch kann das Tool immer auf dem aktuellen Stand der Technik gehalten werden. Laut Herstellerangaben kann das Tool über eine API vom Benutzer mit eigenen Kriterien erweitert werden. Ein Nachteil des WebInspekt-Tools ist, das der Client nur für Windows Plattformen verfügbar ist. Linux oder MacOX Nutzer können folglich das Tool nicht benutzen [SM].

Umfang der Prüfkriterien [WD]:

- Parameter Injection
- Cross-Site-Scripting
- Parameter Overflow
- Path Manipulation
- MS-DOS 8.3 Short Filename
- Application Mapping
- SSL Support
- State Management
- HTTP Compliance
- Command Execution
- Directory Traversal
- Buffer Overflow
- Path Truncation
- Character Stripping
- Crawl
- Proxy Support
- Directory Enumeration
- WebDAV Compliance
- SQL Injection
- Abnormal Input
- Parameter Addition
- Character Encoding
- Site Search
- Automatic Form-Filling
- Client Certificate Support
- Web Server Assessment
- SSL Strength

- Certificate Analysis
- Client-Side Pricing
- Absolute Path Detection
- Known Attacks
- Assessment
- Content Investigation
- Brute Force Authentication attacks
- Sensitive Developer Comments
- WebServer/Web Package Identification
- Error Message Identification Permissions
- Spam Gateway Detection

Zusammenfassung:

SPI Dynamics bietet mit ihrem WebInspect-Tool einen leistungsfähigen Web-Application Scanner an. Das besonderes Merkmal des Tools ist sein Umfang an Prüfkriterien, seine leistungsfähige Report-Schnittstelle und die Geschwindigkeit einer Überprüfung. Preislich ist er im oberen Feld angesiedelt und zielt vornehmlich auf große Firmen.

WebInspect Lizenzen [WIC]:	
Scan eines Netzwerkes + Systempflege für ein Jahr	25.000 $ (19083 €) + 5.000 $ (3816 €)
Scan eines Servers	6.000 $ (4580 €)
Scan einer einzelnen IP-Adresse für 30 Tage	2.500 $ (1908 €)
Zweimaliger Scan eines IP-Bereiches innerhalb eines Jahres + für jedes weitere Jahr	20.000 $ (15267 €) + 5.000 $ (3816 €)

Tabelle 11: WebInspect Preistabelle

5.1.5 eEye (Retina)

"Retina Network Security Scanner" ist ein kommerzieller Web-Application Scanner, der in seinem Funktionsumfang am ehesten dem Programm Nessus entspricht. Von Retina werden nicht nur Fehler in Web-Application entdeckt, sondern prinzipiell in jeder Anwendung. Retina kann nur unter Windows betrieben werden und ist in der Benutzung sehr einfach gehalten. Der Hersteller eEye stellt seinen Kunden eine 30-Tage Probeversion zur Verfügung, so dass das Tool vor dem Kauf ausgiebig getestet werden kann. Nach der Testperiode kostet das Programm zum Beispiel für das Scannen von 16 IP-Adressen 760 €. Durch den Einsatz des Firmeneigenen CHAM (Common Hacking Attack Methods) erkennt Retina nicht nur bekannte Sicherheitslücken, sondern arbeitet sich auf Wunsch wie ein Hacker durch das zu testende Netzwerk und sucht nach unbekannten Schwachstellen, Passwörtern und Angriffspunkten. Diese Methode entspricht weitestgehend der "Intelligent Engine" getauften Technik der Firma SPI Dynamics. Darüber hinaus gibt das Programm, falls erwünscht, dem Benutzer genaue Anweisungen, wie die aufgedeckten Schwachstellen gelöst werden können. So können zum Beispiel laut Aussage des Hersteller falsche Konfigurationen auf einem Server teilweise mittels weniger Mausklicks korrigiert werden [SECO].

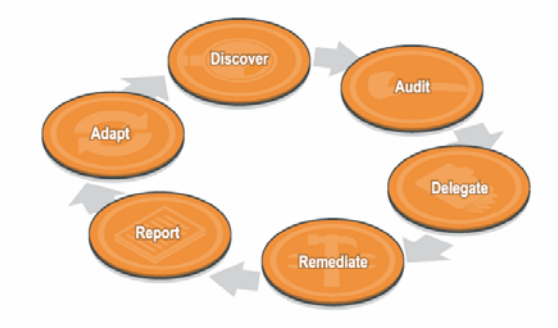

Abbildung 30: Retina Sicherheits-Audit Zyklus [EYE]

Durchführung eines Sicherheits-Audit [EYE]:

Insgesamt fällt der Zyklus einer Sicherheitsüberprüfung der Firma Retina etwas ausführlicher aus. In seiner Architektur orientiert er sich nichtsdestotrotz an den üblichen Vorgehensschritten eines Web-Application Scanners.

1. Discover (erforschen):
Im ersten Schritt wird eine genaue und für den Unternehmensablauf unterbrechungsfreie Untersuchung des angegebenen Netzwerkes durchgeführt. Hierbei werden alle schützenswerten Firmenbausteine identifiziert und katalogisiert.

2. Audit (analysieren):
Der zweite Schritt besteht in einem umfassenden Scan, durch den die Gegenwart bestehender Schwachstellen aufgedeckt werden soll.

3. Delegate (delegieren):
In diesem Schritt werden den aufgedeckten Schwachstellen sowohl zuständige Personen im Unternehmen, als auch Prioritäten bei der Behebung zugeordnet. Diese Strukturierung fördert die Effektivität bei der Verwaltung und Abarbeitung von Schwachstellen.

4. Remediate (beseitigen):
Von Software- und Konfigurationsfehler verursachte Schwachstellen können automatisiert korrigiert werden. Betrifft dies Fehlerstellen die unternehmensweit vorkommen, können diese im Folgeschritt überall geschlossen werden.

5. Reports (berichten):
Umfassende Berichte verfolgen und ergänzen den Prozess der Schließung der Schwachstellen und stellen eine klare Firmenpolitik bei ähnlichen Schwachstellen in der Zukunft sicher.

6. Adapt (anpassen):
Um zukünftige Schwachstellen im IT-System zu vermeiden, wird gezielt nach gefährdeten Bereichen gesucht und eine Empfehlung zur Handhabung bzw. der Anpassung des Systems gegeben.

Zusammenfassung:

Retina richtet sich an Firmen von mittlerer bis großer Größe. Das Tool zeichnet sich insbesondere dadurch aus, dass es eine "Rundumpflege" eines zu testenden Systems bietet. Es ist kein reiner Vulnerability-Scanner, der nur Schwachstellen aufdeckt. Vielmehr bietet es zusätzlich eine strukturierte Vorgehensweise zur Schließung von Angriffspunkten. Aufgrund dieses Funktionsumfangs ist es preislich im oberen Mittelfeld angesiedelt. Weiterhin erfordert es durch den Umfang sowohl vom Toolanbieter, als auch vom Auftragnehmer einen hohen individuellen Arbeitsaufwand.

Retina Lizenz Preise [HA]	
Scannen von 16 IP-Adressen	760 €
128 Asset Pack (Lizenzlaufzeit: 1 Jahr - inkl. Updates und Support)	1.799 €
256 Asset Pack (Lizenzlaufzeit: 1 Jahr - inkl. Updates und Support)	3.577 €

Tabelle 12: Retina Preistabelle

5.1.6 GFI LANguard (N.S.S.)

Die Firma GFI LANguard vertreibt mit dem Network Security Scanner (N.S.S.) einen weiteren Web-Application Scanner. Das Tool setzt sich aus drei Komponenten zusammen: Die erste Komponente stellt der eigentliche Vulnerability-Scanner. Die zweite Komponente besteht aus einem Modul, dass den Benutzer bei der Installation von Softwareupdates unterstützt. Die dritte Komponente ist ein Report Tool, dass einen ausführlichen Bericht über das zu untersuchende Netzwerk und die verwendete Software liefert. Der Network Security Scanner zeichnet sich im Vergleich mit den Lösungen anderer Herstellern durch die Eigenschaft aus, dass er durch zusätzliche Module erweitert werden kann. Diese können von einem Kunden, abgestimmt auf seinen individuellen Bedarf, zugekauft werden. Wünscht zum Beispiel ein Kunde grafisch aufbereitete Berichte (Reports) der durchgeführten Analysen, kann er dies durch den Zukauf des Zusatzmodul mit der Bezeichnung "Report Pack" erreichen. Eine automatisierte und netzwerkweite Analyse von Sicherheitsvorfällen wird durch den Erwerb des "GFI EventsManager" unterstützt. Das Modul gestattet einem Benutzer die Erstellung von Ablaufplänen die anschließend automatisiert durchgeführt werden (Scheduling). Die verschiedenen Module müssen bei einem Kostenvergleich mit den Produkten anderer Anbieter berücksichtigt werden, da sie

den gesamt Preis erhöhen. So kostet der Zukauf von EventsManager-Lizenzen 725€ für drei Netzwerkknoten und endet bei 28.800€ für 500 Knoten, die zusätzlich zum N.S.S-Tool investiert werden müssen.

Abbildung 31: GFI LANguard N.S.S. [GFI]

Aufbau des Network Security Scanner (N.S.S.) [GFI]:

1. Das erste Modul besteht aus einem Analyse-Tool, dass ein Netzwerk und aller darauf laufenden Anwendungen überprüft (Netzwerk- und Software-Auditing). Dabei wird der Schwerpunkt auf die folgenden Kriterien gesetzt:

- Automatische Warnmeldungen bei neu entdeckten Sicherheitslücken.
- Speicherung unterschiedlicher Zugangsdaten für Netzwerkrechner, damit auf diese bei einer Überprüfung automatisch zugegriffen werden kann .
- Überprüfung auf netzwerkweite Aktivierung des Sicherheits-Auditing.
- Scanning und Abruf von Linux-Systeminformationen.

2. Das zweite Modul besteht aus einem Schwachstellenscanner, der über eine erweiterte Funktionalität im Hinblick auf das Netzwerk- und Software-Auditing verfügt. Die aufgedeckten Schwachstellen werden katalogisiert und wenn möglich werden Gegenmaßnahmen für die Schließung angeboten. Die Analyse kann individuell an die persönlichen Bedürfnisse angepasst werden. So wird zum Beispiel das Untersuchen von Benutzerkonten oder USB-Geräten unterstützt. Des weiterten kann mit der Hilfe eines Assistenten eigene Kriterien in das Tool aufgenommen werden.

3. Das dritte Modul bietet die Möglichkeit, dass wenn Patches und Service Packs für Betriebssysteme und Anwendungen fehlen, diese automatisch im gesamten Netzwerk verteilt und installiert werden. Gleiches kann auch mit Drittanbieter-Software oder -Patches durchgeführt werden.

Zusammenfassung:

Der GFI LANGuard Network Security Scanner hebt sich durch seine modulare Bauweiße von den anderen Mitbewerbern ab. Des weiteren ist er in seiner Grundausstattung mit 525 € ein vergleichsweiße günstiger Vulnerability Scanner. Werden aber zusätzliche Funktionen benötigt schließt er an die teuren Scanner im Feld auf. In seinem Funktionsumfang geht er über den eines reinen Vulnerability Scanner hinaus, da er in gewissem Umfang eine Nachsorge des Benutzers bei der Schließung der Schwachstellen übernimmt (Patch-Management). Erwähnenswert ist noch, dass im Rahmen einer Diplomarbeit an der Uni Hamburg nachgewiesen wurde, dass ein durchgeführter Scan keinerlei erkennbaren Einträge in den Logdateien eines Webservers hinterließ. Dabei wurden sowohl die Ereignis-, Anwendungs- und Sicherheitsprotokolle, als auch die Log-Dateien des IIS-Webservers kontrolliert. Ein durchgeführter Scan auf Schwachstellen hinterlässt folglich auf dem getesteten System keinerlei bleibende Spuren[HH].

N.S.S [GFI_€]	
32 IPs (inklusive 3 Monate Update-Schutz)	525 €
64 IPs (inklusive 3 Monate Update-Schutz)	575 €
128 IPs (inklusive 3 Monate Update-Schutz)	755 €
256 IPs (inklusive 3 Monate Update-Schutz)	1.475 €
512 IPs (inklusive 3 Monate Update-Schutz)	2.650 €
Mehr als 512 IPs	Auf Anfrage
Consultant-Lizenz	Auf Anfrage
EventsManager	
3 Knoten	725 €
5 Knoten	1.025 €
100 Knoten	10.025 €
500 Knoten	28.800 €
N.S.S ReportPack [GFI_R]	
bis 512 IPs	450 €
Über 512 IPs	2.250€

Tabelle 13: GFI LANGuard Preistabelle

5.1.7 Watchfire (AppScan)

Die Firma Watchfire bietet mit dem Application Scanner AppScan ebenfalls ein Vulnerability Scanner an. AppScan ermöglicht einer Firma sowohl im Vorfeld während der Designphase, als auch in der anschließenden Betriebsphase ihre Webanwendungen auf sicherheitsrelevante Fehler hin zu überprüfen. Das Tool richtet sich folglich an die Entwickler von Webanwendungen, aber auch an Personen, die für die Absicherung einer bestehenden Webpräsenz zuständig sind. AppScan besteht aus mehreren Modulen. Die "Developer Edition" (AppScan DE) unterstützt den Entwickler bei der Konzeption und der

Implementierung sicherer Webanwendungen. Das Modul kann in Programmierumgebungen wie JBuilder, Websphere, MS Visual Studio und Eclipse integriert werden. Somit ermöglicht es bereits sehr früh eine Qualitätssicherung im Entwicklungsprozess [WF]. AppScan kann des weiteren dabei helfen typischen Sicherheitsprobleme von Webanwendungen mit Hilfe von automatisierten Tools zu erkennen und zu beheben. Appscan enthält zu diesem Zwecke drei grundlegende Funktionsbereiche: den eigentlichen Scanner, ein Reporting Tool und schließlich Werkzeuge, um bei Bedarf die Ergebnisse einer Untersuchung individuell auszukommentieren. Damit der Scanner nach einer Installation auf den aktuellen Softwarestand gebracht werden kann, d.h. insbesondere um die Schwachstellen-Datenbank zu aktualisieren, wird ein Update über das Internet durchgeführt. Der Benutzer wird bei der Konfiguration des Scanners von dem "Configuration Wizards" unterstützt. Von dem Wizard werden alle relevanten Informationen abgefragt, wie zum Beispiel die URL der zu scannenden Webanwendung oder Authentifizierungs-informationen, sofern diese benötigt werden um Beispielsweiße auf Server zugreifen zu können [LL].

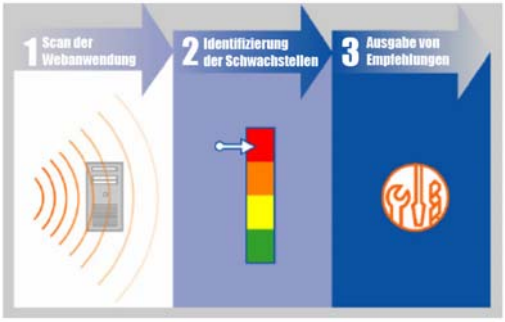

Abbildung 32: AppScan Audit [WFS]

Das erste Modul besteht aus einen Schwachstellen Scanner. Der Scanner ist laut Angabe des Herstellers in der Lage alle Schwachstellen der OWASP Top Ten und eine Reihe weiterer Schwachstellen aufzudecken, dazu gehören[WFS]:

- Cross-Site-Scripting
- Session Fixation
- Backdoors/Debug Options
- Application Buffer Overflow
- Misconfiguration
- Hidden Field Manipulation
- Content Spoofing
- HTTP Compliance
- Parameter Tampering
- Stealth Commanding
- Cookie Poisoning
- Known Vulnerabilities
- Suspicious Content
- LDAP Injection
- Response Splitting
- SQL Injections
- Forceful Browsing;
- Third-Party
- HTTP Attacks
- XML/SOAP Tests
- XPath Injection

Nach dem Start eines Scanvorgangs "surft" das Tool zunächst durch die Webanwendung, mit dem Ziel den Aufbau zu ermitteln. Hierbei füllt das System auch gefundene Formulare mit Testdaten aus. Dieser Vorgang (Crawling) soll alle Punkte finden, an dem Benutzer-informationen und -Eingaben an die Webanwendung weitergereicht werden. Der Grund

hiefür liegt darin, dass dies die typischen Angriffspunkte für SQL-Injektion, Cross-Site-Scripting und andere Attacken sind.
Das zweite Modul besteht aus einem Reporting-Tool. Es können, je nach Bedarf, verschiedene Arten von Berichten generiert werden. Möglich sind zum Beispiel Berichte für das Management (Executive Summary), Detailberichte inklusive Sourcecode und Beispielen für die Entwickler oder auch Compliance-Berichte zu verschiedenen Standards.

In dem dritten Modul werden dem Benutzer Lösungsmöglichkeiten angeboten, wie er die gefundenen Schwachstellen schließen kann. Das können Empfehlungen in dem Hinblick sein, Updates für Softwarekomponenten zu installieren oder Konfigurationen von Hard- und Softwarekomponenten anzupassen.

Zusammenfassung:

Die Vorteile von AppScan liegen in der Aktualität der verwendeten Schwachstellenliste. Hervorzuheben ist seine Unterstützung von Softwareentwicklern schon während des Designprozesses ihre Anwendungen auf Schwachstellen hin zu unterschen. Kein anderer Application Scanner bietet dies. Nachteilig wirken sich aus, dass im Vergleich zu den Konkurrenzprodukten, hohe Lizenkosten (> 15.000 $) anfallen, dass keine selbst definierten Testskripts unterstützt werden können und das der Client des Scanners nur für die Windows Plattformen verfügbar ist.

AppScan [BW_€]	
Einzelplatz-Lizenz	15.000 $ (11.233 €)
Jahres-Lizenz	20.000 $ (14.976 €)

Tabelle 14: AppScan Preistabelle

5.1.8 N-Stalker (N-Stealth)

N-Stealth ist ein kommerzielles Tool der Firma N-Stalker, dass nach bekannten Sicherheitslücken sucht. Das Programm ist dahingehend optimiert Standard Schwachstellen in verbreiteten Webanwendungen und Webservern zu finden. Die Untersuchung eines Systems wird immer nach einem festen Muster durchgeführt.

Zu Beginn einer Untersuchung versucht der Scanner anhand eines "HTTP-HEAD-Request" den "http-Banner" eines zu untersuchenden Systems auszulesen. Der http-Banner ist eine kurzen Nachricht, die von dem http-Protokoll versendet wird. Die Nachricht wird immer versendet, wenn eine Clientanwendung eine Verbindung zu einer Serversoftware herstellt. Das ist zum Beispiel der Fall, wenn ein Webbrowser eine Webseite von einem Webserver abfragt.

Ein typischer http-Banner sieht wie folgt aus[IEYE]:

```
HTTP/1.1 200 OK
Date: Thu, 17 Jul 2003 10:59:09 GMT
Server: Apache/1.3.27 (Unix) (Red-Hat/Linux) mod_gzip/1.3.19.1a mod_jk/1.2.0
mod_perl/1.26 PHP/4.2.2 FrontPage/5.0.2 mod_ssl/2.8.12 OpenSSL/0.9.6b
Connection: close
Content-Type: text/html
```

Codebeispiel 1: Http-Banner

Aus dieser kurzen Nachricht kann entnommen werden, dass ein Apache Webserver (Version 1.3.27) mit einem darauf laufendem Red-Hat Linux verwendet wird. Außerdem können weitere Module erkannt werden, die an den Apache Webserver angebunden sind (z.B. OpenSSL/0.9.6b, PHP/4.2.2,...). Konnte der http-Banner erfolgreich ausgelesen werden, wird der eigentliche Scanvorgang gestartet. Nach Angaben der Vertreiberfirma werden von N-Stealth bei einem Scan 30.000 vordefinierte Prüfkriterien durchlaufen. N-Stealth läuft nur auf der Windows-Plattform und zeichnet sich, laut eines von der Firma SEC Consult durchgeführten Produktvergleiches unter Vulnerability Scannern, im wesentlichen durch eine strukturierte Bedienung aus. Verglichen wurde N-Stealth in dem durchgeführten Test mit dem Open Source Scanner Nessus und dem Vulnerability Scanner von eEye. [SECO].

N-Stealth wird in drei Versionen angeboten. Diese unterschieden sich in ihrem Funktionsumfang und in ihrer Zielgruppe[NST]:

1. N-Stalker QA-Edition:

Diese Version richtet sich insbesondere an die Bereiche einer Firma, die für die Entwicklung- und die Qualitätssicherung von Softwareprodukten zuständig sind.

2. N-Stalker Infrastructure-Edition:

Die Infrastructure Edition richtet sich insbesondere an Netwerkverwalter und mittelständische Unternehmen die ihre IT-Infrastruktur auf Schwachstellen hin überprüfen wollen.

3. N-Stalker Enterprise Edition:

Die Enterprise Edition setzt sich aus der QA-Edition und der Infrastructure-Edition zusammen. Zusätzlich wird in das Produkt ein Audit- und ein Penetrations-Tool eingegliedert. Mit der Enterprise Edition kann der gesamte Softwarelebenslauf auf Schwachstellen überwacht werden. Sie richtet sich insbesondere an große Unternehmen die eine eigene IT-Abteilung besitzen.

Zusammenfassung:

Der von N-Stalker gebotene Komfort wird durch eine relativ hohe Preisgestaltung an den Kunden weiter gegeben. Aufgrund des über Module erweiterbaren Scanners bietet die Firma ihren Kunden die Möglichkeit Zielgerichtet das Produkt zu kaufen, dass für ihre Ansprüche am besten geeignet ist. Eine Erweiterbarkeit mit eigenen Kriterien ist nicht vorgesehen [HH].

N-Stalker Preisliste [NST_€]			
	1 IP (1 Jahr)	8 IPs (1 Jahr)	Consultant
Enterprise Edition	1.399 $ (1.042 €)	2.899 $ (2.160 €)	6.199 $ (4.620 €)
Infrastructure Edition	299 $ (222 €)	499 $ (371 €)	1.599 $ (1.191 €)
QA Edition	1.098 $ (818 €)	2.399 $ (1.788 €)	4.599 $ (3.427 €)

Tabelle 15: N-Stalker Preistabelle

5.1.9 Microsoft (MBSA)

Der Microsoft Baseline Security Analyzer (MBSA) 2.0 ist ein einfach zu nutzendes Tool für kleine und mittlere Unternehmen. Auf Basis der regelmäßig aktualisierten Sicherheitsempfehlungen der Firma Microsoft können die Unternehmen ihren Sicherheitsstatus überprüfen. Primär werden bei der Überprüfung oft auftretende Schwachstellen aus Fehlern bei der Konfigurationen und fehlende Sicherheitsupdates ermittelt. Der MBSA basiert auf einer Kombination des Windows Update-Agent und der Microsoft Update-Infrastruktur. Er bietet auf diesem Wege eine Brückenfunktion zu anderen Microsoft-Produkten wie dem Microsoft-Update (MU), dem Windows-Server-Update-Services (WSUS), dem Systems Management Server (SMS) und dem Microsoft-Operations-Manager (MOM) sicher. Das Tool kann kostenlos von der Microsoft Homepage heruntergeladen werden [SIN].

Durchführung eines Sicherheits-Audit:

Nach der Installation fordert das Programm den Benutzer auf, auszuwählen ob ein einzelner Rechner oder mehrere Rechner in einem Netzwerk überprüft werden sollen. Der zu überprüfende Rechner wird mittels seiner IP-Adresse oder seinem Rechnernamen angegeben. Sollen mehrere Rechner überprüft werden, so werden diese über ihren Adressebereich festgelegt. Voraussetzung für den Zugriff auf diese Rechner ist, dass der für den Scan verwendete Rechner, beziehungsweiße der angemeldete Benutzer, administrative Rechte auf den zu untersuchenden Rechnern besitzt. In dem nachfolgenden Schritt werden

die angegebenen Rechner auf Schwachstellen hin untersucht. Bei einem Testdurchlauf an der Universität Koblenz benötigte das Tool für eine Überprüfung von 256 IP-Adressen knappe 30 Minuten. Im Anschluss an diesen Test wird ein kurzer Bericht mit aufgedeckten Schwachstellen oder nicht installierten Patches ausgegeben.

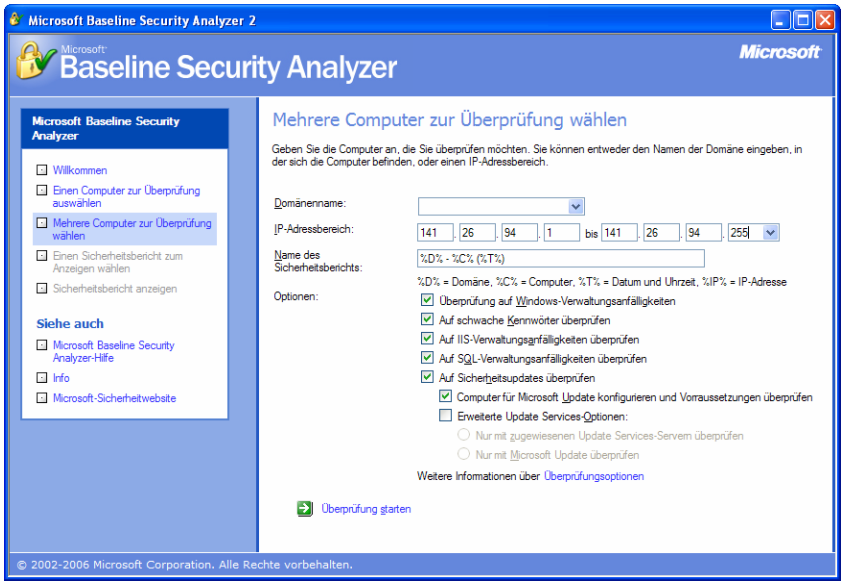

Abbildung 33: Microsoft Baseline Security Analyzer – Screenshot

Eine Überprüfung bezieht sich dabei auf folgende Bereiche:

• Windows XP, NT, 2k; Webserver IIS in Version 5.0 und 6.0; SQL Server 7.0 und 2000; Internet Explorer ab Version 5.01; Office 2000, 2002 and 2003 *(Anwendungen).*
• Treiber und BIOS-Versionen *(Hardware).*
• Aktualität der Patches und Updates *(Betriebssystem).*
• Einstellungen bezüglich Kennwortsicherheit und Benutzerkonten *(Konten).*
• Exchange, SQL, und andere Microsoft-Anwendungen *(Serveranwendungen).*

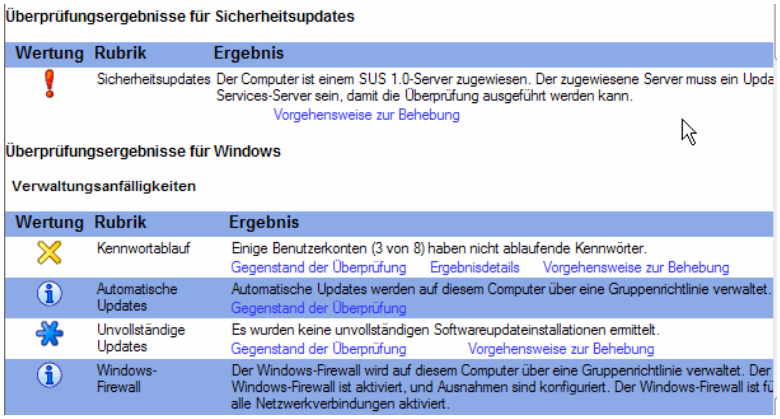

Abbildung 34: Microsoft Baseline Security Analyzer – Prüfbericht

Zusammenfassung:

Der Microsoft Baseline Security Analyzer muss zu den vorher vorgestellten Web-Application Scanner außer Konkurrenz betrachtet werden. Seine Prüfkriterien behandeln nur Microsoft Produkte. Eine Überprüfung einer Webanwendung ist folglich nur soweit möglich, wie diese auf Microsoft Produkten aufbaut. So ist zum Beispiel die Überprüfung einer verwendeten Datenbank (MS-SQL-Server) oder eines Webservers (MS-IIS) denkbar. Eine Überprüfung der Webanwendung auf Cross-Site-Scripting oder anderer auf Script-Sprachen basierenden Schwachstellen kann durch den MBSA nicht durchgeführt werden. Der Funktionsumfang den er bietet, erfüllt er aber zuverlässig und komfortabel. Ferner ist sein Einsatz ohne weitere Kosten verbunden und kann von einem Administrator benutzt werden, um die Background-Architektur eines Webauftrittes abzusichern. Somit ergibt sich eine Empfehlung in der Hinsicht, dass der MBSA parallel zu einem Web-Application Scanner eingesetzt werden sollte.

6. Praxisszenarien

Der Verlauf der Diplomarbeit begann mit einer Analyse eines E-Commerce-Webauftrittes. Das Ziel dieses ersten Schrittes war es, sowohl den grundsätzlichen Aufbau als auch die konkret verwendeten Bausteine vorzustellen. Anschließend konnten durch die Untersuchung der Bausteine mögliche Schwachstellen für Angriffe aufgedeckt werden. In den darauf folgenden Schritten wurden Prüfkriterien ausgearbeitet, die die zuvor vorgestellten Schwachstellen abdecken. Das zuletzt behandelte Kapitel bot einen Ausblick über Möglichkeiten, wie diese Kriterien automatisiert überprüft werden können. Während die ersten fünf Kapitel grundsätzlich die theoretischen Grundlagen behandelten, werden in diesem Kapitel nun anhand von praktischen Beispielen mögliche Attacken und deren Auswirkungen auf Webpräsenzen dargelegt. Der Focus liegt hierbei auf den ersten sechs Schwachstellen, die in der OWASP Top Ten aufgelistet werden. Die Praxisszenarien dienen zur Veranschaulichung und zeigen als Abschluss dieser Arbeit die individuellen Gefährdungen, die von den häufigsten Angriffsformen ausgehen. Weiterhin kann anhand der Szenarien gezeigt werden, dass Handlungsbedarf besteht, um für eine technische Absicherung eines E-Commerce-Webauftrittes zu sorgen. Gerade dieser Handlungsbedarf bietet eine Basis dafür, Kunden von Trusted Shops die möglichen Risiken ihrer individuellen Webanwendungen an Beispielen zu zeigen und somit den Umfang einer Überprüfung abzustimmen.

6.1 XSS-Attacken

Cross-Site-Scripting-Attacken (XSS) basieren auf Scriptsprachen wie JavaScript, CGI und PHP. Es sind zwei wesentliche Gefährdungsszenarien denkbar. Das erste Szenario besteht in dem Diebstahl von Cookiedateien von Webseitenbesuchern *(Session-Hacking)*. Das zweite Szenario liegt in einer möglichen Verfremdung von Webseiten *(Defacement)*. Aufgedeckte XSS-Schwachstellen von bekannten Webpräsenzen können auf der Webseite "XXS-Archiv" [XSS_A] nachgelesen werden. Für einen E-Commerce-Anbieter sind beide Szenarien gefährlich. Lässt der Anbieter durch technische Schwachstellen zu, dass einem Kunden die Identität mittels Session-Hacking gestohlen wird, sind Schäden am Image der Firma und indirekt auch ein wirtschaftlicher Schäden zu befürchten. Ähnliches gilt für Defacement Attacken, hier muss der Anbieter Verunglimpfungen auf seinen Webseiten oder finanziell schädigende Veränderungen von Preisen oder Produkten befürchten. Die beiden geschilderten Attacken werden oft über in E-Mails eingebettete und präparierte Links verbreitet. Nach eine aktuellen Studie der "Arbeitsgruppe Identitätsschutz im Internet"

haben diese Attacken eine Erfolgsrate von 14%. Wenn der geringe Aufwand für das oft massenhaft angewendete Versenden von E-Mails beachtet wird, kann diese zuerst gering erscheinende Erfolgsrate, schnell zu einer großen Zahl von Opfern führen [AI].

6.1.1 Session Hacking / Identitätsdiebstahl

Cookies von E-Commerce-Webseiten können schützenswerte Informationen wie Sitzungsnummern (Session-ID) und zusätzliche Authentifizierungsdaten enthalten. Von den Webservern des Webshops werden Cookies auf dem PC des Kunden abgelegt und von dort bei Bedarf wieder abgefragt. Beispiele für E-Commerce-Webseiten die Cookies zur Authentifizierung verwenden sind Amazon und eBay. Mit Cross-Site-Scripting ist ein Angreifer in der Lage, Cookies abzufangen, einzusehen, zu manipulieren oder zu löschen. Dadurch kann er die Session eines E-Commerce-Kunden stehlen [BD_A].

Session Hacking im Verbund mit Apache Struts

Neben dem erwähnten präparierten Link sind keine Informationen veröffentlicht worden, wie der eigentliche Diebstahl eines Cookies durchgeführt wurde. Dahingehend können nur die Ursachen und die Auswirkungen für einen Benutzer dargelegt werden.

```
<script>location.href="http://evil.org/log.jsp?ID=" +
document.cookie;</script>
```

Codebeispiel 2: Link für einen Cookiediebstahl

Der präparierte Link wird im Vorfeld einem Benutzer untergeschoben. Klickt der Benutzer auf diesen Link, wird ein JavaScript (JavaServer Pages) gestartet, durch das die Cookiedatei entwendet wird. Ein Programmierfehler in dem Apache Struts Projekt ermöglicht dem Angreifer im Anschluss an den Diebstahl die Cookiedateien einzusetzen. Apache Struts ist ein Open-Source-Framework für die Präsentationsschicht von Java-Webanwendungen.

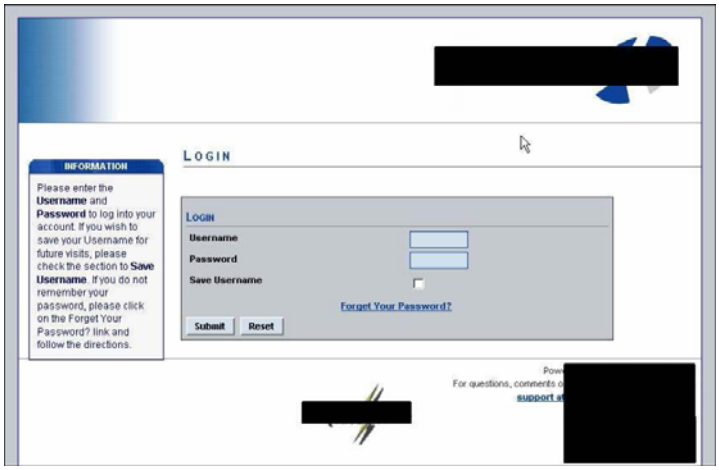

Abbildung 35: XSS - Session Hacking [ATA]

Die Abbildung 35 zeigt eine beliebige Webanwendung, die Apache Struts einsetzt. Sind die Benutzer dieser Webanwendung angriffen und ihrem Cookie beraubt worden, kann der Angreifer durch eine geschickte Farbwahl Fehlermeldungen für den Benutzer ausblenden. So ist im obigen Beispiel tatsächlich eine Fehlermeldung in der dargestellten Webseite enthalten. Diese Fehlermeldung soll den Benutzer dahingehend warnen, dass ein zweiter Benutzer (der Angreifer) mit seinem Account (Session) online ist. Da diese Fehlermeldung aber von dem Angreifer auf einen weißen Farbwert eingestellt wird, kann sie von dem Benutzer nicht wahrgenommen werden. Folglich kann der Benutzer den Diebstahl seiner Session weder erkennen, noch kann er darauf reagieren [ATA].

PayPal

Im Juni 2006 wurde die Online-Bezahlplattform PayPal das Opfer einer Phising-Attacke. Die Phising-Attacke wurde mittels Cross-Site-Scripting durchgeführt. Ursache war eine Schwachstelle in einer Anwendung der Webseite. Die Anwendung konnte von den unbekannten Angreifern dahingehend verfremdet werden, dass Benutzer der Bezahlplattform dazu verleitet wurden gutgläubig Daten wie Kreditkartenummer und andere persönliche Daten preiszugeben. Der Angriff wurde mittels einer E-Mail durchgeführt, die an PayPal Benutzer verschickt wurde. Diese E-Mail beinhaltete die Aufforderung eine URL der PayPal-Webseite aufzurufen. Die aufgerufene PayPal-Webseite war mittels Cross-Site-Scripting verändert worden und leitete den Benutzer auf eine Webseite aus Südkorea weiter. Dort wurde dann um die Eingabe des PayPal-Benutzernames und des Passworts gebeten. Im angeblich offiziellen "Resolution Center" erfolgte anschließend die Abfrage vertraulicher Kreditkartendaten, PIN-Nummern von Bankkarten oder Sozialversicherungsnummern. Begünstigt wurde die Täuschung der Benutzer durch die Begebenheit, dass der Code für die böswillige Aktion per Cross-Site-Scripting in die offizielle Website des Unternehmens eingeschleust worden war und die Abfrage somit vertrauenswürdig zu sein schien [PP].

Banking-Portale

Die Banking-Portale verschiedener deutscher Geldinstitute wiesen im März 2007 ernsthafte Sicherheitsmängel durch sogenannte Cross-Site-Scripting-Schwachstellen auf. Die Schwachstellen lagen nach Aussage der Zeitschrift Ct in den Suchfunktionen der Webseiten, die Benutzereingaben nicht ausreichend filterten. Bei einem Cross-Site-Scripting-Angriff lockt ein Angreifer seine Opfer mit Hilfe manipulierter Links in E-Mails oder Webseiten auf die anfällige Website. Durch die Schwachstelle ist es dem Angreifer möglich das Erscheinungsbild der Bankseite lokal im Browser des Opfer zu verändern. Da die verfremdete Webseite über eine reguläre Webadresse der Bank geöffnet wird, ist es für das Opfer nicht direkt zu erkennen, dass es sich nicht um eine original Webseite der jeweiligen Bank handelt. So kann der Angreifer unter Umständen etwa an Cookiedateien oder andere

vertrauliche Daten der Opfer gelangen. Zu diesem Zweck kann der Angreifer mit seinen Skripten beispielsweise Eingabemasken für PINs und TANs nachahmt. Für Bankkunden könnte leicht der Eindruck entstehen, es handle sich um eine vorgesehene Funktion des Banking-Portals [RC].

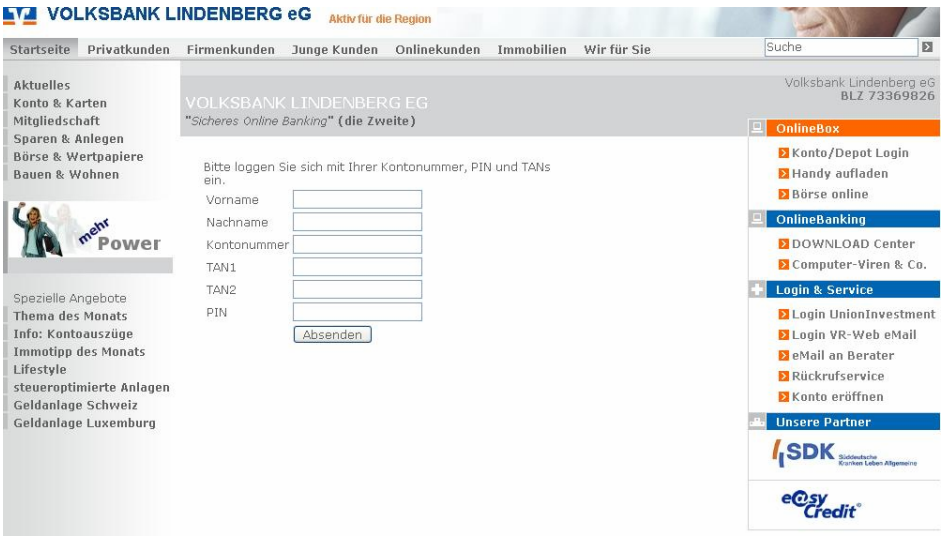

Abbildung 36: XSS - Attacke auf ein Banking-Portal [RC]

6.1.2 Defacement

Defacement-Attacken sind daraufhin ausgerichtet, einen Benutzer dazu zu verleiten, Informationen preiszugeben. Hierfür wird das Aussehen der eigentlichen Webseite durch PHP-, CGI- oder JavaScripte verfremdet. Weiterhin wird Defacement zur inhaltlichen Verfremdung eingesetzt, um Webseitenbetreiber einen Imageverlust zuzufügen.

CBS-News

Im August 2006 wurde die Wetterseite von CBS-News das Opfer einer Cross-Site-Scripting-Attacke, ausgeübt von russischen Softwareentwicklern. Die Attacke bediente sich des "Zip or City" Eingabefeldes der Wetterseite, um einen eigenen schadhaften Code einzuschleusen. Durch diese Codeeinschleusung konnte die eigentlich korrekt funktionierende Webseiteb in ihrem Erscheinungsbild verfremdet werden (Defacement). Das Ziel der Attacke war es, einen fiktiven Bericht in einen vertrauenswürdigen Kontext einzuschleusen. Die verfremdete Seite gab an, dass George W. Bush einen neunjährigen Securityexperten zum neuen Vorsitzenden des "Information Security Deportment" gemacht habe. Inzwischen ist die betreffende Schwachstelle behoben worden und der verwendete Link hat keine Auswirkungen mehr auf die Webseite.

Abbildung 37: XSS – Defacement [ATA]

```
http://www.cbsnews.com/stories/2002/02/15/weather_local/main501644.shtml?
zipcode=1--%3E%3Cscript%20src=http://www.securitylab.ru/test/sc.js
%3E%3C/script%3E%3C!-
```

Codebeispiel 3: Präparierter Link für ein XSS-Defacement

Die Attacke wurde gestartet, indem ein Benutzer den präparierten Link anklickte. Dieser Link wurde von den russischen Hackern speziell für die CBS-Newsseite von Hand entwickelt und funktionierte nur bei dieser einen Seite. Klickte ein Benutzer den präparierten Link an, wurden die CBS-Newsseite und parallel dazu ein JavaScript File (`http://www.securitylab.ru/test/sc.js`) von dem Server der russischen Softwarefirma heruntergeladen. Das JavaScript File enthielt den verunglimpfenden Text und integrierte diesen bei der Ausführung in die betreffende News-Webseite. Verbreitet wurde der Link über "soziale Netzwerke" wie Instant Messenger und per E-Mail. Die Attacke kann sowohl als eine aggressive Werbeaktion der russischen Softwarefirma verstanden werden, als auch als verunglimpfende Attacke auf die CBS-Newsseite, bzw. die amerikanische Politik [ATA].

Netscape

Im Juli 2007 wurde das Nachrichten-Portal der Social Media-Webseite der Firma Netscape durch einen Angriff per Cross-Site-Scripting gehackt. Aufgedeckt wurde der Angriff von der finischen Firma F-Secure, die sich auf Sicherheitssoftware spezialisiert hat. Die Firma berichtet, dass die Verfremdung der Webseite auf die Anhänger des Wettbewerbers Digg.com zurückgeht. Über Cross-Site-Scripting-Schwachstelle konnte JavaScript-Code unter anderem in die Startseite der Webseite eingeschleust werden [FSN].

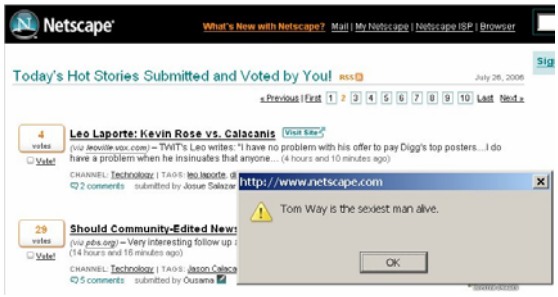

Abbildung 38: Netscape Defacement [FSN]

E-Plus Webshop

Im Januar 2007 konnte gezeigt werden, wie ein möglicher Cross-Site-Scipting-Angriff auf den Webshop des Mobilfunkanbieters E-Plus aussehen würde. Auf welche Weise das Defacement erreicht wurde, ist nicht dokumentiert. Es kann aber davon ausgegangen werden, dass die Suchfunktion der Webseite für den Angriff missbraucht wurde. Dies lässt sich aus dem Screenshot in der Abbildung 39 erkennen. In der linken Hälfte der Webseite ist das Feld für die Suchfunktion zu erkennen. Durch diese Attacke ist der Angreifer in der Lage das Aussehen der Webseite beliebig zu verändern [SJ_PHP].

Abbildung 39: E-Plus Webshop XSS-Defacement [SJ_PHP]

6.2 SQL-Attacken

Die im vorhergehenden Kapitel betrachteten Cross-Site-Scripting-Attacken (XSS) sind in vielen Fällen an die Benutzer von Webseiten gerichtet. SQL Injection Attacken werden hingegen von Angreifern mit dem Ziel eingesetzt, direkten Zugriff auf einen Datenbankserver zu erlangen. Infolgedessen ist der Datenbestand des Opfers einer SQL-Attacke gefährdet, da dieser von dem Angreifer nicht nur ausgelesen, sonder auch dauerhaft verändert oder gar gelöscht werden kann. Eine zweite Angriffsform besteht darin, dass in der Praxis SQL-Attacken (SQL-Injection) oft eingesetzt werden, um schadhafte Daten und Programme in fremde Server einzubetten (Code-Injection). Die höhere Gefahr geht hierbei nicht von dem eigentlich SQL Angriff, sondern von der anschließenden Code-Injection aus. Durch diese kann von einem Angreifer beliebiger Code auf den Datenbankservern des Angegriffen abgelegt und anschließend verbreitet werden. Besonders arglistig ist hierbei, das der abgelegte Code durch die benutze Webpräsenz vertrauenswürdig erscheint und von Benutzern nicht als schadhaft erkannt werden kann.

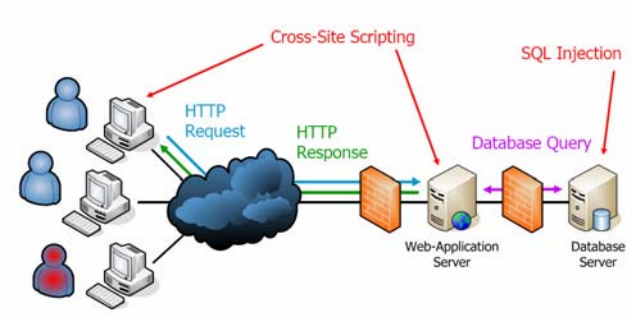

Abbildung 40: Cross-Site-Scripting ↔ SQL-Injection [RM]

SQL-Attacken nutzen Software- oder Konfigurationsfehler von Datenbankumgebungen aus. Die folgenden Praxisszenarien von SQL-Attacken ergeben sich deshalb aus der jeweils eingesetzten Softwarearchitektur.

Aktuelle SQL-Injection Szenarien:

1. **Ausspähen von Benutzerdaten**

Im Frühjahr 2007 wurde die populäre Studentenplattform StudiVZ das Opfer einer SQL-Attacke. Es konnte von einem unbekannten Angreifer, über eine SQL-Injection eine unbekannte Anzahl von Profilen samt persönlichen Daten wie E-Mail-Adressen, ausgelesen werden [STU]. Das StudiVZ ist zwar keine eigentliche E-Commerce-Webpräsenz, zeigt aber das Risikopotential das besteht, wenn eine Webpräsenz nur ungenügend Entwickelt wurde und Schwachstellen bestehen. Ein ähnlicher Angriff auf einen E-Commerce-Händler hätte weit aus schwerwiegendere Folgen gehabt. Dort hätten neben Adressdaten auch Bankdaten und andere sensitive Datensätze gestohlen werden können.

2. Ausspähen von Passwörtern

Durch eine Schwachstelle im Content-Management-System "PHP-Nuke" wurde es im Oktober 2006 einem Angreifer ermöglicht, mittels SQL-Injection Zugriff auf die zugrundeliegende Datenbank zu erlangen. In einem konkreten Fall ist das Auslesen von Passwörtern ermöglicht worden. Die Ursache für das Problem liegt in einer fehlerhaften Filterung von Eingebewerten. Somit können von einem Angreifer eigene SQL-Befehle eingeschleust und an die Datenbank übergeben werden [NUKE].

3. Verbreiten von Malware

Im April 2007 konnte von einem Angreifer mittels einer SQL-Attacke eine schadhafte Datei auf einem Webserver eines Spiele Forums (Team Speak) abgelegt werden. Der Angriff erfolgte über eine bekannte Lücke in der Forensoftware vBulletin 3.6.5. Hierdurch konnte in den Foren-Server eingedrungen werden. Dort hat der Angreifer eine infizierte Datei "patch.exe" abgelegt und anschließend per Mailverteiler alle registrierten Forum-Mitglieder dazu aufgefordert, den "Patch" herunterzuladen und zu installieren[SPE].

6.3 Schadhafte Dateiausführungen

In der OWASP Liste der zehn häufigsten Schwachstellen von Webanwendungen nehmen "malicious file execution", zu Deutsch "schadhafte Datei Ausführungen", den dritten Platz nach Cross-Site-Scripting und SQL-Injection ein. Der Unterschied zu Cross-Site-Scripting-Attacken besteht darin, dass von einem Angreifer der schadhafte Code nicht zwischen einem sogenanntem "guten" Code versteckt wird. Ferner wird der eingeschleuste Code nicht wie bei Cross-Site-Scripting einmalig interpretiert und ausgeführt. Es werden vielmehr direkte Schwachstellen von Anwendungen ausgenutzt und komplette Programme dauerhaft in das angegriffene System eingeschleust.

Abhängig von dem gebotenen Funktionsumfang bieten Webanwendungen von E-Commerce-Seiten einem Benutzer die Möglichkeit, eigene Dateien von seinem Rechner auf einen Webserver hochzuladen. Denkbar sind zum Beispiel E-Commerce-Webseiten die über ein Forensystem, eine Photoalbumfunktion oder eine sonstige Datei-Upload Schnittstelle verfügen. Realisiert werden diese Dienste über Anwendungen, die in PHP programmiert wurden. In der Vergangenheit sind des Öfteren Schwachstellen aufgrund von Programmierfehlern in PHP-Anwendungen aufgetreten [SJ_PHP].

Aktuelle Szenarien:

1. **Forensysteme (phpBB)**

Im Februar 2007 wurde bekannt, dass diverse auf phpBB basierende Forensysteme für "malicious file execution" anfällig sind. Die Schwachstelle betraf infolgedessen eine Vielzahl von Forensystemen, unter denen sich auch bekannte wie das EclipseBB Forum befanden. Ursache war die sogenannte *phpbb_root_path-Schwachstelle*, die von den Entwicklern der verschiedenen Forensysteme unverändert von phpBB übernommen wurde. Von Angreifern konnte somit auf den betroffenen Systemen ein beliebiger Schadcode nachgeladen und zur Ausführung gebracht werden [RC].

2. **File-Upload-Funktion (PHP 3.x und 4.x)**

Die bei vielen Webanwendungen verwendete File-Upload-Funktion der Skriptsprache PHP enthält diverse Schwachstellen, die zu Sicherheitsproblemen führen. Betroffen sind besonders ältere PHP Versionen, wie die Versionen 3.x und 4.x. Diese Schwachstelle betrifft folglich Webseiten, die über einen längeren Zeitraum nicht mehr gepflegt wurden. Unter "Pflege" kann in diesem Zusammenhang insbesondere das regelmäßige Updaten der auf den Webservern verwendeten PHP-Versionen verstanden werden.

6.4 Unsichere interne Objektreferenzen

"Insecure Direct Object Reference" Schwachstellen treten immer dann auf, wenn einem Angreifer die Möglichkeit geboten wird an interne Informationen, zum Beispiel Objektreferenzen, zu gelangen und diese für einen Angriff zu missbrauchen. In der OWASP Top Ten Schwachstellen Liste nehmen sie aktuell den vierten Platz ein. Von der "National Vulnerability Database" der Amerikanischen Regierung werden mehrere Praxisbeispiele genannt.

1. **Joonas Viljanen JV2 Folder Gallery** (17.01.2007)

JV2 Folder Gallery ist eine frei verfügbare PHP-Anwendung, die von Webseitenbetreiber benutzt werden kann um ihre Webpräsenz mit einer Fotoalbumfunktion auszustatten. Die Schwachstelle dieser Webanwendung beruht auf der Datei "Download.php". Die Voraussetzung für einen Angriffspunkt ergibt sich daraus, dass die Datei von jedem Besucher der betreffenden Webseite ausgelesen werden kann. Bei der Programmierung der Datei Download.php wurde nicht beachtet, dass interne Dateiverzweigungen als nicht lesbar dargestellt werden müssen. So kann ein Angreifer aus der Datei entnehmen wo und welche Dateien auf der Webserverstruktur abgelegt werden [NVD_JV2].

```
strcat(sendCommand,"download.php?file=config/gallerysetup.php \n\n");
```

Codebeispiel 4: Schwachstelle in der Datei download.php

2. **IntegraMOD Portal** (26.08.2006)

IntegraMOD ist eine veränderte Version der Foren-Anwendung phpBB. Der Ausdruck "IntegraMOD" bedeutet "integrated modifications" (integrierte Änderungen). Aus der Datei "includes/functions_portal.php" können von einem Angreifer Informationen für einen Angriff auf den verwendeten Webserver entnommen werden. Wenn die Funktion "Magic_quotes_gpc" deaktiviert wurde, ist es Angreifern erlaubt, beliebige Dateien über einen absoluten Pfadnamen aus dem "Phpbb_root_path Parameter“ auszulesen [NVD_MOD].

6.5 Cross Site Request Forgery (CSRF)

CSRF-Attacken haben besonders in den letzten Jahren an Bedeutung gewonnen. In der OWASP Top Ten Liste nehmen sie inzwischen den fünften Platz ein. Der Unterschied zu einfachen Cross-Site-Scripting-Attacken ist darin zu sehen, dass der Angreifer das Zielsystem nicht direkt angreift, sondern sich eines Opfers bedient. Dieses Opfer muss über einen berechtigten Zugang zu der angegriffenen Webanwendung besitzen. In den letzten Monaten sind zwei Schwachstellen bekannt geworden die besonders gut die Gefährdung von CSRF-Angriffen aufzeigen:

1. **MKPortal** (12.01.2007)

MKPortal ist ein kostenloses Portal- bzw. Content Management System (CMS). Anfang 2007 ist eine Schwachstelle in der primären Speicherfunktion aufgedeckt worden. Betroffen war hiervon die Datei "admin.php", in der die Systemrechte der Benutzer gespeichert werden. Durch diese Schwachstelle konnten Angreifer Privilegien ändern und Zugriff auf das System erlangen. Der Angriff wurde wie folgt ausgeführt [SF_MK]:

Im ersten Schritt wird von einem Angreifer eine präparierte Flashdatei erstellt. Diese präparierte Flashdatei enthält die "save_main" Funktion. Wird die Datei von einem MKPortal Administrator ausgeführt, werden die Zugangsrechte von allen Benutzern des Portals, auch von Gästen, auf Administrator-Rechte gesetzt.

```
var idg:Number = 9;
var p13:Number = 1;
var Salva:String = "Save+Permissions";
getURL("http://victim.com/mkportal/admin.php?ind=ad_perms&op=save_main",
"_self", "POST");
```

Codebeispiel 5: Präparierte Flashdatei

Beachtet werden muss hierbei, dass der "Save+Permissions" Code in der jeweils verwendeten MKPortal Sprache übersetzt wird. Bei einer italienischen MKPortal-Version zum Beispiel "Salva+questi+permessi".
Im zweiten Schritt wird dieses Flash-Programm in eine Webseite integriert. Diese Webseite wird dann im dritten Schritt mittels eines Links in einer Mail an den betreffenden Administrator der anzugreifenden Seite versendet. Dieser Systemadministrator wird anschließend für die eigentliche Attacke "missbraucht". Ruft der Administrator die präparierte Webseite auf, wird die Flash-Datei ausgeführt und alle Benutzer des Portals erlangen Administrator-Rechte.

2. **phpMyAdmin** (03.10.2006)

PhpMyAdmin ist eine freie PHP-Anwendung zur Administration von MySQL-Datenbanken. Durch eine im Oktober 2006 aufgedeckte CSRF-Schwachstelle können von Angreifern beliebige Aktionen auf einem angegriffenen System ausgeführt werden. Hierzu werden von einem Angreifer fremde Benutzer ausgenutzt, die Zugriffsrechte auf die angegriffenen Systeme besitzen. Der Angriff geht in mehreren Schritten vor. In einem ersten Schritt wird eine dynamische URL mit einem bestimmten Token verändert. Im zweiten Schritt werden in mehreren PHP-Dateien Variablen abgeändert. Betroffen sind hiervon die Dateien, "libraries/common.lib.php", "session.inc.php", und "url_generating.lib.php". Nähere Informationen über die Ausführung eines Angriffes wurden nicht veröffentlicht um Nachahmern keine Anleitung zu bieten [NVD_MY].

6.6 Informationslecks & Fehlermeldungen

Webanwendungen bieten einem Angreifer oft ungewollt Hilfestellung bei der Gewinnung von Informationen. Informationen können zum Beispiel aus unzureichenden oder zu umfangreichen Fehlermeldungen hergeleitet werden. Zusätzlich sind sogenannte Informationslecks Schwachstellen, da sie öffentlich zugängliche Informationen im Klartext speichern und somit bei der Vorbereitung von Angriffen helfen. Die OWASP Top Ten listet diese Schwachstelle an sechster Stelle auf und gibt mehrere Praxisbeispiele aus der näheren Vergangenheit an:

1. **CA - Security Command Center** (22.09.2006)

Nach Angaben des Herstellers besteht das Security Command Center primär aus einer Sicherheits-Management-Konsole mit drei integrierten Softwaremodulen. Diese Module überwachen die Zugangskontrolle (Access Management), die Erstellung und Verwaltung von Benutzerprofilen (Identity Management), sowie den Schutz vor internen und externen Bedrohungen (Threat Management). Die aufgedeckte Schwachstelle tritt auf, wenn von

einem Angreifer eine vorbereitete URL-Anfrage an das "ePPIServlet script" der Anwendung geschickt wird. Diese URL-Anfrage beinhaltet einen ungültigen Wert (’), der eine Fehlermeldung verursacht. Diese Fehlermeldung kann vom Angreifer ausgelesen werden und beinhaltet den vollständigen Installationspfad der Anwendung. Anhand dieses Installationspfades kann der Angreifer anschließend weitere Angriffe starten, da er nun die genauen Adressen der Programmdateien kennt [ISS].

2. **WordPress** (06.07.2006)

WordPress ist ein Weblog-System, dass zum Veröffentlichen persönlicher Beiträge genutzt wird. Der Schwerpunkt der Anwendung liegt auf einer ansprechenden Ästhetik, die Einhaltung von Webstandards des weiteren wird Wert auf eine hohe Benutzerfreundlichkeit gelegt. Die Schwachstelle der Anwendung geht von der Datei "index.php" aus, die auch von fremden Personen über das Internet ausgelesen werden kann. Wird zum Beispiel die Webanwendung mit einem ungültigen Seitenparameter aufgerufen geschieht Folgendes: Die Datei index.php wird erstellt und beinhaltet in Form einer SQL-Meldung Informationen über die Dateistruktur [NVD_WP].

3. **WorkforceROI** (18.06.2002)

Das Projektverwaltungsprogramm Workforce ROI erlaubte es Angreifern die Namen von Benutzern der Anwendung auszulesen. Hierfür wurde von dem Angreifer die öffentlich zugängliche Datei "datasource.asp" ausgelesen. In dieser waren die Benutzernamen des Programmes enthalten. Konnte der Angreifer in den Besitz der Benutzernamen gelangen, ist ihm schon die halbe Arbeit abgenommen worden, da er bei einem anschließenden Brute-Force Angriff nur noch die zugehörigen Kennwörter der Benutzer herausfinden muss.

7. Zusammenfassung

Das Hauptaugenmerk bei der Auswahl der Kriterien lag darin, dass diese insbesondere die spezifischen Schwachstellen von E-Commerce-Webseiten abdecken sollten. Aufgrund ihres Aufbaues können E-Commerce-Webseiten über Schwachstellen verfügen, die bei anderen Webseiten nicht vorkommen oder dort weniger bzw. keinen Schaden anrichten würden. Folglich müssen die Kriterien auf eine E-Commerce-Webseite angepasst werden. Damit die Kriterien angepasst werden können, musste in einem ersten Schritt spezifiziert werden, wie eine E-Commerce-Webseite aufgebaut ist. So enthält eine E-Commerce-Webseite eine Datenbank, die den Warenbestand verwaltet. Sie besitzt eine Bezahlplattform, die meist von einem externen Anbieter übernommen wird und über die der Bezahlvorgang abgewickelt wird. Des Weiteren verfügt sie über Webanwendungen, die die Webseite um Funktionen wie einen Warenkorb oder Ähnliches erweitern. Letzten Endes wird noch eine Serverstruktur benötigt, auf der die Webseite gehostet wird. Jeder dieser Bausteine bietet mögliche Schwachstellen und dadurch Ansatzpunkte für einen Angriff. Anhand dieser Bausteine konnten in dem nachfolgenden Kapitel verschiedenste Schwachstellen benannt werden. Um die Schwachstellen in eine geordnete Form zu bringen, die später auch bei der Erstellung des Kriterienkataloges beibehalten werden kann, wurde eine Einordnung nach der Vorgabe des Bundesamtes für Sicherheit in der Informationstechnik vorgenommen. Dies bestand darin, Schwachstellen anhand ihrer Ursachen einzuordnen. Die Ursachen für Schwachstellen lagen in Programmierfehlern, Konfigurationsfehlern, Konzeptionsfehlern und vom Benutzer verursachten Fehlern.

Während der Einsortierung der Schwachstellen hat sich gezeigt, dass aktuell die meisten Schwachstellen und somit Angriffspunkte in den Anwendungen, welche in die Webseiten integriert sind, zu finden sind (Webanwendungen). Diese These wurde durch die OWASP-Top Ten Kriterien unterstützt. Während zum Ende der 90er Jahre primär Webserver, also Hardwarekomponenten, angegriffen wurden, um diese durch eine Überlastung auszuschalten, werden aktuell vermehrt E-Shops angegriffen. Die Motivation der Hacker hat sich geändert. Inzwischen verfolgen Hacker das Ziel sich durch einen Datendiebstahl zu bereichern. Gerade dieser Umstand macht es für E-Commerce-Unternehmen zwingend notwendig, ihre Webseiten technisch abzusichern. Im gleichen Feld wie Bankseiten, gehören sie zu den lohnenswerten Zielen für einen Hacker, da er sich dort durch einen Datendiebstahl finanziell bereichern kann.

Im Anschluss an die Ausbildung der technischen Schwachstellen wurde ein kurzes Resümee gezogen, dass die generellen Risikopotentiale der an einem E-Commerce-Vorgang

beteiligten Gruppen aufzeigte. Hierbei wurde herausgestellt, dass es drei beteiligte Gruppen gibt. Die Anbieter von E-Commerce-Seiten, die Kunden, die auf das Webangebot der E-Commerce-Anbieter zugreifen, und die Provider, die den Anbietern durch Dienstleistungen, den Aufbau einer Webseite erleichtern. Jede dieser drei Gruppen nimmt auf Grund einer anderen Motivation an der Interaktion teil und verfügt über individuelle Risikopotentiale. Dieses Ergebnis wird benötigt, da es bei der späteren Gewichtung der Kriterien berücksichtigt werden sollte.

In dem dritten Kapitel erfolgte eine Validation bestehender Kriterienkataloge. Hierbei konnte herausgestellt werden, wie ein Kriterienkatalog aufgebaut und welche Typen von Kriterien verwendet werden können. Es konnte gezeigt werden, dass die etablierten Kriterienkataloge meist auf allgemeine IT-Sicherheitsrichtlinien prüfen und nicht die individuellen Bausteine einer E-Commerce-Webseite berücksichtigen. Somit können diese Kriterienkataloge nicht zuverlässig die gesetzte Aufgabe erfüllen, eine E-Commerce-Webseite auf ihre technische Sicherheit hin zu überprüfen. Eine Ausnahme der betrachteten Kriterienkatalogen bot der BSI-Grundschutz-Katalog. Er verfügt mit dem Bausteinkatalog, dem Gefährdungskatalog und dem Maßnahmenkatalog über umfangreiche Quellen, die eine angemessene Überprüfung ermöglichen. Hierdurch kann er viele Schwachstellen, die im zweiten Kapitel vorgestellt wurden, abdecken. Allerdings ist der Grundschutz-Katalog zu umfangreich, als das er praktikabel für die angestrebte technische Untersuchung und Absicherung einer E-Commerce-Webseite verwendet werden könnte. Vielmehr kann der Grundschutz-Katalog als eine Basis für den zu erstellenden Kriterienkatalog herangezogen werden, auf den die später gewählten Kriterien verweisen.

Der weitere Verlauf der Diplomarbeit bestand darin, dass der eigentliche Kriterienkatalog erstellt wurde. Dies begann mit einer kurzen Einführung, die dem Verständnis dienen sollte, was überprüft werden kann, und was nicht geprüft werden kann. Die gewählten Kriterien müssen eine Reihe von Bedingungen erfüllen, damit sie zur Anwendung kommen können. So müssen die Kriterien in einem Blackboxverfahren ein fremdes System von außen überprüfen. Es darf für eine Untersuchung keine Software auf dem zu prüfenden System installiert werden. Eine Überprüfung darf das getestete System nicht in seiner Funktion beeinträchtige. All diese Bedingungen müssen bei einer Überprüfung eingehalten werden. Weiterhin sollte im Vorfeld festgehalten werden, wie ein Kriterium zu der Verbesserung der technischen Sicherheit betragen kann. Das bedeutet insbesondere, dass Kriterien oft an ein System angepasst werden müssen, damit eine Überprüfung aussagekräftige und anwendbare Resultate liefert. Diese oft nötige Anpassung stellte die größte Problemstellung bei der Entwicklung der Kriterien da. Bei der gewünschten hochgradig automatisierten Überprüfung werden einige Schwachstellen nur oberflächlich behandelt werden können.

Folglich musste bei der Auswahl der Kriterien ein anwendbarer Mittelweg zwischen der Automatisierung und der Aussagengüte einer Überprüfung gefunden werden.

Nachdem diese Vorbedingungen definiert worden waren, konnte damit begonnen werden, die Kriterien aus dem zweiten Kapitel in eine Listenform zu bringen. Dabei wurde berücksichtigt, dass die nach einer Überprüfung aufgedeckten Schwachstellen zuständigen Personen in einem Unternehmen zugeordnet werden sollen. Hierfür wurden die Kriterien in drei Schwachstellenebenen (Protokoll-, Dienst- und Anwendungsebene) und eine Qualitätsebene unterteilt. Jede Schwachstellenebene bezieht sich auf einen bestimmten Personenkreis in einem Unternehmen. So werden zum Beispiel Kriterien der Protokollebene Systemadministratoren zugeordnet. Aufgedeckte Kriterien bzw. Schwachstellen in der Anwendungsebene wiederum werden an die Entwickler der Software weitergereicht, damit diese die Schwachstellen schließen können. Bei einigen Kriterien konnte allerdings keine eindeutige Zuordnung zum BSI-Grundschutz-Katalog vorgenommen werden. So gab es zum Zeitpunkt der Erstellung dieser Diplomarbeit im Grundschutz-Katalog noch keine Einträge für Cross-Site-Scripting. Dies mag daran liegen, dass diese Angriffsform erst in den letzten zwei Jahren stark zugenommen hat. Das Einsortieren der bekannten Schwachstellen hatte einen Liste von ca. 61 Kriterien zu folge.

Da nicht alle Kriterien gleichermaßen zu der Sicherheit einer E-Commerce-Webseite beitragen und die Anzahl an Kriterien zu umfangreich war, wurde eine Bewertung der Kriterien vorgenommen. Das Ziel hinter dieser Bewertung war es, sowohl die Anzahl der Kriterien zu verringern, als auch eine Gewichtung innerhalb der Kriterien einzuführen. Die Gewichtung erfolgte nach den Gesichtspunkten der Automatisierbarkeit, dem verbundenen Schadenspotential und der Verbreitung der Schwachstelle nach OWASP. Kriterien, deren Überprüfung gut normierbar ist, wurde ein hoher Grad an Automatisierung zugesprochen. Dies waren insbesondere Kriterien der Protokollebene und Kriterien aus dem Bereich der Qualitätsebene. Bei Kriterien der Dienst- und Anwendungsebene ist die erreichbare Automatisierung bei jedem Kriterium unterschiedlich. Das Schadenspotential eines Kriteriums wurde dahingehend bewertet, ob nur eine oberflächliche Veränderung einer Webseite, das Auslesen von Daten oder sogar das Löschen von Daten möglich wäre. Die letzte Bewertung erfolgte über die Verbreitung des Kriteriums. Hiefür wurde die OWASP Top Ten Liste der aktuell brisantesten Schwachstellen von Webanwendungen, zu Rate gezogen. Bei der Auswahl der Kriterien lässt sich festhalten, dass gegenwärtig, aber auch zukünftig, durch eine automatisierte Überprüfung nicht alle technischen Schwachstellen aufgedeckt werden können. Viele Schwachstellen können nur überprüft werden, wenn dynamisch und intelligent auf den Kontext einer Internetseite eingegangen wird. Für einen Scanner gestaltet sich die Aufgabe des Erkennens von logischen Zusammenhängen sehr schwierig [siehe GJ]. Aufgrund dieser Begebenheit fand bei der Wahl der Kriterien eine

Ausrichtung auf sogenannte Low-Hanging Fruits (LHF) statt. Diese LHF zeichnen sich durch die Eigenschaften aus, dass sie sowohl gut automatisiert, als auch mit einer hohen Aussagengüte überprüft werden können.

Im Anschluss an die Erstellung des Kriterienkataloges wurde erläutert, auf welche Weise der Katalog abgearbeitet werden kann. Eine Abarbeitung mittels manueller Überprüfung kam nicht in Frage, da der damit verbundenen Aufwand zu hoch wäre. Es musste folglich ein Tool gefunden werden, dass diese Aufgabe übernehmen konnte. Hierfür kamen zwei Programmklassen in Frage. Zum einen sogenannte Portscanner und zum andern sogenannte Vulnerability Scanner (Web Application Scanner). Portscanner konnten nach einer Begutachtung ausgeschlossen werden, da sie aufgrund ihres eingeschränkten Funktionsumfanges nicht alle Kriterien abdecken würden. Web Application Scanner schienen hingegen die richtige Wahl zu sein, da mit ihnen alle Kriterien überprüft werden können. Nachdem der Aufbau eines Application Scanner und seine Funktionsweise vorgestellt wurde, konnte damit begonnen werden einen Überblick über die am Markt verfügbaren Scanner zu erlangen. Zu Beginn des Fünften Kapitels wurden die frei verfügbaren Application Scanner BOSS/NESSUS und Whisker vorgestellt. Beide können als Basis genommen werden, in die die erarbeiteten Kriterien eingegliedert werden. Das BOSS-Tool bot hierfür die bessere, wenn auch nicht perfekte Lösung an. So ist das Tool eigentlich auf das Scannen von Schwachstellen auf der Netzwerkebene und nicht auf das Scannen von Schwachstellen auf der Anwendungsebene ausgelegt. Zusätzlich muss für jedes Kriterium ein spezielles Plugin konfiguriert werden, das die Handhabung des Scanners erschwert. Nach der Vorstellung der Open Source Scanner wurde ein Überblick über die aktuell am Markt etablierten kommerziellen Application Scanner geboten. Hier konnte gezeigt werden, welchen Funktionsumfang diese Programme bieten und insbesondere welche Kriterien von ihnen betrachtet werden. Die Kosten für die Scanner und die Untersuchung einer IP-Adresse belaufen sich zwischen 525€ und 15.000€. Diese Kosten werden nur von größeren Firmen investiert werden können. Dieser Umstand ist insbesondere interessant, da Trusted Shops einen Service im Bereich Technische Sicherheit (Application Scanner) und ein zugehöriges Zertifikat anbieten möchte. Das Produkt von Trusted Shops ist auf kleine bis mittelgroße E-Commerce-Anbieter ausgerichtet und wird sich aufgrund dessen in einer niedrigeren Preiskategorie bewegen. Besonderer Wert bei dem Vergleich der Application Scanner wurde auf die folgenden Eigenschaften gelegt:

- Umfang der Prüfkriterien
- Erweiterbarkeit durch eigene Kriterien
- Güte der Meldungen auf gefundene Schwachstellen
- Geschwindigkeit einer Überprüfung
- Handhabung der Software

• Unterstützung des Benutzers bei der Schließung von Schwachstellen

Die Verwendung von Application Scannern ist zum aktuellen Zeitpunkt die geeigneteste Lösung, um eine Webseite auf Schwachstellen hin zu untersuchen. Als Fazit der Untersuchung der Application Scannern muss aber festgehalten werden, dass diese über zwei wesentliche Mängel verfügen. Zum Einen kann ein Scanner über eine nicht ausreichende Datenbank von Prüfkriterien verfügen. Damit ein Scanner ein optimales Untersuchungsergebnis liefern kann, muss seine Schwachstellendatenbank regelmäßig mit neuen Signaturen von Schwachstellen aktualisiert werden. Trotzdem können Signaturen fehlerhaft sein, oder es können zu Schwachstellen gar keine Signaturen vorhanden sein. Zum anderen können falsche Meldungen von gefundenen Schwachstellen (false Positives) erzeugt werden, die das Vertrauen des Benutzers in ein Untersuchungsergebnis beeinträchtigen.

7.1 Fazit und Ausblick

Im Laufe dieser Arbeit konnte über mehrere Schritte hinweg ein Kriterienkatalog entwickelt werden, anhand dessen eine Webseite auf ihre technische Sicherheit hin untersucht werden kann. Die technische Sicherheit einer Webseite kann erreicht bzw. verbessert werden, indem alle bekannten technischen Schwachstellen aufgedeckt und geschlossen werden. Jede offene Schwachstelle bietet einem Angreifer einen Ansatzpunkt für einen Angriff. Wird eine Webseite mittels eines Kriterienkataloges auf Schwachstellen überprüft, können diese im Anschluss geschlossen werden. Demzufolge steigt durch eine Untersuchung und einer nachfolgenden Systempflege die Sicherheit einer Webpräsenz.

Für die Entwicklung von Application Scanner und Zertifizierungsfirmen von IT-Sicherheit kann die folgende Prognose gemacht werden. Insbesondere kleine E-Commerce-Unternehmen entdecken das Internet immer mehr als Medium, um dort ihre Geschäftsideen und ihre Produkte zu präsentieren. Gleichzeitig steigt das Bedrohungspotential für Webauftritte, speziell für die Anbieter und die Kunden von Webseiten im E-Commerce-Bereich. Die aktuelle Kriminalitätsstatistik aus dem Jahr 2006 zeigt, dass Computerbetrügereien und in diesem Feld Phishing-Attacken immer mehr zunehmen [AIK]. Aus diesen beiden Umständen, dem steigenden Markt und der gleichzeitig steigenden Gefährdung, kann mit einem ebenfalls steigenden Bedarf für Sicherheitslösungen gerechnet werden. Diese Aufgabe übernehmen die Application Scanner und die anbietenden Firmen. Weiterhin wird es für E-Commerce-Unternehmen immer wichtiger werden, bei einem

Kunden Vertrauen zu erwecken. Hierfür können sie das Zertifikat benutzen, dass sie von der Zertifizierungsstelle erhalten.

Die Entwicklung der Application Scanner weißt in die folgende Richtung. Es kann davon ausgegangen werden, dass in nächster Zukunft die Scanner immer intelligenter bei der Suche nach Schwachstellen vorgehen werden. Dadurch wird die Güte einer Überprüfung stetig steigen. Die Application Scanner müssen sich dabei mehreren großen Herausforderungen stellen. Zum Einen Schwachstellen, die auf Logikfehlern beruhen oder nur aus dem Kontext einer Webseite erkannt werden können. Zum Anderen werden immer weniger False Postives Meldungen vorkommen da die Signaturen für Schwachstellen immer ausgereifter werden.

Im Rahmen dieser Arbeit wurden Sachverhalte nur am Rande aufgedeckt, deren detaillierte Erarbeitung den Rahmen dieser Arbeit überschritten hätte. Mögliche Erweiterungen, die noch in einer Erweiterungsarbeit realisiert werden könnten wären:

- Entwicklung von Vorgehensweißen zur Überprüfung der Kriterien der Kataloges.
- Entwicklung einer Test-Webseite bestehen aus einem Webserver, einer Datenbank, einer Bezahlplattform und typischen E-Commerce-Anwendungen, um diese manuell oder automatisch auf Schwachstellen zu untersuchen.
- Integration des Kriterienkataloges in einen bestehenden Web Application Scanner.
- Entwicklung eines Application/Vulnerability Scanners.
- Integration des Kriterienkataloges in den entwickelten Web Appllication Scanner.

Literaturverzeichnis

[AC] **Ac**unetix
▪ Website Tiefen-Audit
http://www.acunetix.de/site-audit/web-applications.htm

[AI] **A**rbeitsgruppe **I**dentitätsschutz Internet
▪ Phishing-Attacken per E-Mail haben eine Erfolgsrate von bis zu 14 %
https://www.a-i3.org/content/view/901/214/

[AIK] **A**rbeitsgruppe **I**dentitätsschutz Internet
▪ **K**riminalitätsstatistik: Straftaten gehen zurück, Computer-Betrügereien nehmen zu
https://www.a-i3.org/content/view/1154/214/
http://www.bmi.bund.de/Internet/Content/Common/Anlagen/Broschueren/2007/Polizeiliche__Kriminalstatistik__2006__de,templateId=raw,property=publicationFile.pdf/Polizeiliche_Kriminalstatistik_2006_de.pdf

[ARD] **A**rbeitsgemeinschaft der öffentlich-rechtlichen **R**undfunkanstalten der Bundesrepublik **D**eutschland
▪ Verfassungsschutz warnt vor Chinas Industriespionen
http://www.tagesschau.de/aktuell/meldungen/0,1185,OID6390336_TYP6_THE_NAV_REF1_BAB,00.html

[ATA] Ferguson, Dave:
▪ **A**ttacking **t**he **A**pplication
http://www.owasp.org/images/2/2c/KC_Dec2006_Attacking_The_App.pdf

[BD_A] **B**achfeld, **D**aniel:
▪ Cross-Site-Scripting: Datenklau über Bande.
Heise, Hannover 2003
http://www.heise.de/security/artikel/print/38658

[BD_B] **B**achfeld, **D**aniel:
▪ Giftspritze: SQL-Injection - Angriff und Abwehr.
Heise, Hannover 2004
http://www.heise.de/security/artikel/43175

[BG] **B**rands, **G**ilbert :
▪ IT-Sicherheitsmanagement: Protokolle, Netzwerksicherheit, Prozessorganisation.
Springer, Berlin 2005

[BH] **B**alzert, **H**elmut:
▪ Lehrbuch Grundlagen der Informatik.
Spektrum Akademischer Verlag, Heidelberg, 2004

[BJ] **J**ürgen, **B**erke:
▪ Wirtschaftsspionage: Ohne Spuren
http://www.handelsblatt.com/news/printpage.aspx?_p=204016&_t=ftprint&_b=1180306

[BKSW] **B**reitschaft, Markus; **K**rabichler. Thomas; Dr. **S**tahl, Ernst; **W**ittmann Georg:
▪ Sichere Zahlungsverfahren für E-Government
Universität Regensburg, Bundesamt für Sicherheit in der Informationstechnik 2005
www.bsi.bund.de/fachthem/egov/download/4_Zahlv.pdf

[BMW] "**BMW** von Google gekickt"
http://www.suchmaschinen-optimierung-seo.info/sosblog/2006/02/04/bmw-von-google-gekickt/

[BOSS_A] BSI-**BOSS**
▪ BSI OSS Security Suite
http://www.bsi.de/produkte/boss/index.htm#hinweise

[BOSS_B] BSI-**BOSS**
▪ BSI-Boss Images
http://bsi.customer.ktit.de/produkte/boss/boss2/doc/images/einsatz-szenarien.png

[BP] **B**oehm, **P**eter:
▪ Cisco Security Agent lässt sich austricksen.
Heise, Hannover 2004
http://www.heise.de/security/news/meldung/53192

[BSI_A] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ Sicherheit von Webanwendungen, Maßnahmenkatalog und Best Practices
http://www.bsi.de/literat/studien/websec/WebSec.pdf

[BSI_CC] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ BSI-Kurzinformationen: Common Criteria (ISO/IEC 15408)
http://www.bsi.bund.de/literat/faltbl/F06CommonCriteria.htm

[BSI_IS] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ **I**nternet-**S**icherheit, Gefährdungen.
http://www.bsi.de/fachthem/sinet/gefahr/

[BSI_IT] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ IT-Sicherheitskriterien und Evaluierung nach ITSEC.
http://www.bsi.de/zertifiz/itkrit/itsec.htm

[BSI_ITIL] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ "IT Infrastructure Library (ITIL) und Informationssicherheit"
http://www.bsi.de/literat/studien/ITinf/index.htm

[BSI_PEN] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ Studie: Durchführungskonzept für Penetrationstests
http://www.bsi.bund.de/literat/studien/pentest/penetrationstest.pdf

[BTSU] **B**iege, **T**homas:
▪ Sicherheitsrelevante Programmierfehler.
SuSe Papers.
http://www.suse.de/~thomas/papers/SecProg/Sicherheitsrelevante Programmierfehler.pdf

[BUCC] **B**undesamt für **S**icherheit in der **I**nformationstechnik:
▪ Gemeinsame Kriterien für die Prüfung und Bewertung der Sicherheit von
Informationstechnik.
http://www.bsi.bund.de/cc/

[BW_€] **B**usiness**W**eek
▪ A Net Security Outfit with a New Kind of Lock
http://www.businessweek.com/technology/content/feb2001/tc20010227_320.htm

[CADV] **C**ert **Adv**isories:
▪ Technical Cyber Security Alerts 2003.
http://www.cert.org/advisories/#2003

[DFS] **D**olle, Wilhelm; **F**ritzinger,Thomas; **S**chmidt, Jürgen:
Heimliche Hintertüren: Rootkits aufspüren und beseitigen.
Heise, Hannover 2003.
http://www.heise.de/security/artikel/38057

[DFW] **D**ehms, Björn; Dr. **F**uhrberg, Kai; Dr. **W**olf, Stefan:
▪ Sicherer Internet-Auftritt im E-Government.
Bundesamt für Sicherheit in der Informationstechnik 2005
http://www.bsi.de/fachthem/egov/download/4_IntAuf.pdf

[DR] **D**eraison, Renaud:
▪ Nessus Benutzerhandbuch
http://bsi.customer.ktit.de/produkte/boss/boss1/doc/users-manual-de-20050426.pdf

[EC_A] **E**ckert, **C**laudia:
▪ IT-Sicherheit.
Oldenbourg Verlag, Darmstadt 2004

[EC_B] **E**ilers, **C**arsten:
▪ About Security #57: Angriffe auf TCP/IP – Spoofing.
http://entwickler.de/zonen/portale/psecom,id,99,news,29003,.html

[EYE] Retina **eEye**
▪ eEye Product Tours
http://www.eeye.com/html/resources/tours/retina/index.html

[EMS] Eggert, Bodo; Messerschmidt, Michel; Seedorf, Jan:
▪ Klassifikation von bösartiger Software und aktuelle Testergebnisse des Virus Test Centers von AntiMalware-Software unter Linux.
http://www.semmel.ch/Linuxtag-DVD/talks/172/paper.html

[FAZ] Schmidt, Holger:
▪ Computerhacker wollen Geld statt Ruhm und Ehre,
Frankfurter **A**llgemeine **Z**eitung
http://www.faz.net/s/RubEC1ACFE1EE274C81BCD3621EF555C83C/Doc~EBE533C8E1CDD430EB0020DBFEB0521FD~ATpl~Ecommon~Scontent.html

[FJ] **F**orristal, **J**eff;
▪ Vulnerability Assessment Scanners
http://www.networkcomputing.com/1201/1201f1b1.html

[FSN] **F-S**ecure
▪ Netscape.com hacked
http://www.f-secure.com/weblog/archives/archive-072006.html#00000927

[GB] **G**raf, **B**irgit:
▪ XSS-Wurm legt MySpace lahm.
Heise, Hannover 2005
http://www.heise.de/newsticker/meldung/65039

[GFI] **GFI** LANguard
▪ Network Secutity Scanner
http://www.gfisoftware.de/de/lannetscan/lanscanfeatures.htm

[GFI_€] **GFI** LANguard
▪ Network Secutity Scanner / EventsManger **(Preislisten)**
http://www.gfisoftware.de/pricing/pricelist.aspx?product=lanss&curr=eur&lang=de (N.S.S)

http://www.gfisoftware.de/pricing/pricelist.aspx?product=esm&curr=eur&lang=de (EventsManager)

[GFI_R] **GFI** LANguard
▪ GFI EndPointSecurity 3 **R**eportPack
http://www.gfisoftware.de/pricing/pricelist.aspx?product=esec&curr=eur&lang=de

[GJ] **G**rossman, **J**eremiah:
▪ Automated Scanning vs. The OWASP Top Ten
http://www.whitehatsec.com/home/assets/OWASPTop10ScannersF.pdf

[GS] **Gr**aegert, **S**teve:
▪ Wie funktioniert ARP-Spoofing?
http://eth0.graegert.com/?section=docsys&cmd=details&id=15

[GSK_B1] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **B**austein: B5.4 Webserver.
http://www.bsi.de/gshb/deutsch/baust/b05004.htm

[GSK_B2] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ IT-Grundschutz - Basis für IT-Sicherheit
http://www.bsi.bund.de/gshb/deutsch/baust/01002.htm

[GSK_G1] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **G**efährdungen: G 5.48 IP-Spoofing.
http://www.bsi.bund.de/gshb/deutsch/g/g05048.htm

[GSK_G2] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **G**efährdungen: G. 5.78 DNS-Spoofing.
http://www.bsi.bund.de/gshb/deutsch/g/g05078.htm

[GSK_G3] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **G**efährdungen: G 3.43 Ungeeigneter Umgang mit Passwörtern.
http://www.bsi.de/gshb/deutsch/g/g03043.htm

[GSK_M1] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **M**aßnahmen: M5.39 Sicherer Einsatz der Protokolle und Dienste.
http://www.bsi.bund.de/gshb/deutsch/m/m05039.htm

[GSK_M2] BSI IT-**G**rund**s**chutz-**K**ataloge:
▪ **M**aßnahmen: M2.11 Regelung des Passwortgebrauchs.
http://www.bsi.de/gshb/deutsch/m/m02011.htm

[HA] **Ha**ack IT
http://www.inn.de/shop/katalog.asp/shop/security/kat/eEye+Digitial+Security

[HF] **H**ahne, **F**elix:
▪ Ineraktive Webseites.
Poing 2003

[HH] **M**ichaelsen, **N**ils;
▪ Penetrationstests
http://www.informatik.uni-hamburg.de/AGN/papers/doc/diparb_michaelsen.pdf

[HK] Dr. von **H**elden, Josef; Dr. **K**arsch, Stefan:
▪ Grundlagen, Forderungen und Marktübersicht für Intrusion Detection Systeme (IDS) und Intrusion Response Systeme (IRS).
http://www.stud.tu-ilmenau.de/~traenk/ids-stud.pdf

[HMS] **H**oll, Friedrich-L; **M**enzel, Katharina; **M**orcinek, Peter; **M**ühlberg, Jan Tobias; **S**chäfer, Ingo; **S**chüngel, Hanno:
▪ Studie zum Innovationsverhalten deutscher Software-Entwicklungsunternehmen.
Berlin, November 2006.
http://www.innovationsanalysen.de/de/download/Innovationsverhalten_deutscher_SW-Entwicklungsunternehmen.pdf

[IEYE] **i-eye** Security
▪ http Banner Sammler
http://www.i-eye.de/dokumentation/HTTP_Banner_Sammler.php

[ISS] IBM **I**nternet **S**ecurity **S**ystems
▪ CA eTrust Security Command Center ePPIServlet path disclosure
http://xforce.iss.net/xforce/xfdb/29102

[IW] **I**nformation**W**eek:
▪ IT-Security Studie 2006
http://i.cmpnet.com/informationweek.de/downloads/pdf/mktResearch/it_security_2006_kurzversion.pdf

[JZ] Janowicz, Krzysztof:
▪ Sicherheit im Internet.
O'Reilly, Bejing 2002

[KE] **K**emper, Alfons; **E**ickler, André:
▪ Datenbanksysteme.
Oldenburgverlag, München 2004

[KES] **KES** – Die Zeitschrift für Informationssicherheit:
▪ Die Microsoft Sicherheitsstudie.
http://download.microsoft.com/download/a/8/8/a885d51f-6684-4f8a-af41-c87fd082242a/kes-Microsoft-Studie2004-Sonderdruck.pdf

[KL] **K**orrektur**l**eser:
Bell, Katharina; Blank, Susanne; Burbach, Anne; Dieckmeyer, Anja; Eichler, Kerstin; Etges, Christina; Franker, Christel und Karl; Göhring, Christina; Griesel, Hanni; Hoffmeister, Elisabeth; Hupf, Katharina; Körtgen, Verena; Reinecke, Katharina; Schubert, Beate; Wiegmann, Mareen

[KM] **K**rausz, **M**ichael:
▪ Sicherheit im Internet, E-Mail und E-Commerce
http://sicherheitskultur.at/krausz_internet.htm

[LH] **L**oeser, **H**enrik:
▪ Web-Datenbanken: Einsatz objekt-relationaler Datenbanken für Web-Informationssysteme.
Springer, Berlin 2001.

[LH_A] **L**azarek, **H**orst:
▪ Kriteriensysteme
http://web.inf.tu-dresden.de/~lvinfhl4/download/dsds/Datensicherheit_Kriteriensysteme.pdf

[LJ] **L**ong, **J**ohnny:
▪ The Google Hacker's Guide: Understanding and Defending Against the Google Hacker
http://johnny.ihackstuff.com/security/premium/The_Google_Hackers_Guide_v1.0.pdf

[LL] **L**an**L**ine
▪ Gelungener Web- Application-Scanner
http://www.lanline.de/article.html?thes=&art=/articles/2006008/30758350_ha_LL.html

[NST] **N-St**alker
▪ The Web Security Specialists
http://nstalker.com/products/

[NST_€] **N-St**alker
▪ Pricelist
https://secure.nstalker.com/shopping/

[NUKE] Bachfeld, Daniel:
▪ PHP-**Nuke** verrät Passwörter
http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/79934&words=SQL%20injection%20Injection

[NVD_JV2] **N**ational **V**ulnerability **D**atabase
▪ Joonas Viljanen **JV2** Folder Gallery
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2007-0329

[NVD_MY] **N**ational **V**ulnerability **D**atabase
▪ php**My**Admin
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2006-5116

[NVD_MOD] **N**ational **V**ulnerability **D**atabase
▪ Integra**MOD** Portal
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2006-4369

[NVD_WP] **N**ational **V**ulnerability **D**atabase
▪ WordPress 2.0.3
http://nvd.nist.gov/nvd.cfm?cvename=CVE-2006-3389

[OS_XP] **O**perating **S**ystem Documentation Project:
http://www.operating-system.org/betriebssystem/_german/bs-windows.htm

[OWASP] **O**pen **W**eb-Application **S**ecurity **P**roject:
▪ The ten most critical web application security vulnerabilities 2007

http://www.owasp.org/images/c/c7/OWASP_Top_10_2007_RC1.pdf

[PL] **P**eterson, **L**arry:
▪ Computernetze.
dpunkt-Verlag, Heidelberg: 2000

[PP] Kleijn Alexandra:
▪ **P**aypal-**P**hishing via Cross-Site-Scripting
http://www.heise.de/newsticker/meldung/74369

[RC] **R**ütten, **C**hristiane; Bachfeld, Daniel:
▪ Phishing-Loch in Volksbanken-Seiten geschlossen
http://www.heise.de/newsticker/meldung/86607
http://www.heise.de/newsticker/meldung/76258

[RM] **R**ennhard, **M**arc:
▪ Aktuelle Sicherheitsprobleme im Internet: Angriffe auf Web-Applikationen
http://home.zhwin.ch/~rer/presentations/20061110_SiemensDozierendentag.pdf

[RP] **R**ieber, **P**hilipp:
▪ Dynamische Webseiten in der Praxis.
Mitp-Verlag 2006

[RZ] **R**uef, Marc; **Z**umstein, Simon:
▪ Sicherheitslücke Mensch.
http://www.scip.ch/publikationen/smss/scip_mss-19_07_2005-1.pdf

[SA] **S**eifert, **A**strid:
▪ Wirtschaftsspionage: Trojaner-Autor wohnte in Deutschland.
Heise, Hannover 2005
http://www.heise.de/security/news/meldung/60064

[SAP] **SAP** Info:
▪ Cross-Site Reference Forgery (XSRF/CSRF)
http://www.sap.info/public/DE/de/glossary/de/glossaryletter/Word-29299459e3a24c6b36_glossary/C

[SD] **S**tärk, **D**oris:
▪ Leitfaden für die Einrichtung einer Internetvertriebsplattform.
Bundesamt für Sicherheit in der Informationstechnik 2005
www.bsi.bund.de/fachthem/egov/download/5_EShop.pdf

[SEC] **Sec**ureNet:
▪ Security, Phishing.
http://www.securenet.de/papers/Phishing.html

[SECO] **SE**C **Co**nsult
▪ Tools für automatisierte Security Scans
http://www.sec-consult.com/fileadmin/Whitepapers/tools.pdf

[SECW] **Sec**ureNet:
▪ Web Applikation Scanner
http://www.securenet.de/front_content.php?idcatart=93

[SF_MK] **S**ecurity **F**ocus
▪ **Mk**Portal "All Guests are Admin" Exploit
http://www.securityfocus.com/archive/1/archive/1/455894/100/100/threaded

[SH] **S**chumacher, M; **H**urler, M:
▪ Systematische Analyse von Schwachstellen.
TU-Darmstadt
http://www.ito.tu-darmstadt.de/publs/pdf/guug.pdf

[SIN] **S**icher **i**m **N**etz
▪ MBSA Microsoft Baseline Security Analyzer
https://www.sicher-im-netz.de/content/sicherheit/ihre/mittelstand/db/03_Risikoanalyse/MBSA2_0.doc

[SJ_PHP] **J**ürgen **S**chmidt
▪ Das **PHP**-Dilemma
http://www.heise.de/security/artikel/84149/3

[SM] **S**eiler, **M**artin:
▪ Software simuliert Hacker-Denkweise.
http://www.tecchannel.de/news/themen/sicherheit/442201/

[SPE] Bachfeld, Daniel:
▪ Team Speak-Forum verbreitet Trojaner
http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/88285&words=SQL%20injection%20Injection

[SPI] **Spi**dynamcis labs research brief:
▪ Detecting, Analyzing, and Exploiting Intranet Applications using JavaScript.
http://www.spidynamics.com/assets/documents/JSportscan.pdf

[SR] **S**penneberg, **R**alf:
▪ Linux-Firewalls mit iptables & Co.
Addison Wesley, München: 2006
http://www.opensource-training.de/PDFs_FW/3827321360%20Kapitel%2004.pdf

[ST] **S**chreiber, **T**homas:
▪ Session Riding: A Widespread Vulnerability in Today's Web Applications
SecureNet GmbH 2004
http://www.securenet.de/papers/Session_Riding.pdf

[ST] **S**transky**,** Sabine; **T**aschl Mario:
▪ Programmiersprachen.
TU Wien, 2005
http://www.big.tuwien.ac.at/teaching/offer/ws05/gwa_ps/D2.pdf

[STU] Briegleb Volker:
▪ **Studi**VZ … Hacken verboten
http://www.heise.de/newsticker/result.xhtml?url=/newsticker/meldung/86879&words=SQL%20injection%20Injection

[WA] **W**ussow, **A**ndre:
▪ Sichere Webanwendungen.
Entwickler.press, 2007

[WD] **Web**Inspect
▪ WebInspect **Da**tasheed
http://www.spidynamics.com/assets/documents/WebInspect_DataSheets.pdf

[WF] **W**atch**f**ire
▪ Produkte AppScan
http://security.media-solutions.de/de/p_AppScan.html

[WFS] **W**atch**f**ire
▪ App**S**an 7.0
http://security.media-solutions.de/download/AppScan_7_0.pdf

[WIC] **W**eb**I**nspect
▪ Costs (Lizenzen)
http://www.securityinnovation.com/security-report/october/tools/WebInspect.htm

[WIK] **Wik**ipedia
▪ **Pe**netrationstest.
http://de.wikipedia.org/wiki/Penetrationstest_%28Informatik%29

[WIV] **Wi**kipedia
▪ **V**ulnerability **S**canner
http://de.wikipedia.org/wiki/Vulnerability_Scanner

[WTS] **W**ack John, **T**racy Miles, **M**urugiah Souppaya
▪ Computer Security
http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf

[XSS_A] **XSS-A**rchiv
http://www.xssed.com/

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Andere als die angegebenen Quellen und Hilfsmittel habe ich nicht verwendet und die den Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht.

Koblenz, Mai 2007