

Grundlagen der Sicherheit bei Webanwendungen

Christian Gekeler
Hochschule für Technik Stuttgart

Januar 2007

Inhaltsverzeichnis

1. Einleitung
2. Allgemeines über Webanwendungen
 - 2.1 Beispiel für eine Webanwendung
 - 2.2 Vor- und Nachteile von Webanwendungen
 - 2.3 Sicherheitsrisiken und Angriffe im Allgemeinen
3. Ebenenmodell
 - 3.1 Aufbau
 - 3.2 Bereiche der Sicherheit
4. Gefahren und Gegenmaßnahmen
 - 4.1 Diebstahl von Anmeldeinformationen
 - 4.2 Offenlegung vertraulicher Daten
 - 4.3 Abhören des Netzwerks
 - 4.4 Sitzungsübernahmen
 - 4.5 Denial-of-Service (DoS)
 - 4.6 Protokollierung
5. Fazit
6. Literatur

1. Einleitung

Jeder nutzt heutzutage beinahe täglich Anwendungen wie Google, Ebay oder auch Internet Banking und genießt den Komfort den sie einem bieten. In unserer Zeit sind Webanwendungen mittlerweile fast nicht mehr wegzudenken. Sie bieten sehr große Möglichkeiten, verschiedene Prozesse zu simplifizieren, wie z.B. Bestellungen (z.B. Amazon.de) oder Auktionen (z.B. Ebay). Immer mehr werden Geschäftsprozesse im Internet abgebildet und durchgeführt. Hierzu zählen Prozesse zwischen Geschäftspartnern und Prozesse zwischen Unternehmen und deren Kunden (eBusiness).

Da das Web in der Regel fast jedem zu Verfügung steht, können nicht nur die potenziellen Benutzer auf den angebotenen Service zugreifen. Insbesondere ist die Anwendung auch für diejenigen zugänglich, deren Ziel es ist solche Systeme auszuspionieren oder zu hintergehen.

Aus diesem Grund ist einer der wichtigsten Aspekte solcher Anwendungen die Sicherheit für Benutzer und Unternehmen. Diese muss in ausreichendem Maße gewährleistet sein. Dieser Punkt wird häufig beim Optimieren von Prozessen vernachlässigt. Anwendungen werden oft in Betrieb genommen, ohne zu wissen ob die Sicherheit der Anwendung ausreichend Beachtung gefunden hat.

Dabei reicht der Bereich dieser Anwendungen von sehr einfachen bis hin zu sehr komplexen Systemen. Unabhängig davon sollte den Gesichtspunkten der Sicherheit immer ausreichend Beachtung geschenkt werden. Oft wird aufgrund von Zeit- oder Geldmangel diesem Aspekt nicht genügend Aufmerksamkeit gewidmet. Es ist von großer Wichtigkeit, die Taktiken der Angreifer und deren mögliche Ziele zu kennen, um die Erfolgswahrscheinlichkeit dieser Angriffe zu minimieren.

Zur besseren Verteilung der Zuständigkeiten für die Gegenmaßnahmen die eingebaut werden müssen existiert ein Ebenen Modell. Es wird im weiteren Verlauf genauer betrachtet. Dieses Modell erleichtert die Strukturierung und Zuteilung der Aufgaben im Bereich Sicherheit während der Entwicklung einer Webanwendung. Das Organisieren dieser Aufgaben wird bei bei Entwicklungen dieser Art immer wichtiger um die Sicherheit zu garantieren. Ein weiteres Problem stellt das oftmals nicht vorhandene Bewusstsein der Benutzer für die Gefahren im Internet dar.

Es existieren einige bekannte Gefahren für im Internet öffentlich zugängliche Anwendungen. Um diesen Gefahren nicht schutzlos ausgesetzt zu sein existieren gleichermaßen bekannte Gegenmaßnahmen. Diese sind problemlos im Internet zu finden. Einige davon werden im Weiteren Verlauf noch genauer erwähnt. Nun aber zuerst zu den Grundlagen einer webbasierten Anwendung.

Anmerkung: Die mit runden Klammern gekennzeichneten Begriffe sind unter 6. Glossar zu finden. Eckige Klammern markieren einen Verweis auf eine unter 7.Literatur angegebene Quelle.

2. Allgemeines über Webanwendungen

2.1 Beispiel für eine Webanwendung

Ein Betrieb verwaltet firmenspezifische Daten mit einer Softwareanwendung. Diese Software ist für alle Mitarbeiter zugänglich. Da die Anwendung aber auch für Kunden und externe Mitarbeiter der Firma zugänglich sein muss, wird diese als Webanwendung zur Verfügung gestellt. Den Benutzern dieses Systems ist es nun jederzeit möglich, die aktuellen Daten über einen Webbrowser abzurufen oder zu bearbeiten. Interne sowie externe Geschäftsabläufe werden in dieser Software abgebildet. Dies ist nur ein einfaches Beispiel für eine Webanwendung. In vielen Fällen werden in einer Webanwendung ganze Unternehmensprozesse oder darüber hinaus unternehmensübergreifende Prozesse abgebildet. Bei diesen Prozessen handelt es sich oft um vertrauliche oder sicherheitskritische Abläufe, wie z.B. bei Banken oder Versicherungen. Die Übertragung wird hier das HTTP-Protokoll (Hyper Text Transfer Protokoll) verwendet.

2.1 Über Webanwendungen

Eine Web-Anwendung ist ein Softwaresystem, das auf Spezifikationen des World Wide Web Consortium (W3C)(1) beruht und Web-spezifische Ressourcen wie Inhalte und Dienste bereitstellt, die über eine Benutzerschnittstelle, den Web-Browser(2), verwendet werden. In den meisten Fällen besteht diese aus einer Architektur in 3 Phasen:

1. Präsentation im **Browser** des Benutzers
2. Anwendungen auf dem **Web-/Anwendungsserver**(3)
3. **Datenbankserver**(4)

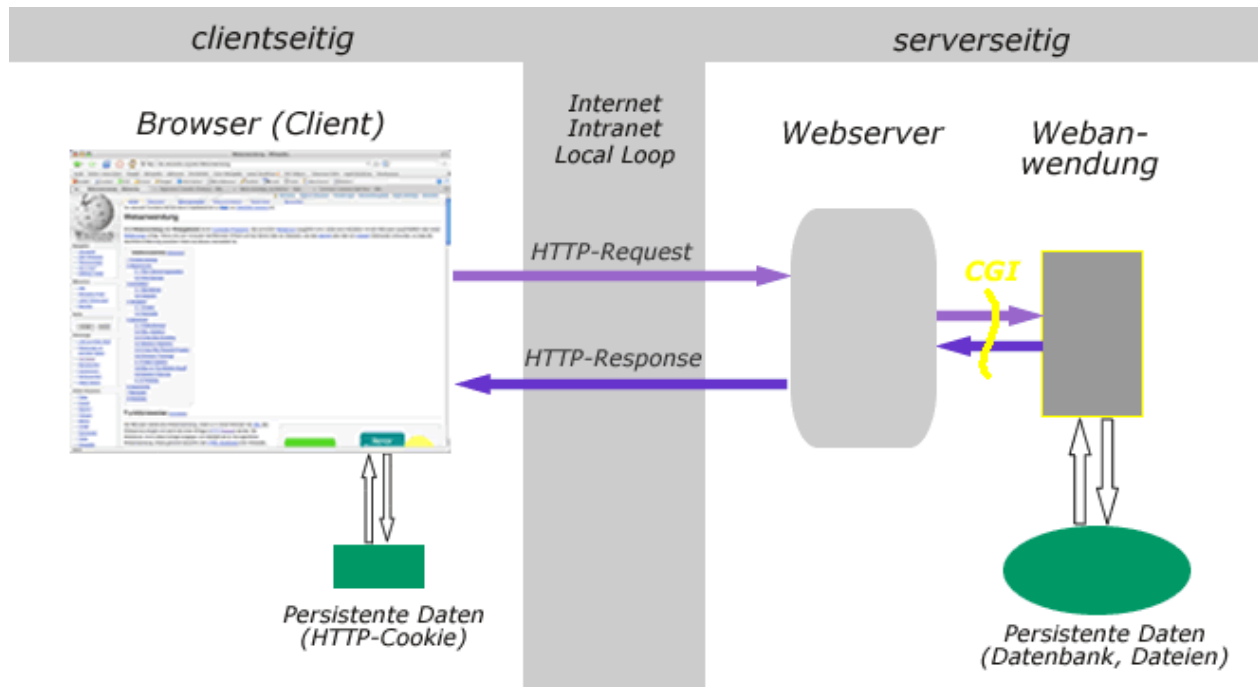


Abb 1: Schematischer Datenfluss einer Webanwendung [2]

Abbildung 1 zeigt den Ablauf einer Webanwendung. Der Client generiert einen **HTTP-Request** (Anfrage), der über das Netzwerk zu dem zuständigen Webserver gelangt. Dieser leitet die Anfrage an die Webanwendung weiter, die aus den mitgelieferten Parametern der Anfrage den gewünschten Inhalt generiert. Diese Parameter werden beispielsweise durch Formularfelder an den Server übermittelt. Das Generieren des gewünschten Inhalts geschieht durch Auslesen von **Dateien** oder durch das Abfragen von Daten aus einer **Datenbank**. Dieser Inhalt gelangt abermals über den Webserver in Form des so genannten **HTTP-Response** (Antwort) zum Client. Der gewünschte Inhalt wird im Browser des Benutzers angezeigt. Dies geschieht mit dem von der Webanwendung generierten HTML-Quellcode.

Ein Beispiel dafür ist die Suchfunktion der Online Enzyklopädie Wikipedia:



Abb 2: Suchseite bei Wikipedia [Quelle: www.wikipedia.de]

Dort kann der Benutzer einen beliebigen Suchbegriff eingeben. Nach dem Abschicken der Suchanfrage wird diese Anfrage über des Netzwerk/Internet an den Anwendungsserver weitergeleitet, worauf dieser eine Datenbankabfrage startet um dem Benutzer den gewünschten Eintrag zu liefern. Dieser wird dann auf dem Bildschirm des Clients dargestellt.

Nach Eingabe des Begriffs „Internet“ erhält der Benutzer folgende Anzeige:



Abb. 3: Bildschirm bei Suchbegriff Internet [Quelle: www.wikipedia.de]

Dies ist ein Beispiel, welches keinen kritischen Prozess enthält, da keine persönlichen Daten oder Anmeldeinformationen über das Netz versendet werden. Die grundlegende Arbeitsweise eines solchen Systems wird damit aufgezeigt.

Kritischen Prozesse werfen einige sehr wichtige Sicherheitsfragen auf, die im Verlaufe der Entstehung einer Webanwendung berücksichtigt werden müssen. Kritische Abläufe sind z.B. bei Bestellungen im Internet die Angabe von Kreditkarten- oder Kontonummern.

2.2 Vor- und Nachteile von Webanwendungen [2]

Vorteile:

- Im Gegensatz zu normaler Software ist hier keine Installation notwendig. Es wird lediglich ein Browser benötigt. Ein solcher Browser wird mit fast jedem Betriebssystem mitgeliefert.
- Dies macht die Anwendung plattformunabhängig und kann von beinahe jedem netzwerkfähigen Betriebssystem verwendet werden.

- Änderungen an der Logik der Software sind nur an zentraler Stelle auf dem Anwendungsserver notwendig.
- Viele dieser Anwendungen können auch von anderen Endgeräten, wie z.B. internetfähige PDAs, verwendet werden.

Nachteile:

- Eine ständige Netzwerkverbindung mit Hilfe eines Netzwerkprotokolls ist notwendig, was zu **Sicherheitsrisiken** führen kann. Da es viele Benutzer gibt, muss jedem Client eine ID zu dessen Identifikation zugewiesen werden.
- Daten müssen übers Internet transferiert werden, was zu weiteren Risiken führt.
- Die Geschwindigkeit der Netzwerkverbindung muss der Anwendung angemessen sein.
- Trotz identischem HTML Code, kann die Anzeige in verschiedenen Browsern variieren, was zu Schwierigkeiten in der Anpassung für verschiedene Browser führen kann.

2.3 Sicherheitsrisiken und Angriffe im Allgemeinen[10]

Anwendungen dieser Art sind auch für nicht berechtigte Benutzer des Internets erreichbar. Diese besitzen zwar keine Anmeldedaten (Benutzername, Passwort) und können sich nicht am System anmelden, jedoch gibt es verschiedene Gefahren denen das System und ihre Benutzer ausgesetzt sind. Durch nicht berechtigtes Eindringen in das System oder durch Verfälschung der Daten beim Transferieren, kann dem Benutzer oder dem Unternehmen selbst großer Schaden entstehen. Die Zielsetzung und Motivation von Benutzern mit böswilligen Absichten kann stark variieren. Ziel könnte es sein, ein Absturz des Systems zu provozieren, damit die Anwendung für einen unbestimmten Zeitraum nicht verwendet werden kann. Ein solcher Angriff würde auf der technischen Ebene erfolgen. Eine andere Intention könnte es beispielsweise sein, über verschiedene Wege in das System einzudringen um Daten auszuspionieren oder diese im schlimmsten Fall sogar zu verfälschen. Die Angreifer können sich im Unternehmen selbst befinden oder aber auch in konkurrierenden Firmen. Aufgrund dieser Vielfältigkeit von möglichen böswilligen Angriffen auf das System sollten Sicherheitsfragen auf verschiedene Ebenen verteilt werden. So existieren auf technischer Ebene andere Sicherheitsrisiken als auf inhaltlicher oder logischer Ebene.

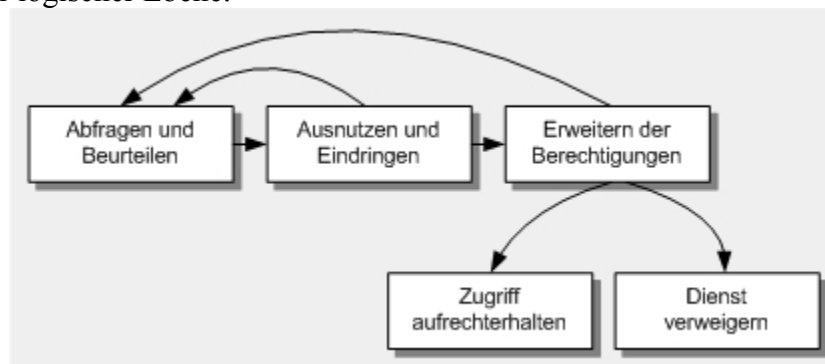


Abb 2: Anatomie eines Angriffs auf eine Webanwendung[9]

Abfragen und Beurteilen:

Der Erste Schritt eines Angreifers ist das Abfragen des potenziellen Ziels, um die Merkmale des Systems zu beurteilen. Diese Merkmale können die unterstützten Dienste und Protokolle zusammen mit potenziellen Schwachstellen enthalten. Diese Phase dient zum Planen des Angriffs.

Ausnutzen und Eindringen:

Nach dem Überwachen des potenziellen Ziels besteht der nächste Schritt im Ausnutzen und Eindringen. Wenn Netzwerk und Host(5) gut geschützt sind, ist die Anwendung das nächste Ziel des Angriffs.

Für einen Angreifer ist der einfachste Weg in eine Anwendung derselbe Eingang, den auch berechtigte Benutzer verwenden - zum Beispiel durch die Anmeldungsseite einer Anwendung oder eine Seite, die keine Authentifizierung erfordert.

Erweitern und berechtigen:

Nachdem Angreifer die Anwendung oder das Netzwerk offen gelegt haben, versucht er umgehend, die Berechtigungen zu erweitern. Speziell wird dabei nach administrativen(6) Berechtigungen gesucht, um weitreichender Rechte zu erlangen.

Zugriff aufrecht erhalten:

Nun versucht der Angreifer den Zugriff aufrecht zu erhalten und sich den Zugang für die Zukunft zu erleichtern. Hierzu können z.B. unzureichend geschützte Konten dienen. Zusätzlich versucht er seine Spuren zu verwischen. Dabei ist der Hauptangriffspunkt die Protokollierung des Systems. Protokolldateien(6) sollten geschützt und regelmäßig geprüft werden. Die Überprüfung von Protokolldateien kann oft frühe Anzeichen von versuchten Einbrüchen aufdecken, bevor der Angreifer erheblichen Schaden anrichten kann.

Dienst verweigern:

Gelingt es dem Angreifer nicht sich Zugriff zum System zu verschaffen, wenden dieser oft Angriffe an, die die regulären Benutzer davon abhält die Anwendung zu benutzen. Andere Angreifer verfolgen von Beginn an dieses Ziel. Hier hat der Angreifer die Möglichkeit dies über eine gewollte Überlastung der Warteschlange des Servers zu erreichen.

3. Ebenenmodell

3.1 Aufbau[1],[7]

Wie schon erwähnt, gibt es zur besseren Zuordnung der Zuständigkeiten im Punkt der Sicherheit einer Webanwendung ein 5 bzw. 6 Ebenenmodell, welches die jeweiligen Bereiche einer Organisation den Aufgaben in diesem Sektor zuordnet. Die Aufgaben sind einer verantwortlichen

Einheit (z.B. Abteilung) zugeordnet. Hieraus resultieren auch die erforderlichen Fachkenntnisse für die jeweiligen Ebenen.

Dazu gibt es drei Phasen der Entwicklung, denen die einzelnen Ebenen zugehörig sind:

Plan: Dies sind die Probleme und Risiken die schon in der Planungsphase berücksichtigt werden müssen, da Sie im späteren Verlauf schwer nachträglich zu beheben sind.

Build: Entwicklung, das heißt hier werden Aspekte betrachtet, die während der Programmierung der Anwendung Beachtung finden müssen.

Run: Im laufenden Betrieb muss die ständige Verfügbarkeit gewährleistet werden.

Übersicht:

	<i>Ebene</i>	<i>Inhalt</i>	<i>Zuordnung</i>
6	Vorschriften und Bestimmungen	Einhaltung gesetzlicher Regelungen und unternehmensspezifischer Vorgaben	<i>Phase: Planung</i> <i>Fachkenntnisse:</i> interne Abläufe <i>Verantwortung:</i> Zentrale Organisation
5	Semantik	Schutz vor Täuschung und Betrug	<i>Phase: Planung</i> <i>Fachkenntnisse:</i> interne Abläufe <i>Verantwortung:</i> Zentrale Organisation
4	Logik	Absicherung von Prozessen und Workflows als Ganzes	<i>Phase: Planung</i> <i>Fachk.:</i> Softwareentwicklung <i>Verantwortung:</i> Kunde, Entwickler
3	Implementierung	Vermeiden von Programmierfehlern, die zu Schwachstellen führen	<i>Phase: Entwicklung</i> <i>Fachk.:</i> Softwareentwicklung <i>Verantwortung:</i> Entwickler
2	Technologie	Richtige Wahl und sicherer Einsatz von Technologie	<i>Phase: Entwicklung</i> <i>Fachkenntnisse:</i> IT-Sicherheit <i>Verantwortung:</i> Betrieb, Entwickler
1	System	Absicherung der auf der Systemplattform eingesetzten Software	<i>Phase: Entwicklung</i> <i>Fachkenntnisse:</i> IT-Sicherheit <i>Verantwortung:</i> Betrieb, Entwickler
0 (*)	<i>Netzwerk & Host</i>	<i>Absicherung von Host und Netzwerk</i>	<i>Phase: laufender Betrieb</i> <i>Fachkenntnisse:</i> <i>Netzwerk, Administration</i> <i>Verantwortung:</i> <i>Betrieb</i>

Tabelle 1: Ebenenmodell

(*) nicht direkt der Webanwendung zugeordnet

Ebene 0 – Netzwerk und Host

Nicht direkt der Anwendung zugeordnet, eher ein allgemeiner Sicherheitsaspekt für öffentliche Server im Web. Auf dieser Ebene Sicherheit zu gewährleisten ist zwingend notwendig, da dies die Grundlage für alle weiteren Maßnahmen der darüber liegenden Schichten ist. Ständige Erreichbarkeit und eine schnelle Verarbeitung der Anfragen ist hier zu gewährleisten.

Ebene 1 – Systemebene

Auf dieser Ebene werden die verwendeten Programme auf Systemebene des Anwendungsservers betrachtet. Diese Programme sind diejenigen, die für die zu implementierenden Funktionalitäten notwendig sind. In diese Kategorie sind u.a. einzuordnen, der Webserver oder auch Datenbanken und andere Applikationen die sich auf Systemebene befinden auf die die Webanwendung zurückgreift.

Ebene 2 -Technologie

Hierbei handelt es sich um den Einsatz der richtigen Technologie und deren korrekter Anwendung. Dabei ist zu unterscheiden, ob die falsche Technik als solche eingesetzt wird oder die richtige Technologie verwendet wird aber diese in einer falschen Weise zum Einsatz kommt. Wird zum Beispiel eine Verschlüsselung verwendet, sollte diese auch die größtmögliche Länge für die Schlüssel verwenden, sonst werden die Möglichkeiten dieser Technologie nicht im gesamten Maße ausgenutzt.

Ebene 3 – Implementierung

Diese Ebene richtet sich vor allem an die Entwickler der Software. An dieser Stelle werden Fehler in der Programmierung verhindert. Ein Problem bei diesem Vorhaben stellt die Tatsache dar, das dieser Teil der Entwicklung häufig aus Zeitmangel und Kostengründen vernachlässigt wird. Dazu zählen z.B. auch Testphasen um eventuelle Programmierfehler aufzuspüren und gegebenenfalls zu eliminieren. Viele Anwendungen werden veröffentlicht, obwohl sie nur ungenügende Tests durchlaufen haben.

Ebene 4 – Logik

Hierbei handelt es sich um Logik innerhalb des Systems, welche auch Interaktionen mit dem Benutzer beinhaltet. Beispiel dafür ist die Behandlung fehlerhafter Eingaben wie z.B. beim Anmeldevorgang einer Anwendung. Ist die Verarbeitung dieser Eingaben nicht korrekt, wird es Dritten ermöglicht, Angriffe auf das System durchzuführen, welcher die Leistung eines Systems nachhaltig beeinflussen kann. Dies könnte zum Beispiel eine Denial of Service(7) Attacke sein, welche zur Überlastung des Systems und dem angebotenen Dienst zur Folge haben kann. Allgemein ist dies die Ebene, in der es um die Absicherung von Prozessen geht, wie das Registrieren neuer Benutzer oder dem Benutzer einen sicheren Login zu garantieren.

Ebene 5 – Semantik[5]

In diesem Bereich geht es um die Integrität der Interaktionen mit dem Benutzer. So ist es hier besonders wichtig zu verhindern dass Dritte die Anwendung missbrauchen um andere Benutzer zu täuschen. Dies sollte in den meisten Fällen nicht nur auf den Kontext der Anwendung reduziert werden. Aus dem Missbrauch einer Webanwendung können auch weitreichender Schäden entstehen, die die eigentliche Webanwendung nicht betreffen.

Ebene 6 – Vorschriften und Bestimmungen

In dieser Ebene kommen interne Vorschriften, Gesetzgebung und Regelungen von außen zum Tragen. Hierbei steht nicht mehr der technische Sicherheitsaspekt im Vordergrund sondern das Verhindern eventueller Schäden durch die Nichteinhaltung dieser Regelungen und Vorschriften. Aus diesem Grund werden Teile dieser Ebene im Weiteren nicht mehr genauer beleuchtet. Abhängig vom Teilgebiet der Anwendung können die Anforderungen in diesem Bereich stark variieren.

3.2 Bereiche der Sicherheit

Sicherheit kann in folgende Elemente aufgeteilt werden. Jedes Element kann Aspekte mehrerer oben genannter Ebenen enthalten.

1. Authentifizierung:

Dies ist das Identifizieren der Clients(8) und der Benutzer. Es muss dokumentiert werden, wer die Anwendung verwendet hat und zu welchem Zweck. Dadurch ist das Zurückverfolgen illegaler oder auch der regulärer Zugriffe möglich.

2. Autorisierung:

Es stellt sich die Frage was darf der jeweilige Benutzer oder Client. Dabei kann es sich um Rechte auf Dateien, Tabellen und anderen Ressourcen handeln. Auf Systemebene zählen noch Konfigurationsdateien und Registrierungsschlüssel dazu. Außer Ressourcen werden auch noch Operationen verwaltet, die der jeweiligen Anwendung zugeordnet sind, wie z.B. eine Überweisung beim Internet-Banking.

3. Überwachung:

Hier ist das Stichwort Nachweisbarkeit, um einem Kunden einen bestimmten, von ihm abgeschlossenen Vorgang, nachweisen zu können. Wenn ein Benutzer eines Webshops bestreitet, ein Produkt in einer gewissen Menge bestellt zu haben, muss der Betreiber dieses Systems dem Kunden nachweisen können, dass dieser die Bestellung in dieser Form vorgenommen hat.

4. Vertraulichkeit:

Datenschutz ist hier das Schlüsselwort. Es ist sicherzustellen, dass Daten vertraulich behandelt werden und nicht in die Hände Dritter gelangen, die nicht autorisiert sind auf diese Daten zuzugreifen.

5. *Integrität:*

Ähnlich dem Stichpunkt Vertraulichkeit. Daten sind vor ungewollten oder böswilligen Änderungen zu schützen. So dürfen nach einem Transfer Daten keine Änderungen gegenüber den gesendeten Daten aufweisen. Beim Transfer über Netzwerke muss dies sichergestellt sein. Hierzu dienen Verschlüsselung und Zugriffssteuerung um die Daten vor illegalem Zugriff zu schützen.

6. *Verfügbarkeit:*

Das Ziel vieler Angriffe auf Anwendungen im Netz ist es diese zum Absturz zu bringen, so dass auch für die regulären Benutzer kein Zugriff mehr möglich ist. Dies sollte verhindert werden, um die Verfügbarkeit des Systems in hohem Maße zu gewährleisten.

4. Beispiele für Gefahren und Gegenmaßnahmen

Folgendes Schema hat sich als sehr hilfreich für das Verständnis von Schwachstellen in Webanwendungen erwiesen – bei der Darstellung gegenüber Fachfremden ebenso wie bei der Schulung von Softwareentwicklern und Sicherheitsverantwortlichen. So liegt es auch der BSI-Studie[1] zugrunde. In der Studie sind Schwachstellen und zugehörige Schutzmaßnahmen den Ebenen zugeordnet.

	Ebene	Schwachstellen
6	Vorschriften und Bestimmungen	Fehlende Belehrungen zum Datenschutz. Nichteinhalten von Bestimmungen des KontraG.(9) Preisgabe vertraulicher Informationen.
5	Semantik	Gebrauch von Popups u.ä. erleichtern Phishing(10)-Angriffe. Keine Absicherung für den Fall der Fälschung der Website. Diebstahl der Anmeldeinformationen.
4	Logik	Verwendung unsicherer Email in einem ansonsten gesicherten Workflow(11). Angreifbarkeit des Passworts durch nachlässig gestaltete "Passwort vergessen"-Funktion. Die Verwendung sicherer Passworte wird nicht erzwungen.
3	Implementierung	Programmierfehler. SQL-Injection.(12). Sitzungsübernahme. Diebstahl der Anmeldeinformationen
2	Technologie	Unverschlüsselte Übertragung sensibler Daten. Authentisierungsverfahren, die nicht dem angemessen sind.
1	Netzwerk / System	Fehler in der Konfiguration des Webservers. Mangelnder Zugriffsschutz in der Datenbank. Mangelnde Protokollierung Denial of Service Angriffe (DoS)

Tabelle 2: Schwachstellen auf den Ebenen[1]

Hier einige Beispiele für Gefahren und deren bekannten Gegenmaßnahmen:

4.1 Diebstahl der Anmeldeinformationen

Ziel ist hier das zu Eigen machen fremder Anmelddaten um diese zur eigenen Anmeldung am System zu nutzen. Der Verlauf eines Browsers und sein Cache(13) speichern außerdem Benutzeranmeldeinformationen für den zukünftigen Gebrauch. Wenn jemand anderes als der Benutzer, der sich zuvor angemeldet hat, Zugang zu dessen Computer hat, und dieselbe Seite aufgerufen wird, ist die gespeicherte Anmeldung verfügbar.

Mit diesen Gegenmaßnahmen beugen Sie einem Diebstahl der Anmeldeinformationen vor:

- Verwenden und erzwingen von sicheren Passwörter für die Kennungen.
- Erzwingen von Kontosperrungen für Endbenutzerkonten nach einer bestimmten Anzahl von Anmeldeversuchen.
- Es sollte eine Funktion eingesetzt werden, mit der der Benutzer festlegen kann, dass Anmeldeinformationen im Cache des Webbrowsers nicht gespeichert werden. Diese Einstellung sollte als Standardrichtlinie, um eine Anmeldung mithilfe des Caches zu verhindern, erzwungen werden.

Eine weitere Möglichkeit des Identitätsdiebstahls ist das Phishing, dabei wird dem User eine gefälschte Webseite präsentiert, um dadurch seine Anmeldedaten der Originalwebseite zu erhalten. Um ein Missbrauch dieser Information zu verhindern muss der Benutzer umgehend seine Daten auf der Webseite ändern. Identitätsdiebstahl ist eine der am stärksten zunehmenden Kriminalitätsformen in hochtechnisierten Ländern. Bei der US-amerikanischen Handelsaufsicht FTC gingen beispielsweise im Jahr 2002 insgesamt 168.000 Anzeigen sowie 380.000 Beschwerden wegen Identitätsdiebstahls ein.

4.2 Offenlegung vertraulicher Daten

Die Offenlegung vertraulicher Daten kann auftreten, wenn unautorisierte Benutzer Zugriff auf diese Daten haben. Vertrauliche Daten beinhalten anwendungsspezifische Daten wie Kreditkartennummern, Informationen über Mitarbeiter, finanzielle Aufzeichnungen. Um eine Offenlegung vertraulicher Daten zu verhindern, sollten diese in permanenten Speichern wie Datenbanken und Konfigurationsdateien und während der Übertragung im Netzwerk gesichert werden. Nur authentifizierten und autorisierten Benutzer darf der Zugriff auf die ihnen bestimmten Daten gewährt werden. Der Zugriff auf Konfigurationsdaten der Systemebene sollte Administratoren vorbehalten bleiben.[10]

- Es ist eine Überprüfung der Rollen notwendig, bevor der Zugriff auf Vorgänge freigegeben wird, die sensible Daten preisgeben könnten.
- Verwenden einer Standardverschlüsselung, um vertrauliche Daten in Konfigurationsdateien und Datenbanken zu speichern

4.3 Abhören des Netzwerks

HTTP-Daten für Webanwendungen werden durch Netzwerke in Klartext übermittelt und sind Ziel von Angriffen die diese Daten abhören sollen, bei denen Angreifer Netzwerküberwachungssoftware verwenden, um vertrauliche Daten aufzuzeichnen und eventuell zu modifizieren. [10]

Gegenmaßnahmen um dies vorzubeugen:

- Daten müssen verschlüsselt werden.
- Zum Verschlüsseln der Daten kann z.B. SSL(14) verwendet werden.

4.4 Sitzungsübernahmen

Jeder Benutzer erhält nach seiner Anmeldung am System eine SitzungsId(15), welche in eindeutig identifiziert während er sich im System aufhält. Eine Sitzungsübernahme erfolgt, wenn ein Angreifer eine Netzwerküberwachungssoftware verwendet, um einen Cookie(16) abzufangen, der dem Benutzer zugeordnet ist. Mit dem abgefangenen Cookie kann der Angreifer die Sitzung des bereits angemeldeten Benutzers vortäuschen und sich Zugriff auf die Anwendung verschaffen. Der Angreifer hat dabei dieselben Berechtigungen wie der regulär am System angemeldete Benutzer. [10]

- Verwendung von SSL, um eine sichere Kommunikation zu garantieren und dies sollte am besten nur über eine HTTPS(17) Verbindung geschehen.
- Wenn eine Sitzung abgelaufen ist, wird nach der nächsten Operation des Benutzers im eine neue Anmeldung ermöglicht.
- Demzufolge sollte die Gültigkeit der Sitzungs-Id begrenzt werden, dadurch kann die Zeit die dem Angreifer bleibt verkürzt werden.

4.5 Denial-of-Service (DoS)

Bei einem Dienstverweigerungsangriff wird legitimen Benutzern der Zugriff auf einen Server oder Dienst verweigert. Der SYN-Flooding-Angriff(18) ist ein gutes Beispiel für einen solchen Angriff auf Netzwerkebene. Er ist leicht einzuleiten und schwer zu verfolgen. Ziel des Angriffs ist es, den Server mit Anfragen zu überlasten. Der Angriff zielt auf eine Schwachstelle der TCP/IP(19)-Verbindungsherstellung aus und überflutet die Verbindungswarteschlange des Servers.[10],[3]

Zugehörige Gegenmaßnahmen:

- TCP/IP-Stack absichern, indem die geeigneten Registrierungseinstellungen angewandt wird. Dadurch vergrößert sich die Verbindungswarteschlange, zugleich verkürzt sich der Zeitraum der Verbindungsherstellung und es können dynamische Protokollierungsmechanismen greifen, die sicher stellen, dass die Warteschlange niemals voll aus- bzw. überlastet ist.

- Es kann ein Netzwerksystem zur Erkennung von Eindringversuchen (IDS - Intrusion Detection System), mit dem SYN-Angriffe automatisch aufgespürt und verhindert werden können.

4.6 Protokollierung

Alle durchgeführten Transaktionen müssen vom System aufgezeichnet und verfolgt werden. Dies verhindert, dass ein Benutzer abstreiten kann, eine bestimmte Transaktion durchgeführt zu haben.

Schutz vor diesem Sachverhalt:

- Überprüfen und protokollieren der Aktivitäten auf dem Webserver, Datenbankserver und auch auf dem Anwendungsserver
- Protokollieren von Schlüsselereignissen, wie Transaktionen, Anmelde- und Abmeldevorgänge.
- Es sollten keine gemeinsamen Benutzerkonten verwendet werden., da die Ursprungsquelle nicht bestimmt werden kann.

Auch Vorgänge wie An- und Abmeldungen sowie Neuanmeldungen sollten dokumentiert werden.[10]

Leider können in diesem Text nicht alle Schwachstellen und Gegenmaßnahmen ausführlich behandelt werden. Für eine genauere Betrachtung verweise ich auf die oben genannte BSI-Studie[1], dort sind detailliertere und ausführlichere Darstellungen zu finden. Es existieren noch weit mehr Gefahren und Angriffe und die jeweiligen Gegenmaßnahmen. Dieser Text dient dem Zweck, dass Zielpublikum auf diese Thematik aufmerksam zu machen und einige Grundlagen darzustellen.

5. Fazit

Für Betreiber von Webanwendungen ist es von immenser Wichtigkeit, dass diese sich den Gefahren bewusst sind, denen diese Anwendungen im Internet ausgesetzt sind. Diese Gefahren sind sehr Vielfältig und werden auf viele verschiedene Arten durchgeführt. Zudem sollte man die Vorgehensweise und die Ziele der bekannten Angriffe kennen. Den nur wer die Taktik seines Gegners kennt, kann eine wirkungsvolle Verteidigung aufbauen. Werden diese Angriffe nicht vom System blockiert oder werden Sachverhalte nicht sachgemäß dokumentiert, können diese zu erheblichen Schäden führen. Sicherheit ist ein sehr wichtiger Punkt in der Entwicklung einer solchen Anwendung und sollten schon in der Modellierung einer solchen Anwendung Beachtung finden. Wichtig ist hierbei eine Strukturierte Vorgehensweise z.B. mit dem im Text vorgestellten Ebenenmodell, um die Zuständigkeit der verschiedenen zu implementierenden Schutzmaßnahmen festzulegen. Zu einem späteren Zeitpunkt sind viele Maßnahmen nur schwer oder sogar gar nicht mehr zu realisieren. Was einen weiteren längeren Entwicklungsprozess in Anspruch nehmen würde, in dem die Anwendung weiterhin mit diesem Sicherheitsloch online zur Verfügung steht. Ständig werden von Hackern neue Angriffe und Hinterhalte entwickelt, so dass es von großer Bedeutung ist, hier immer auf dem neuesten Stand zu bleiben um seine Anwendungen bestmöglich zu schützen.

6. Glossar

- (1) Das World Wide Web Consortium, oder auch W3C, ist das Gremium zur Standardisierung des World Wide Web betreffender Techniken. Gründer und Vorsitzender des W3C ist Tim Berners-Lee, der auch als der Erfinder des World Wide Web bekannt ist. Es wurde 1994 gegründet.
- (2) Webbrowser oder Browser // (engl. für "Stöberer") sind Computerprogramme zum Betrachten verschiedener Arten von Dokumenten. Vorwiegend werden sie verwendet, um HTML-Seiten aus dem Internet anzuzeigen. Wegen der Metapher, dass man rasend schnell durch das Web braust und der Wortähnlichkeit zu Browser, wird manchmal ironisierend der Begriff Brauser verwendet
- (3) Bezeichnung für Server im WWW, die Webinhalte anbieten.
- (4) Ein Datenbankserver ist ein Computersystem auf dem Datenbanken, meist verbunden mit einem hohen Datenvolumen, verwaltet werden, mit dem Ziel Daten über viele Clients zu sammeln oder vielen Clients zur Verfügung zu stellen.
- (5) Allgemein Rechner oder Server auf dem in der Regel Dienste für Benutzer bereitgestellt werden. Der Host ist ein Rechner, zu dem man Verbindung aufnimmt. Das Gegenteil ist der Client. Einer oder mehrere dieser Clients greifen auf eine Anwendung zu, die auf einem anderen Rechner gehostet wird.
- (6) Administration ist die Verwaltung von Netzwerken. Deren Systemverwalter wird Administrator genannt. Der Administrator hat uneingeschränkte Zugriffsrechte und ist für die Verwaltung und Betreuung des Netzwerks zuständig.
- (7) Hacker-Angriff, bei dem ein PC/ Server so lange mit Anfragen bzw. Datenpaketen bombardiert wird, bis er überlastet seinen Dienst quittiert.
- (8) Client ein Computer oder Anwendung, die die Dienste eines Servers in Anspruch nimmt.
- (9) Das „Gesetz zur Kontrolle und Transparenz im Unternehmensbereich“, kurz KonTraG ist ein umfangreiches Artikelgesetz, das der Deutsche Bundestag am 5. März 1998 verabschiedete. Es trat am 1. Mai 1998 in Kraft (wenn auch einige Vorschriften erst später angewandt werden mussten bzw. durften).
- (10) Phishing ist eine Form des Trickbetruges mit Methoden des Social Engineerings. Es ist der Oberbegriff für illegale Versuche, weitgestreut Anwendern Zugangsdaten für sicherheitsrelevante Bereiche zu entlocken. Phishing ist eine Variante des Identitätsdiebstahls.
- (11) Ein Workflow ist eine Abstraktion eines Geschäftsprozesses, die vor allem auf den Fluss digitalisierter Dokumente bzw. Objekte gerichtet ist. Menschliche Aktivitäten bzw. Entscheidungen im Rahmen eines Geschäftsprozesses werden dabei weitgehend ausgeklammert bzw. auf Interaktionen mit Anwendungssystemen reduziert
- (12) SQL: Universelle Abfragesprache für Datenbanken.
SQL Injektion: SQL Injection bezeichnet das Ausnutzen einer Sicherheits-Lücke. Der Angreifer versucht SQL-Abfragen zu manipulieren. Hierzu werden über die Applikation, die den Zugriff auf die Datenbank bereitstellt, SQL Statements eingefügt.
- (13) Generell ein Zwischenspeicher, Lokales Verzeichnis, in dem der Web-Browser die heruntergeladenen Daten zwischenspeichert, um sich ggf. ein erneutes Laden vom Server zu sparen.

- (14) Secure Sockets Layers. Von Netscape entwickeltes Protokoll, um vom Browser gesendete Daten durch Verschlüsselung zu sichern.
- (15) Weist die effektive Verweildauer eines Besuchers innerhalb einer Web-Seite aus. Dient zur Identifikation, von welchem Client eine Anfrage kam.
- (16) Ein Cookie ist eine Information, die ein Web-Server bei einem Klientenprogramm ablegt. Damit lassen sich Zustände speichern, so dass ein Benutzer bei einem späteren Besuch seine gewohnte Umgebung vorfindet. Cookies haben üblicherweise ein "Verfallsdatum", nach denen sie gelöscht werden. Zur Sicherheit werden die Informationen eines Cookies nur an den Web-Server zurückgegeben, der den Cookie ursprünglich angelegt hat.
- (17) Hypertext transfer protocol secure (HTTPS) ist ein Netzwerkprotokoll, das eine gesicherte HTTP-Verbindung zwischen Rechnern ermöglicht.
- (18) Ein SYN-Flood ist eine Form von Denial of Service Attacken auf Computersysteme. Der Angriff verwendet den Verbindungsaufbau von TCP/IP, um einzelne Dienste oder ganze Computer aus dem Netzwerk unerreichbar zu machen.
- (19) Transmission Control Protocol/Internet Protocol. Standard-Kommunikationsprotokoll für alle mit dem Internet verbundenen Rechner.

7. Literatur

- [1] secureNet, Sicherheit von Webanwendungen, (im Auftrag vom Bundesamt für Sicherheit in der Informationstechnik)
Maßnahmenkatalog und Best Practices
<http://www.bsi.de/literat/studien/websec/WebSec.pdf>
Version 1 vom 1. August 2006
Zugriff: Dezember 2006
- [2] Wikipedia (Autor unbekannt), Webanwendungen
<http://de.wikipedia.org/wiki/Webanwendungen>
Zugriff: Januar 2007
- [3] Wikipedia (Autor unbekannt), Denial of Service
http://de.wikipedia.org/wiki/Denial_of_Service
Zugriff: November 2006
- [4] Wikipedia (Autor unbekannt), Session-Hijacking
http://de.wikipedia.org/wiki/Session_Hijacking
Zugriff: Dezember 2006
- [5] secureNet, Die semantische Ebene der Sicherheit bei Webanwendungen.
<http://www.securenet.de>
Version vom: Whitepaper Mai 03
Zugriff: Dezember 2006
- [6] Dr. Micheal David, Angriffe auf Sicherheitsinfrastrukturen der IT.
Version vom: unbekannt
[Zugriff: Dezember 2006](http://www.securenet.de/papers/Klassifizierungsschema_Web_Application_Security.pdf)
- [7] Thomas Schreiber, Ein Klassifizierungsschema zur Sicherheit von Webanwendungen.
http://www.securenet.de/papers/Klassifizierungsschema_Web_Application_Security.pdf
Version: It-Sicherheitskongress 2005

Zugriff: Januar 2007

[8] Abbildung 2:

<http://www.microsoft.com/germany/msdn/library/security/ErhoehenDerSicherheitVonWebanwendungen>

Zugriff: Januar 2007

[9] Einleitung - Erhöhen der Sicherheit von Webanwendungen

<http://www.microsoft.com/germany/msdn/library/security/ErhoehenDerSicherheitVonWebanwendungen>

Zugriff: Januar 2007