Software Design Patterns

Ausarbeitung über

Security Patterns

SS 2004

Dennis Völker [dv04@hdm-stuttgart.de]
Steffen Schurian [ss59@hdm-stuttgart.de]

Überblick

Sicherheit sollte eine Eigenschaft moderner, verteilter Anwendungen sein, jedoch ist ein adäquates Sicherheitsniveau noch lange nicht erreicht.
Die gleichen Fehler treten immer wieder auf, wie beispielsweise Buffer Overflows in Anwendungen oder Standardpasswörter in IT-Systemen.

Kodierungsfehler werden in modernen Entwicklungsumgebungen spätestens beim Kompilieren bemerkt und sind somit schnell verbessert.
Jedoch werden konzeptionelle Fehler nicht vom Compiler entdeckt und treten erst zur Laufzeit auf. Ihre Auswirkungen sind vorher nicht absehbar und können verheerende Folgen haben.
Ein Beispiel hierfür ist der obligatorische „Division durch 0"-Fehler.

Diese Fehler schleichen sich aber oft durch die menschliche Art Fehler zu beheben ein.
Wir lösen Fehler erst wenn sie auftreten ohne die Nebenwirkungen unserer Lösung zu betrachten. Problematisch ist auch die Einstellung Fehler bewusst in Kauf zu nehmen und erst zu beheben, wenn sie entdeckt werden.
Aus diesen Gründen können nur erfahrene Experten zuverlässig Sicherheitslücken aufdecken.
Damit auch Laien von deren Wissen profitieren können ist der Patternansatz aus der Softwareentwicklung auch in die IT-Security übertragen worden.

Grundlagen von Security Patterns

Kontext

Der Kontext eines Security Patterns wird am besten mit einem Beispiel beschrieben. Mit Hilfe einer Auflistung der Annahmen, in der das IT-System benutzt werden soll, können Eigenschaften der Umgebung beschrieben werden. Oftmals müssen IT-Systeme bestimmten Sicherheitsvorgaben entsprechen. Eine Beschreibung solcher Aussagen erleichtert daher die eindeutige Bestimmung der Schutzziele.

Problem

Bedrohungen bilden den Kern von Sicherheitsproblemen. Ein IT-System gilt grundsätzlich nur dann als sicher wenn Maßnahmen gegen alle bekannten Bedrohungen getroffen wurden.
Des Weiteren können auch Schutzziele für noch nicht bekannte Bedrohungen formuliert werden, deren Aufgabe darin besteht, zukünftigen Gefahren entgegen zu wirken.

Anforderungen

Verschiedene Anforderungen beeinflussen unmittelbar die Lösung. Dazu gehören außer den Sicherheitsanforderungen noch allgemein Anforderungen. Diese werden durch die Forderung nach Sicherheit beeinflusst, wie z.B. Benutzbarkeit oder Performanz.

Lösung

Abhängig von den jeweiligen Anforderungen müssen Maßnahmen gefunden werden, die das Problem im jeweiligen Kontext lösen und die Risiken auf ein Minimum reduzieren. Es sollte mindestens eine Maßnahme für jede Bedrohung geben. Außerdem sollten die Vor- und Nachteile der Anwendung eines Security Patterns diskutiert werden.

Allerdings sind durch die Einführung von Security Pattern Nebenwirkungen nicht ausgeschlossen. Es kann zu Nebeneffekte mit anderen Patterns kommen.

Ein gutes Beispiel in dem die Sicherheitsaspekte im Vorfeld nicht genügend betrachtet wurden sind Cookies.

Security Pattern basierte Lösung

Kontext

Serveranwendungen verwenden Cookies, um Informationen über User abzuspeichern und abzufragen. Ein Server verwendet den HTTP-Header, um ein Cookie dauerhaft bei dem User zu platzieren und so Informationen wie beispielsweise eine eindeutige Benutzerkennung zu hinterlegen. Fragt der User eine weitere Seite des Servers an, wird das Cookie automatisch mit der HTTP-Anfrage versendet.

Problem

Dienstanbieter können die Aktivitäten eines Users verfolgen und somit Profile erstellen.
Userdaten von verschiedenen Anbietern können zusammengeführt werden.
Mit Hilfe von HTML-basierten E-Mail-Nachrichten können Cookies ohne Wissen des Benutzers personalisiert werden.

Anforderungen

- **Anonymität:** Dienstanbietern und andere Benutzern soll es nicht möglich sein, die Identität eines Benutzers aus einer HTTP-Anfrage abzuleiten. Ein Dienst darf weiterhin nicht den richtigen Benutzernamen offen legen.
- **Unverknüpfbarkeit:** Ein Benutzer sollte mehrere HTTP-Anfragen stellen können, ohne dass diese miteinander in Verbindung gebracht werden.
- **Benutzbarkeit:** Viele Angebote sind nicht benutzbar, wenn Cookies nicht akzeptiert werden. Außerdem möchten Benutzer oft keine zusätzliche Software kaufen und installieren bzw. Änderungen in der Konfiguration vornehmen.

Lösung

Die Verwendung von Cookies muss auf Seiten des Benutzers eingeschränkt werden.
Es sollte möglich sein in einem Webbrowser die Verwendung von Cookies einzuschränken, indem der User selbst entscheiden kann ob er ein Cookie annehmen möchte. Cookies sollen auch periodisch gelöscht werden können, so dass die Erstellung von Profilen nicht mehr möglich ist.
Der Schutz der Privatsphäre kollidiere hier mit der Usability.

Vom Security Pattern zum Security Pattern System

Security Pattern existieren nicht unabhängig von einander. Jedes Pattern löst eine spezifische Aufgabe. In komplexen Systemen finden daher viele Security Pattern Anwendung. Zusammen bilden sie ein Pattern System.

Security Pattern Systems nach Yoder und Barcalow:

Die Folgende Abbildung (Abb. 1) gibt einen Überblick über das Security Pattern System nach Yoder und Barcalow.

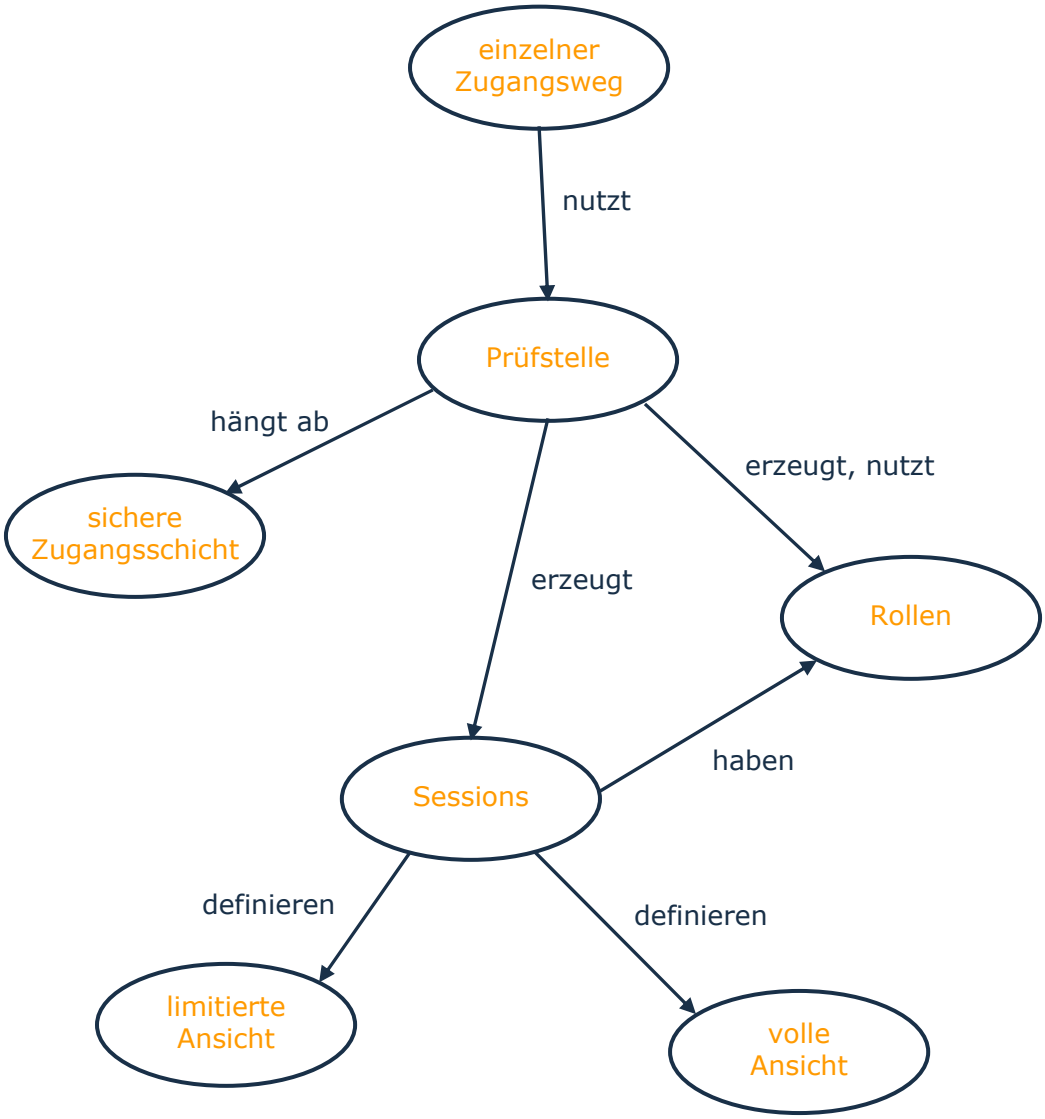

Abb. 1: Abhängigkeit zwischen Patterns im Security Pattern System (nach Yoder & Barcalow, 2002)

Einzelner Zugangsweg
Es ist schwierig eine Anwendung abzusichern, die über beliebige Zugangswege verfügt, deshalb wird der Zugang auf einen beschränkt.

Prüfstelle
Verschiedene User haben unterschiedliche Anforderungen an die Sicherheit. Um diese Anforderungen unabhängig von einer Anwendung umzusetzen wird die Kapselung eines entsprechenden Verfahrens das die Sicherheitsregeln überprüft vorgeschlagen.

Rollen
Verwaltung von individuellen Benutzerrechten ist ab einer gewissen Anzahl von Usern nicht mehr überschaubar bzw. verwaltbar. Hier wird es nötig Rechte auf Rollenbasis einzurichten. Einzelnen Usern werden nun Rollen zugewiesen. Somit nimmt der Verwaltungsaufwand ab da es weniger Rollen als User gibt.

Session
Verschiedene Module einer Anwendung benötigen Zugriff auf sicherheitsrelevante Daten, die global mit Hilfe eines Sitzungskonzeptes gespeichert werden. Somit ist von Außen kein Zugriff auf die Daten möglich. Die Gültigkeitsdauer ist auf die Lebenszeit der Sitzung beschränkt.

Fehlermeldungen
- **Volle Ansicht**
  User dürfen Funktionen nur ausführen wenn sie über nötige Rechte verfügen. Falls die versuchen Funktionen auszuführen für die sie keine Berechtigung haben werden entsprechende Fehlermeldungen angezeigt.

- **Limitierte Ansicht**
  User bekommen nur die Funktionen angezeigt, die sie mit ihren Rechten ausführen dürfen.

Sichere Zugangsschicht
Anwendungen können nur so sicher sein wie ihre Umgebung. Externe Sicherheitsmechanismen von Betriebssystem, Netzwerk oder Datenbanken sollen daher eine sichere Zugangsschicht bieten auf der Nachrichten ausgetauscht werden können.

Vorteile eines Pattern Systemes sind beispielsweise, dass aus der Anwendung eines Patterns zu erkennen ist welche weiteren Pattern noch berücksichtigt werden müssen. Ein weiterer Vorteil bei der Anwendung von Pattern Systemen ist, dass erkannt werden kann welche Auswirkungen ein Fehler in der Implementierung einer Funktion auf die Anwendung oder andere Funktionen haben kann.

Der Nachteil eines Security Pattern Systems ist die Komplexität der einzelnen Pattern. Es gibt weinig gemeinsame Klassifizierungseigenschaften die das Zusammenfügen einzelner Pattern erleichtern würden. Um dies zu ermöglichen müsste eine einheitliche Struktur zur Beschreibung von Security Pattern eingeführt werden. Weitere Pattern müssen noch gefunden und identifiziert werden.

Security Pattern ermöglichen es grundsätzlich auch dem Laien sichere Anwendungen entwickeln zu können.

Literatur

1. Security Patterns Community: Security Patterns Homepage. [http://www.securitypatterns.de] (2002)
2. Yoder, J., Barcalow, J.: Architectural Patterns for Enabling Application Security. 4th Conference on Patterns Languages of Programs (PLoP), Monticello/IL, 1997