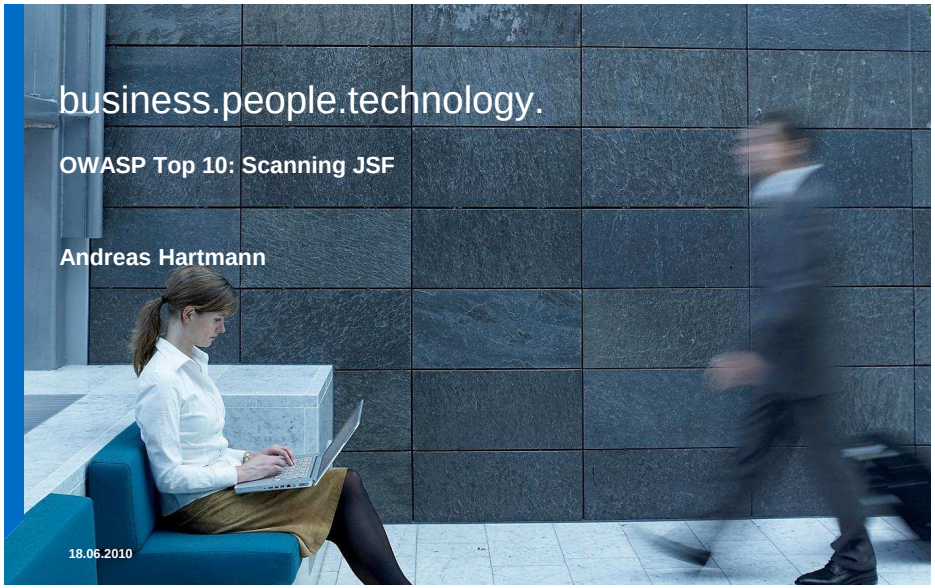
JAVAFORUM 2010 stuttgart am 1. Juli 2010

adesso business. people. technology.



Agenda

- Web Application Security Kick Start
- OWASP Top Ten meets JSF
- Application Security Komponente
- Application Security Startup

Agenda

- **Web Application Security Kick Start**
- OWASP Top Ten meets JSF
- Application Security Komponente
- Application Security Startup

Web Application Security Kick Start

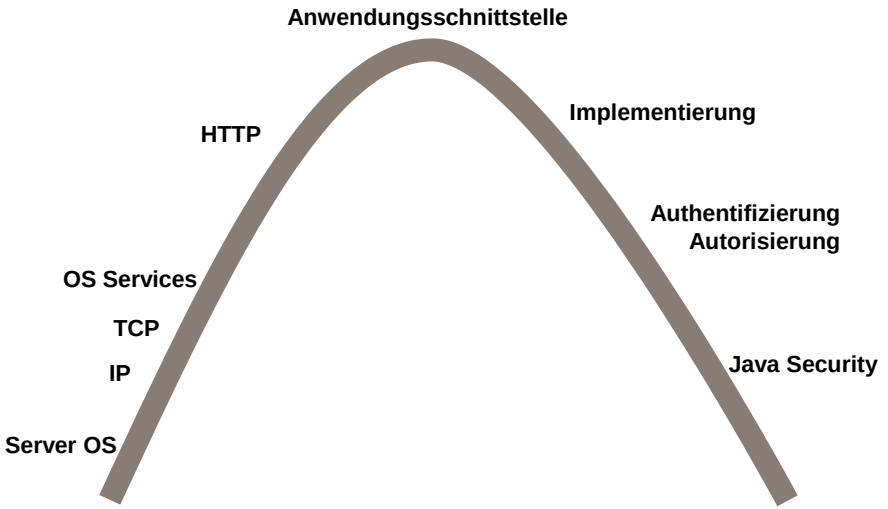
Was ist Application Security?

Application Security umfasst alle Maßnahmen im Lebenszyklus von Software, die geeignet sind, sicherheitskritische Fehler im Design, der Implementierung, dem Deployment und der Wartung von Software zu verhindern.

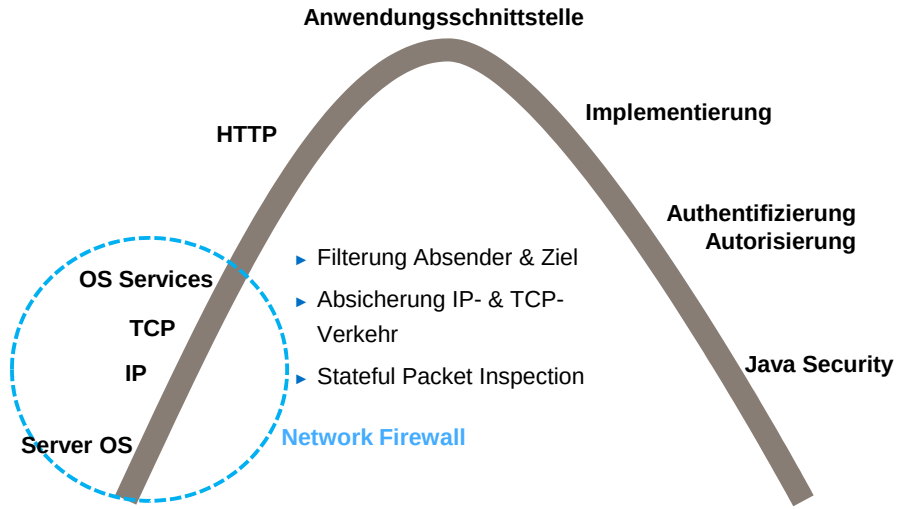
Schutzziele von Application Security

- Vertraulichkeit & Integrität
  - der Daten
  - der Kommunikation
- Authentizität der Kommunikationspartner
- Verfügbarkeit der Services

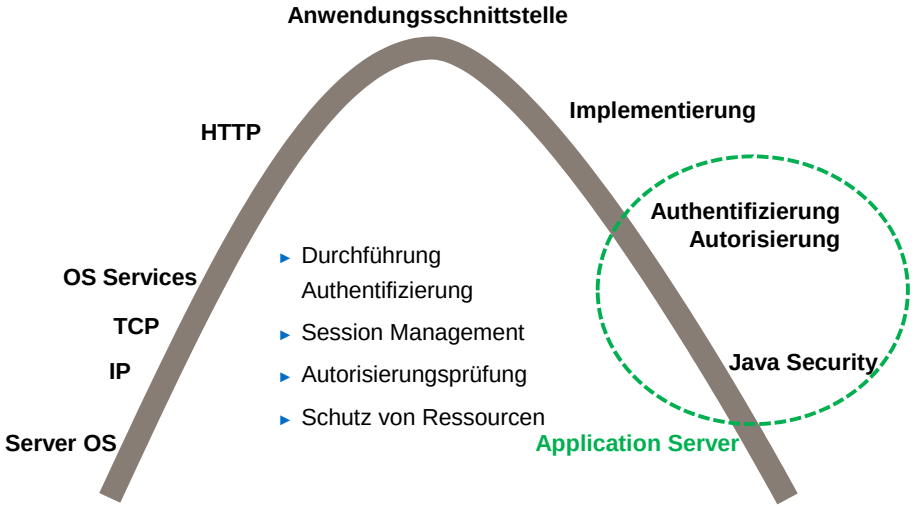
Web Application Security Kick Start



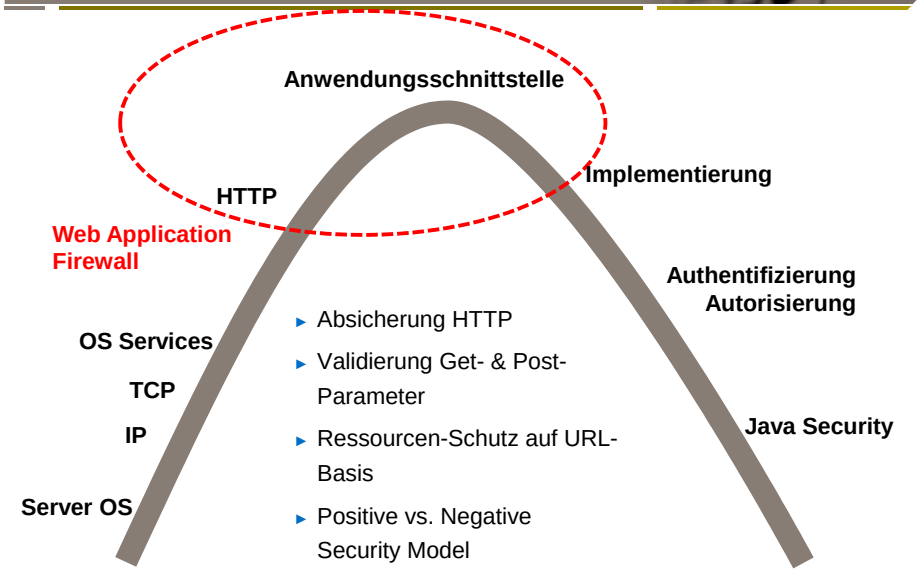
Web Application Security Kick Start



Web Application Security Kick Start



Web Application Security Kick Start



Web Application Security Kick Start

Rahmenbedingungen für den WAF-Einsatz

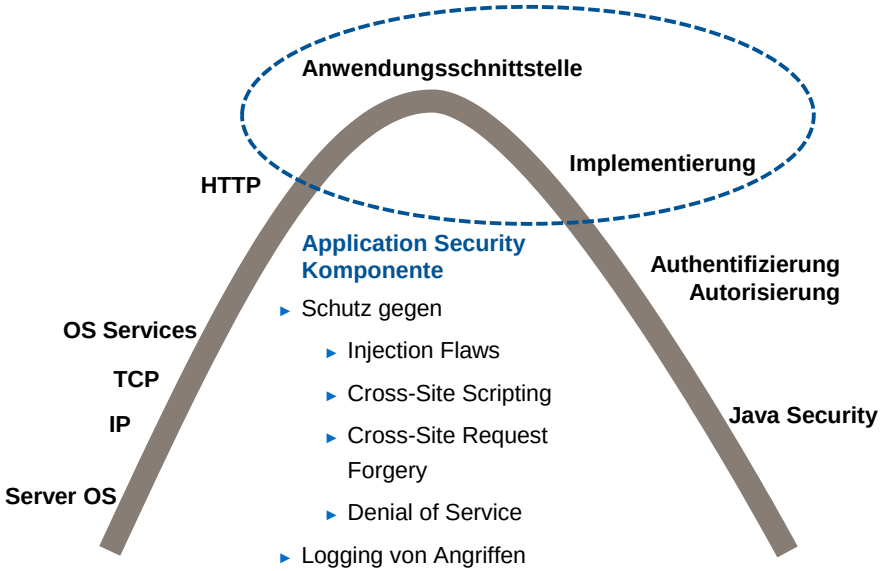
- Abstimmung auf Web Applikation
  - Start bereits während der Entwicklung
  - Kontinuierliche Anpassung an Anwendungsänderungen
- Angleichung der Release-Zyklen von WAF und Applikation(en)
- Testaufwand für jedes Update einplanen
- SSL-Terminierung vor der WAF notwendig
- Bei optimaler Abstimmung gute Erkennungsraten möglich

Web Application Security Kick Start

Probleme beim WAF-Einsatz

- Ansatz negatives Security Model
  - Abhängigkeit von häufigen Updates
  - relativ hohe Gefahr von Anwendungsproblemen nach Updates
- Ansatz positive Security Model
  - lange Lernphase notwendig
  - optimale Abstimmung auf Applikation notwendig
  - relativ hohe Gefahr von False Positives
- allgemeines Risiko einer weiteren Infrastrukturkomponente
- Expertenwissen für sinnvollen Einsatz notwendig

Web Application Security Kick Start

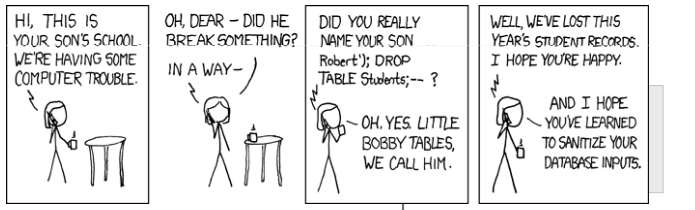


Agenda

A1 - Injection Flaws

Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.

- SQL
- Xpath
- etc.



Whitelist Validation & Escaping Special Characters & Prepared Statements zum Schutz gegen SQL-Injection, …

A2 – Cross Site Scripting (XSS)

XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.

- Session Hijacking
- Fernsteuerung des Browsers

Whitelist Validation & Escaping Special Characters & Output Encoding

A3 - Broken Authentication and Session Management

Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities.

- Logische Fehler bei der Authentisierung und Autorisierung, z.B. beim Logout die Session nicht invalidiert

Durchdachte Konzepte & richtige Konfiguration des Servers

A4 - Insecure Direct Object Reference

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.

- Zugriff auf Daten die für den Benutzer nicht erreichbar sein sollen

Bei der Entwicklung berücksichtigen und niemals darauf vertrauen, dass Referenzen korrekt sind

A5 - Cross Site Request Forgery (CSRF)

A CSRF attack forces a logged-on victim's browser to send a pre-authenticated request to a vulnerable web application, which then forces the victim's browser to perform a hostile action to the benefit of the attacker. CSRF can be as powerful as the web application that it attacks.

- Vertrauensbruch zwischen Browser und Server (Session Riding)

Einsatz eines dynamischen Tokens

A6 – Security Misconfiguration

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes ke

- Keine gehärtete Konfiguration der verwendeten Framwork, Server, etc.

Härten der Infrastruktur & Security Prozess zur Inbetriebnahme von Anwendungen

A7 - Insecure Cryptographic Storage

Web applications rarely use cryptographic functions properly to protect data and credentials. Attackers use weakly protected data to conduct identity theft and other crimes, such as credit card fraud.

- Verschlüsselte Ablage von sensitiven Daten

Einsatz sicherer Kryptographiemechanismen & Vermeidung proprietärer Lösungen

A8 - Failure to Restrict URL Access

Frequently, an application only protects sensitive functionality by preventing the display of links or URLs to unauthorized users. Attackers can use this weakness to access and perform unauthorized operations by accessing those URLs directly.

- Erraten von URLs

URL Pfadzugriff einschränken, nur das was tatsächlich notwendig ist

A9 – Insufficient Transport Layer Protection

Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications.

- Datentransport kann abgehört werden
- Cookies können gestohlen werden

Durchgängige Verwendung von SSL & Einsatz von gültigen Zertifikaten & Secure Flag für Cookies

A10 – Unvalidated Redirects and Forwards

Web applications frequently redirect and forward users to other pages and websites, and use untrusteddata to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

- Zugriff auf nicht autorisierte Seiten
- Umleitung des angemeldeten Benutzers auf eine Phishing Seite

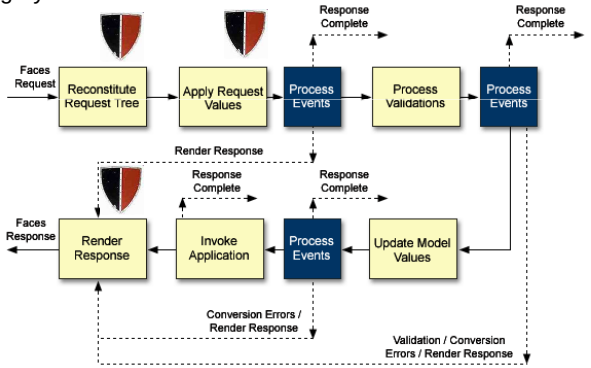
Nach Möglichkeit kein Redirect oder Forward im eigenen Code & KeineBenutzereingaben als Parameter

Agenda

Application Security Komponente

JSF Lifecycle meets Security

- Injection Flaws (z.B. SQL-Injection)
- Cross-Site Scripting
- Cross-Site Request Forgery
- Denial of Service
- Never-Ending Application Usage
- Logging



Application Security Komponente

Konfiguration

security-config.xml
- Actions
- Cross-Site Request Forgery
- Denial of Service
- Never Ending Application Usage

pages.xml
- Zuordnung der Seiten
- Zuordnung der GUI Elemente
- Zuordnung der Actions
- Zuordnung der Regulären Ausdrücke

regular-expressions.xml
- Reguläre Ausrücke
- White-/Blacklisting

Agenda

Application Security Startup

Ansatzpunkte zur Einführung von Application Security Maßnahmen
- Definition eines Security Prozesses
- Schaffung von Security Awareness
- Anpassung der Architekturrichtlinien
- Erweiterung der Programmierrichtlinien
  - Nutzung von Security Infrastrukturcode prüfen
  - Ein- und Ausgabevalidierung
  - Sicheres Errorhandling & Logging
- Durchführung von Security Reviews
  - interne & externe Security Audits einplanen
  - Betrachtung der zu erstellenden Anwendung hinsichtlich allgemeiner und spezieller Sicherheitsrisiken
  - Einsatz von Werkzeugen zur statischen Codeanalyse während der Entwicklung

adesso business. people. technology.

Vielen Dank für Ihre Aufmerksamkeit.

info@adesso.de / www.adesso.de