

Praktische Erfahrungen aus hunderten von Sicherheitsabnahmen

German OWASP Day 2014

Dr. Amir Alsbih – CISO Haufe Gruppe



Agenda

Hintergrund

Zertifizierungen

SaaS Service

Applikationsentwicklung

Summary



Hintergrund

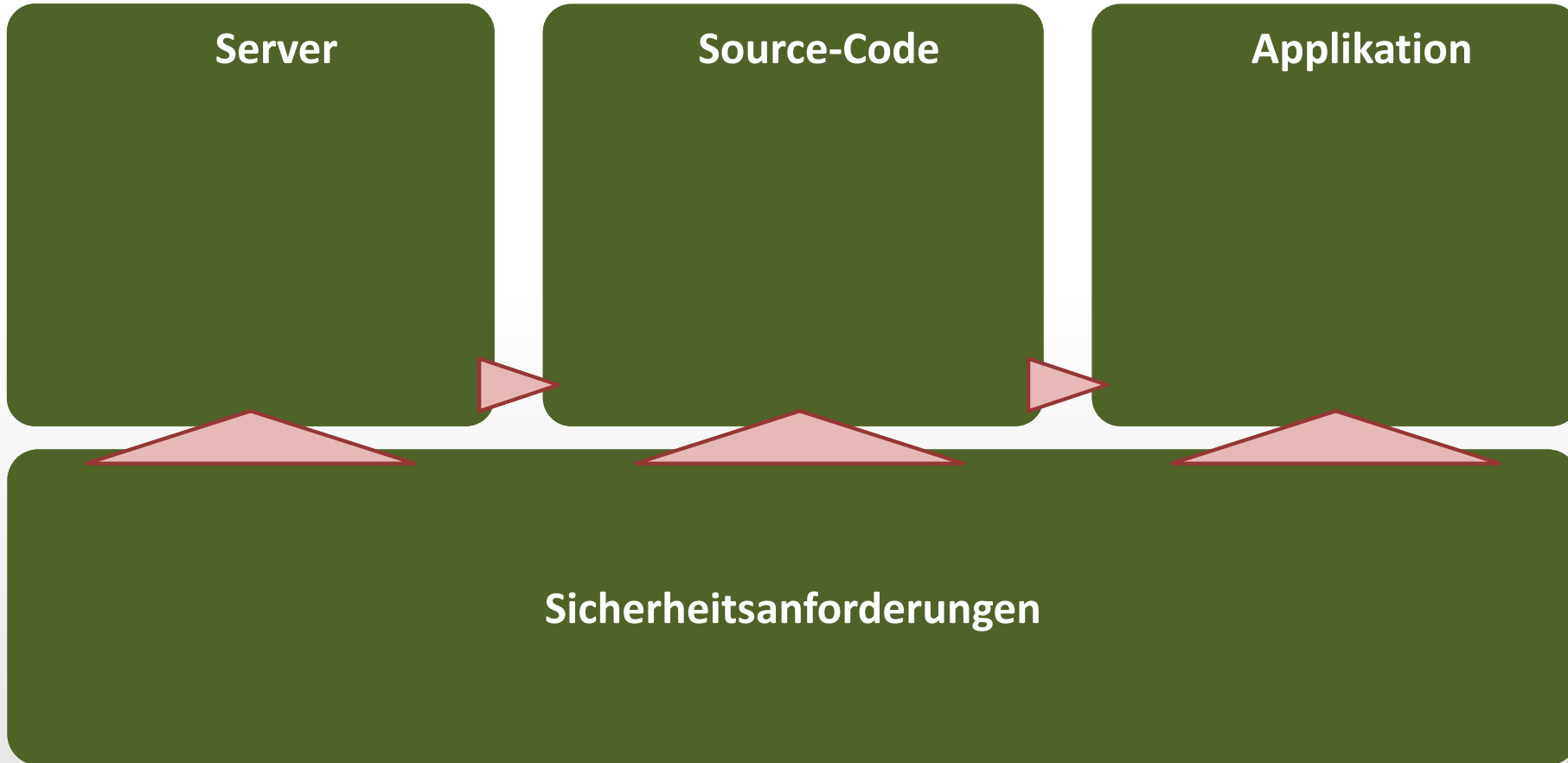
Zertifizierungen

SaaS Service

Applikationsentwicklung

Summary

Vor „Going Live“ die Verpflichtung Sicherheitsabnahmen durchzuführen



Unabhängige Verifikation der Sicherheits-Anforderungen



Hintergrund

Zertifizierungen

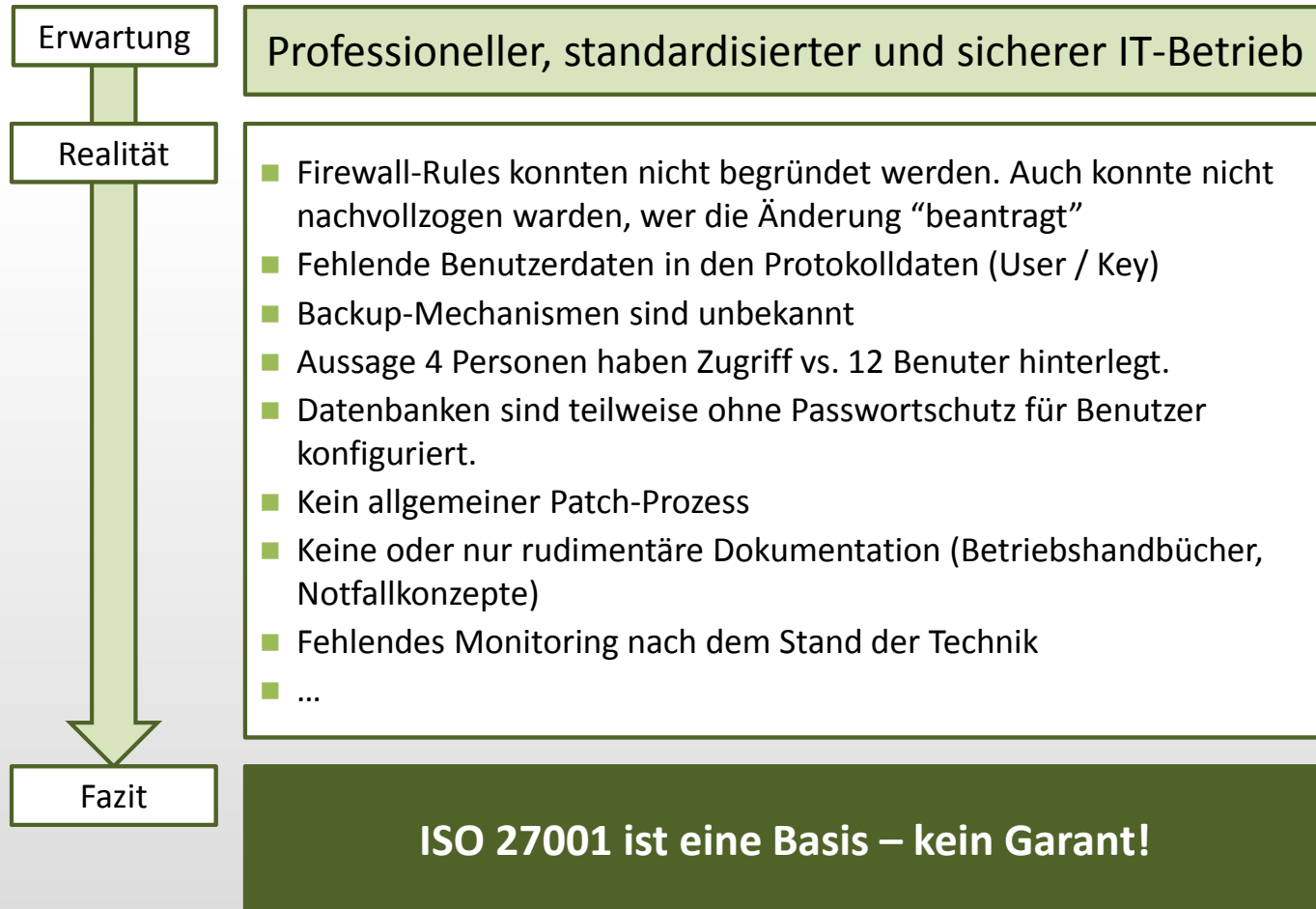
SaaS Service

Applikationsentwicklung

Summary

Der Betrieb nach ISO 27001 ist kein Garant für “Sicherheit”

Vollständiges Outsourcing von Applikationen



Hintergrund

Zertifizierungen



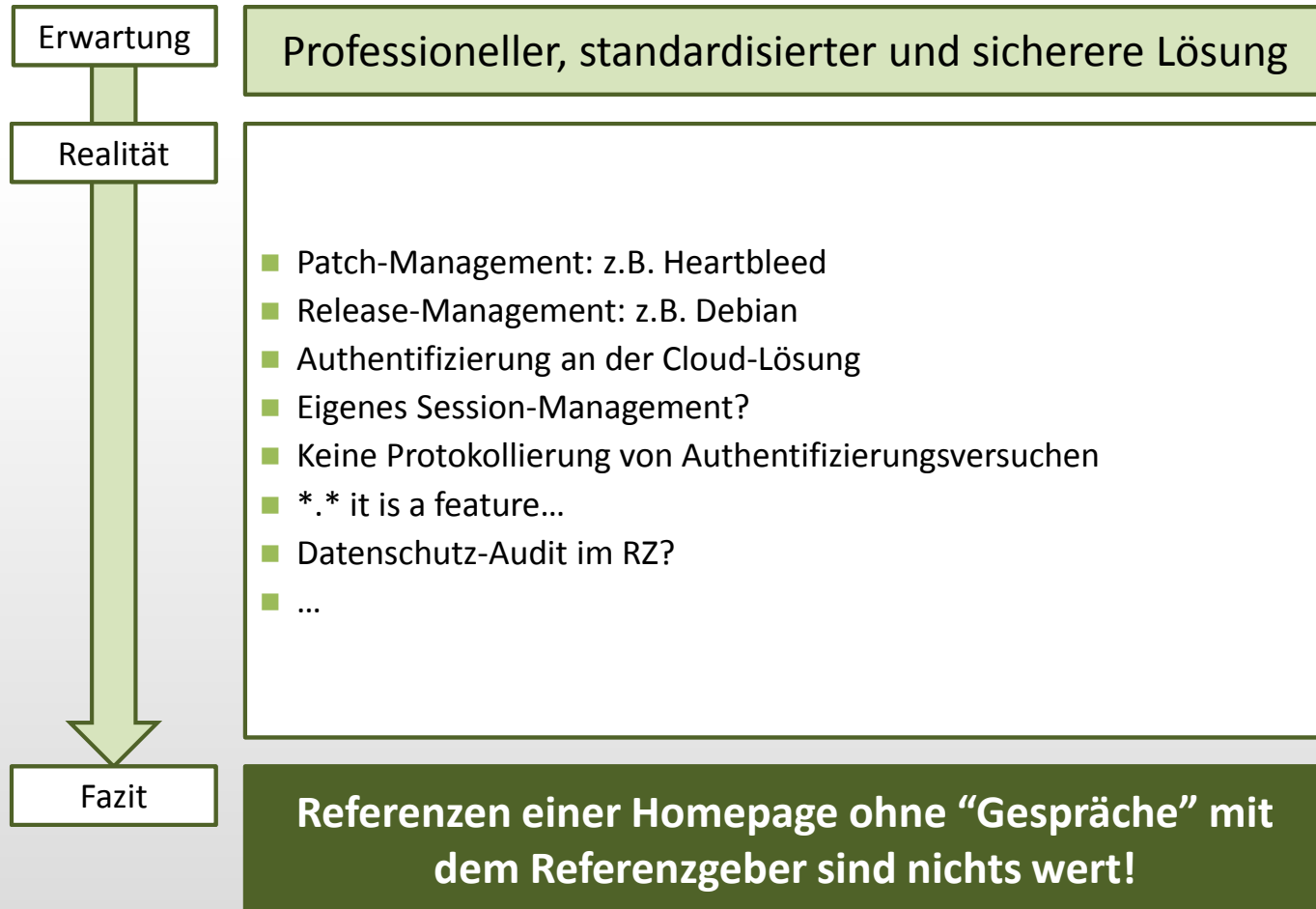
SaaS Service

Applikationsentwicklung

Summary

Das “Bauchgefühl” liegt oft richtig

Nutzung von SaaS Diensten mit „guten“ Referenzen



Hintergrund

Zertifizierungen

SaaS Service

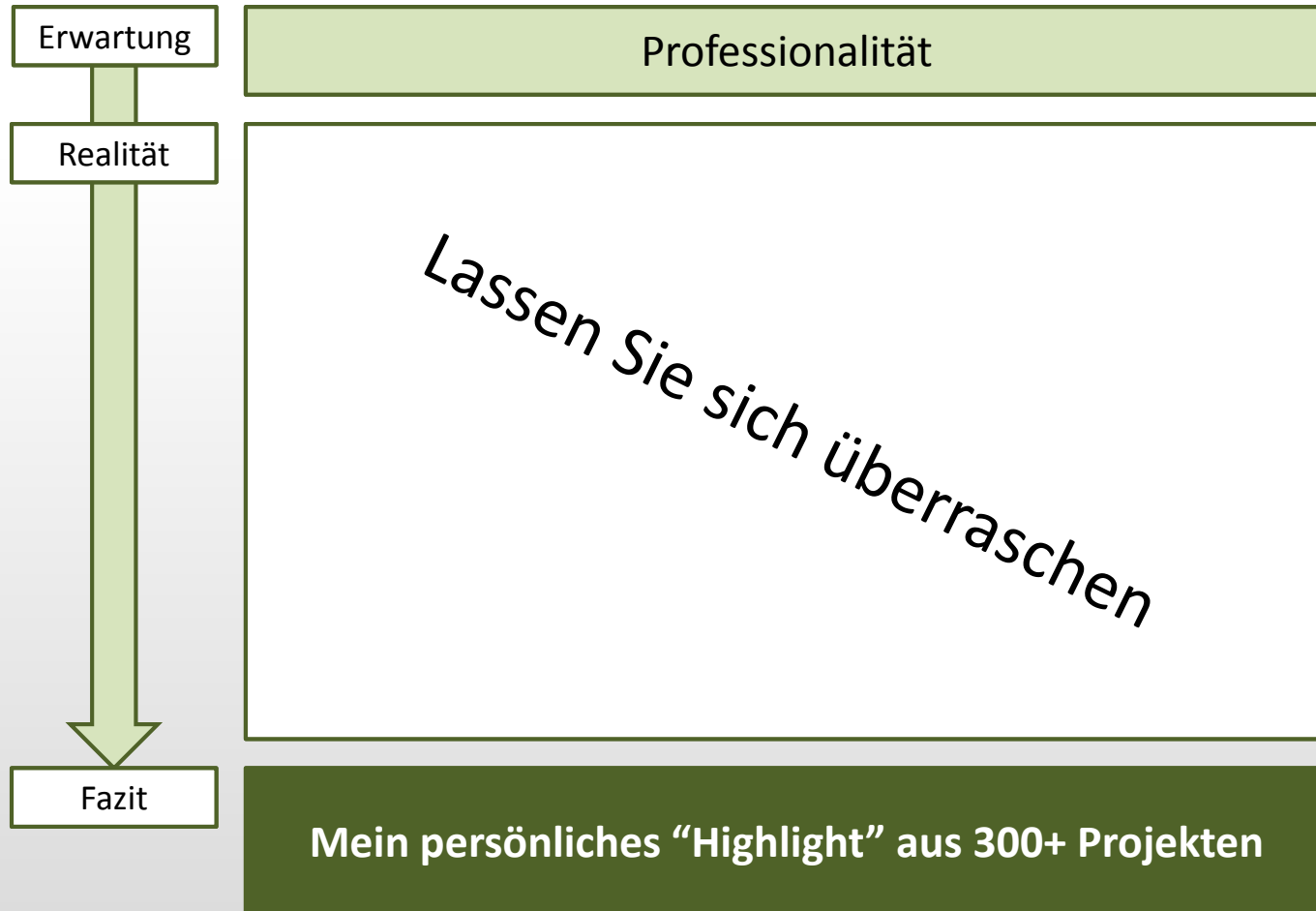


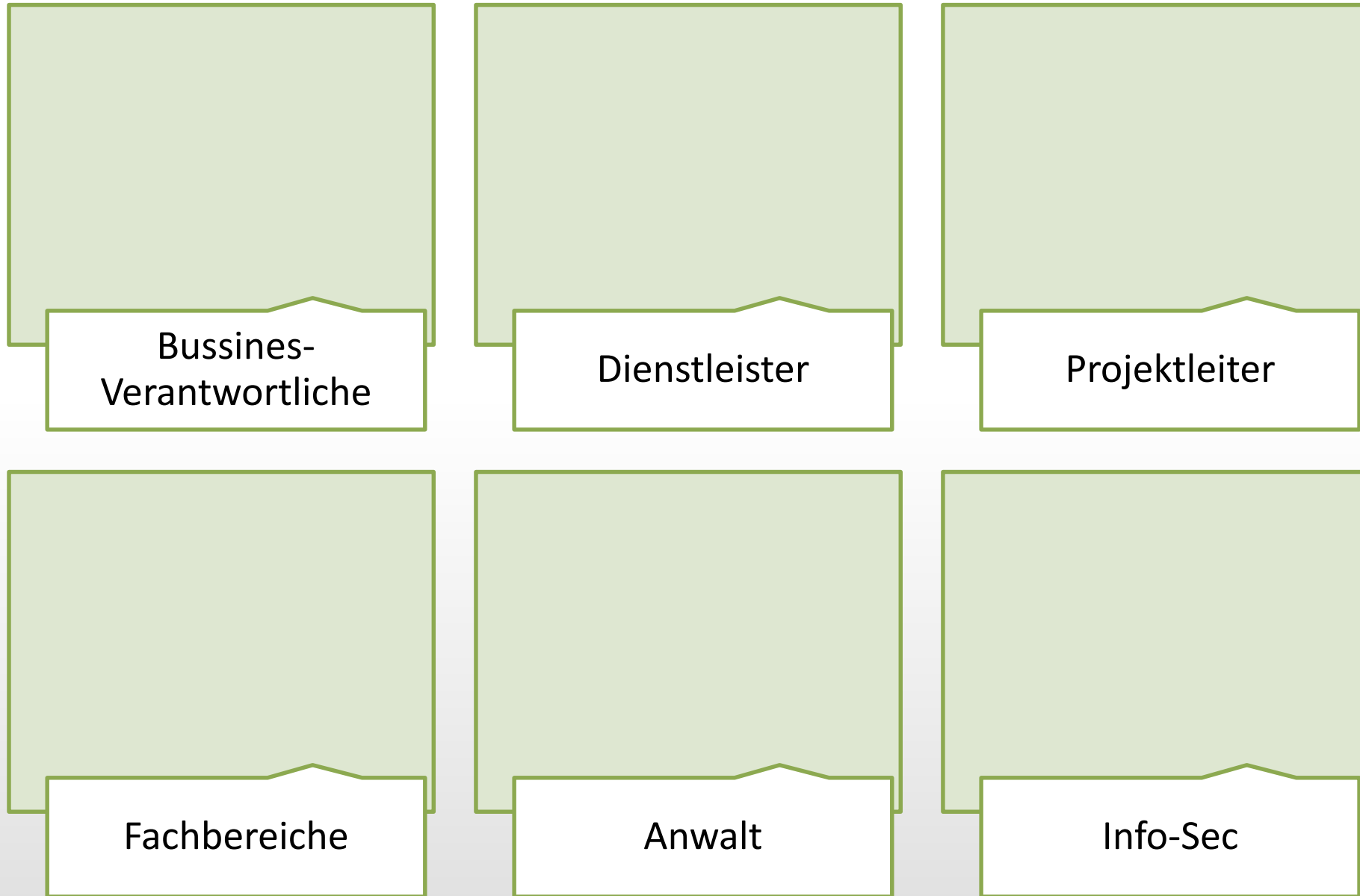
Applikationsentwicklung

Summary

Der “volle Stack” – keine Panik!

Modifikation einer On-Premise-Lösung für die Cloud





Ein „Ausschnitt“ der involvierten Charaktere

5.1.1 Input Validation

/ Beschreibung, welche Formen der Input-Validierung durchgeführt wurden */*

Die Eingaben werden geprüft. Entsprechen die Eingaben nicht der Vorgabe (Zahl, Datum, etc) wird ggf. eine Fehlermeldung für den User angezeigt.

5.1.2 Cross Site Scritping

/ Beschreibung der Sicherheitsmaßnahmen gegen Cross Site Scripting */*

Im Normalfall werden alle Ausgaben mit „c:out“ der JSTL ausgegeben. Dabei werden die Ausgaben maskiert.

Ausnahmen bilden die Inhalte der XXX. Hier müssen die originalen Eingaben unmaskiert ausgegeben werden um die Flexibilität für die Autoren zu erhalten.

Alle neue entwickelten Ansichten verwenden das JSTL Tag „c:out“ oder ein von uns erweitertes Tag. Dazu gehören alle Ansichten für „Nutzer“

Im Autorenbereich erfolgen die meisten Eingaben über Assistenten. In den Assistenten werden die Ausgaben ebenfalls mittels „c:out“ gefiltert.

In älteren Assistenten werden Scriptlets mit einem Filter verwendet. Diese Stellen werden Schrittweise auf das JSTL Tag „c:out“ umgestellt.

5.1.3 Interpreter Injection

/ Beschreibung der Sicherheitsmaßnahmen gegen Interpreter Injection */*

SQL Injections werden durch Prepared Statements verhindert.

Security OWASP (23.01.2014)

A3 – Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Prevention in [REDACTED]

- / Parameters are escaped by the used expression language which generates the input of data in an dynamic web-site.
- / Editors for content generation check for Java-Script input.
- / User generated content (e.g. Community message) are checked for script input and cleaned/escaped.

Statische Code Analyse:

The screenshot displays a static code analysis tool interface. At the top, the 'Filter Set' is set to 'Security Auditor View'. A progress bar shows the following counts: 3668 (Critical), 56630 (High), 36 (Medium), 45044 (Low), and 105378 (Info). The 'High (56630)' category is selected. The 'Group By' dropdown is set to 'Category'. The main list shows various security categories, each with a count of 0 out of a total number of items. The categories listed are:


- Access Control: Database - [0 / 50101]
- Code Correctness: Double-Checked Locking - [0 / 2]
- Code Correctness: Regular Expressions Denial of Service - [0 / 6]
- Cookie Security: Overly Broad Path - [0 / 7]
- Cross-Site Scripting: Reflected - [0 / 134]
- Denial of Service: Parse Double - [0 / 706]
- File Disclosure: J2EE - [0 / 9]
- File Disclosure: Spring - [0 / 383]
- Header Manipulation - [0 / 656]
- Header Manipulation: Cookies - [0 / 26]
- Insecure Randomness - [0 / 9]
- J2EE Bad Practices: Non-Serializable Object Stored in Session - [0 / 3]
- Log Forging - [0 / 1001]
- Null Dereference - [0 / 672]
- Password Management: Empty Password - [0 / 46]
- Password Management: Hardcoded Password - [0 / 1]
- Path Manipulation - [0 / 1533]
- Portability Flaw: File Separator - [0 / 2]
- Privacy Violation - [0 / 48]
- Privacy Violation: Heap Inspection - [0 / 3]
- SQL Injection - [0 / 65]
- System Information Leak: External - [0 / 9]
- Unreleased Resource: Database - [0 / 796]
- Unreleased Resource: Files - [0 / 3]
- Unreleased Resource: Streams - [0 / 231]
- Weak Encryption - [0 / 2]
- XML Entity Expansion Injection - [0 / 52]
- XML External Entity Injection - [0 / 110]
- XML Injection - [0 / 14]

On the right side of the interface, there are summary statistics: a green bar for '105378' and a red bar for '3668'. Below these, the text '/ 32]' is visible.

Die „Realität“

Was sagt der Dienstleister?

1.1 Ergebnisse des Audits

Bei der externen Sicherheitsüberprüfung für das Unternehmen [REDACTED] nach dem Blackbox Ansatz untersuchte [REDACTED] die externe Software [REDACTED]. Das Vorwissen des Überprüfenden über die internen Systeme war vergleichbar mit dem eines authentifizierten Nutzers der Plattform, der versucht seine Rechte im System zu erweitern 

Die identifizierten Schwachstellen deuten auf **systematische Fehler** im Application Security Management der Software hin.

1.1.1 Impact / Worst Case Szenarien

Im Wesentlichen sind folgende Angriffe möglich:

Ein **unauthentifzierter** Angreifer kann:

- Legitime Nutzer aus der Applikation aussperren.
- Effizient Brute-force-Angriffe auf Nutzerkonten durchführen.

Ein **authentifizierter** Angreifer kann abhängig von seiner konkreten Rolle:

- Tenant-übergreifend Daten lesen, manipulieren und löschen.
- Daten direkt vom Dateisystem des Servers lesen und so im Worst Case den Server komplett übernehmen.
- Nutzer der Anwendung mit JavaScript Trojanern und Malware infizieren.

Jetzt ist aber gut oder?

*Der Dienstleiter ist nicht berechtigt ohne vorherige ausdrückliche schriftliche Zustimmung seitens der XXX vor und während der Vertragsdurchführung geschäftliche, **technische oder wirtschaftliche Vorgänge, Abläufe oder sonstige Erkenntnisse gegenüber Dritten, auch nicht der [...] mitzuteilen***

Hintergrund

Beispiel Zertifizierungen

Beispiel SaaS Service

Beispiel Applikationsentwicklung



Summary

Summary

Annahmen und implizierte Vermutungen sind der Anfang vom Ende

Fragen?

- Definition von Security SLAs
- Prüfen, Prüfen, Prüfen
- Es wird getäuscht, verschleiert, getrickst und sich «blöd» gestellt.

Vielen Dank

@checkm4te // amir.alsbih@haufe-lexware.com