APEX – aber sicher

Wie sichert man APEX-Anwendungen gegen schädliche Manipulationen und unerwünschte Zugriffe ab?

Carola Berzl

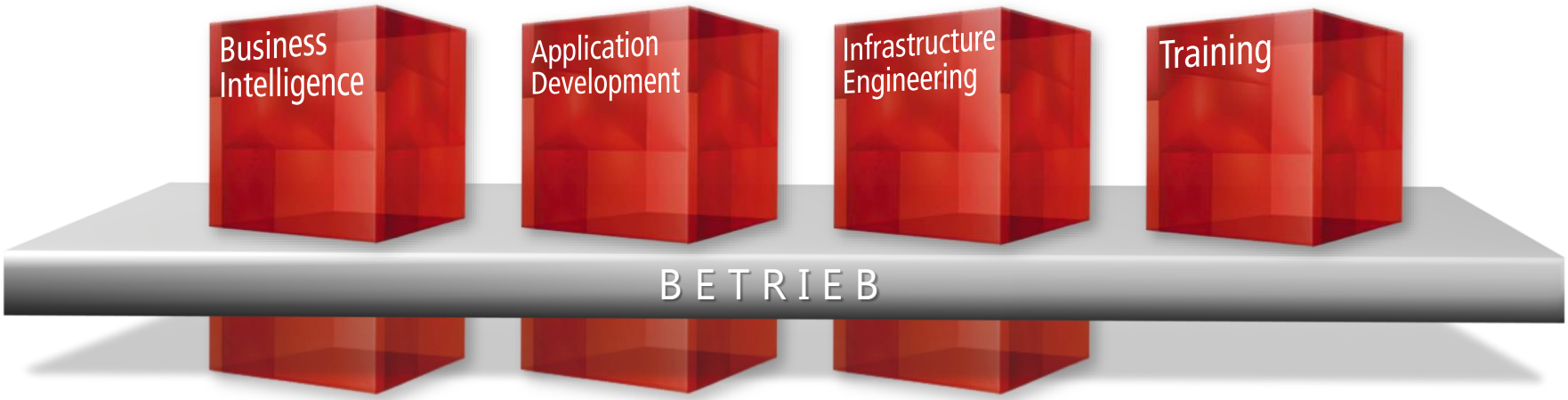
BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

trivadis
makes IT easier.

Unser Unternehmen.

Trivadis ist führend bei der IT-Beratung, der Systemintegration und der Erbringung von IT-Services mit Fokussierung auf ORACLE- und Microsoft-Technologien im D-A-CH-Raum.

Unsere Leistungen erbringen wir aus den strategischen Geschäftsfeldern:



Trivadis Services übernimmt den korrespondierenden Betrieb Ihrer IT Systeme.

Mit über 600 IT- und Fachexperten bei Ihnen vor Ort.

- 14 Trivadis Niederlassungen mit über 600 Mitarbeitenden.
- Über 200 Service Level Agreements.
- Mehr als 4'000 Trainingsteilnehmer.
- Forschungs- und Entwicklungs-budget: CHF 5.0 Mio. / EUR 4.0 Mio.
- Finanziell unabhängig und nachhaltig profitabel.
- Erfahrung aus mehr als 1'900 Projekten pro Jahr bei über 800 Kunden.

20 JAHRE TRIVADIS We love IT.

trivadis makes IT easier.

AGENDA

1. Einleitung
2. APEX Instanz
3. Maßnahmen für den individuellen Apex-Workspace
4. Maßnahmen für die Applikationen
5. Angriffsszenarien auf Anwendungen

Einleitung

Einleitung

- Eine sichere Umgebung für vertrauliche oder geheime Daten erfordert, dass alle Komponeneten die Security-Anforderungen erfüllen:
  - Webserver
  - Datenbank-Server
  - Datenbank
  - APEX Instanz (=Workspace internal)
  - Individueller APEX Workspace
  - Die APEX Anwendung

APEX Instanz

APEX Instanz (=Workspace Internal)

- Passwort-Policy einstellen
  - Ausreichende Passwortlänge und -komplexität
- Session Timeout einstellen
- Workspace Login Control einstellen
  - Passwort lifetime
  - Maximale Anzahl von fehlerhaften logins

APEX Instanz (=Workspace Internal)

- Apex-HTTPS Einstellungen

  (Aufruf von APEX nur über HTTPS)

- Nur die DBAs kennen das Passwort vom Apex-Instance Administrator

- RESTful Access ausschalten

- Automatisches Erstellen der Demo-Objekte abschalten

- Erstellung von Websheet Objects abschalten

APEX Instanz (=Workspace Internal)

- Application Activity Logging einschalten
- Enable Service Requests abschalten
- Set Workspace Cookie abschalten
- Allow Public File Upload „No“
- Provisioning Status „manual“

Individueller Workspace

Maßnahmen für den individuellen Workspace

- Workspace Administrator personell von Entwicklern trennen
- Starke Passwörter verwenden
- Abgelaufene Accounts löschen
- Workspace Login Control einstellen
  - Passwort lifetime
  - Maximale Anzahl von fehlerhaften logins
- Enable RESTful Services auf „No“ stellen
- Zugriff auf fremde Schemata
  - Ggf. „Transfer Schema“ erstellen
  - nur benötige Zugriffsrechte

Maßnahmen für die Applikation

Maßnahmen für die Applikation

- Authentifizierung
- Autorisierung
  - Verwaltung über Berechtigungsschemata
- Session Timeout einstellen
  - Maximum Session Length
  - Maximum Session Idle Time
- Sensible Informationen:
  - Clear Cache
  - Werte des Session Status verschlüsselt speichern

Maßnahmen für die Applikation

- Session State Protection konfigurieren und aktivieren
- Für Passwörter Password Items verwenden
  - Settings : Session Status nicht gespeichert
  - Security: Session State Protection einstellen
  - Does not save state auf „Yes“ setzen
- Session Cookie Attributes -> Secure auf „Yes“ setzen
  (Cookie wird nicht gesetzt wenn http verwendet wird)

Angriffsszenarien auf Anwendungen

Angriffsszenarien auf Anwendungen

- SQL Injection
- Cross Site Scripting
  - Reflexiver Angriff
  - Persistenter Angriff
  - Gegenmaßnahmen gegen XSS

Angriffsszenarien auf Anwendungen

- SQL Injection
  - ein Stück SQL-Code wird in ein Formular einfügt

    -> SQL-Anweisung wird verändert
  - Beispiel:
    - Dynamisches SQL:

      SQL-Befehle werden dynamisch durch Concatenating zusammengesetzt
    - Angreifer gibt SQL- oder PL/SQL-Codestücke ein

      -> Anderes Statement wird ausgeführt

SQL Injection

Demo

SQL Injection

- SQL Injection mit dynamischem SQL
  - Statement wird concateniert
  - z.B.: `l_sql := l_sql || ' WHERE deptno = ' || :P1_DEPTNO;`
  - Das Statement mit dem Schadcode wird zusammengefügt **bevor** der PL/SQL Block geparsed wird
  - Wenn ein Angreifer bestimmte Zeichenketten eingibt (z.B. 50 or 1=1)
    -> alle Datensätze werden angezeigt
- Gegenmaßnahme:
  - Variable in der Bind-Variablen Syntax in die Zeichenkette einbinden
  - z.B.: `l_sql := l_sql || ' WHERE deptno = :P1_DEPTNO';`

SQL Injection - Gegenmaßnahme

Demo

SQL Injection

- Praxistipp
  - SQL-Statement in einem Item anzeigen lassen
  - Debug-Ansicht

Angriffsszenarien auf Anwendungen

- Cross Site Scripting (=XSS)
  - Ausführung von JavaScript ist Ziel des Angriffs
  - Reflexiver Angriff
    - Schadcode wird vom Browser ausgeführt -> ein User ist betroffen
  - Persistenter Angriff
    - Schadcode wird in der DB gespeichert -> wird bei jedem Aufruf ausgeführt
  - Verteidigung: ausgegebene Zeichen maskieren

Cross Site Scripting (XSS) - reflexiv

Demo (Weine)

Cross Site Scripting (XSS) - persistent

Demo

Gegenmaßnahmen gegen XSS

- APEX_ESCAPE verwenden
  - Bei PL/SQL-Regionen

Bilder anzeigen in Berichtsspalten

- Gegenmaßnahmen
  - Attribut „Standard Report Column“ muss verwendet werden
  - In der Abfrage kein HTML verwenden
  - Bei den Report Attributes HTML Expression eintragen

Fragen und Antworten...

Carola Berzl

Carola.berzl@trivadis.com

BASEL BERN BRUGG GENF LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

20 JAHRE TRIVADIS We love IT.

trivadis makes IT easier.