netidee
PROJEKTE

SIKOSI
Checkliste für sichere Software-Entwicklung

Dokumentation | Call 14 | Projekt ID 4533

Inhalt

1 Checkliste

- ✓ Verwende ein aktuelles Framework und update dieses, wenn Updates vorhanden sind
- ✓ Halte verwendete Bibliotheken up-to-date
- ✓ Achte auf Ankündigen bezüglich Sicherheit
- ✓ Verwende Features wie “Always Encrypted” um sensitive Daten verschlüsselt abzuspeichern
- ✓ Verwende parametrierte SQL-Abfragen
  Verhindert SQL Injection
  Geeignete Technologie: EntityFramework
- ✓ Verschlüssele sensitive Daten in Konfigurationsdateien
- ✓ Verwende IMMER HTTPS!
- ✓ Stelle bei Produktivsystemen Tracen aus
- ✓ Verwende HSTS (HTTP Strict Transport Security)
- ✓ Lösche Server Header und Version Header, um keine Informationen nach außen über dein System zu verraten. Versichere dich, dass deine Applikation im Header nichts preisgibt.
- ✓ Verwende ASP.NET CORE Identity
- ✓ Speichere keine Passwörter im Klartext, benutze einen starken Hash
- ✓ Benutze TLS 1.2 für die gesamte Webseite
- ✓ Nutze ein Zertifikat – Kostenlose Zertifikate können bei LetsEncrypt (https://letsencrypt.org) angefordert werden.
- ✓ Achte darauf einen geeigneten XML Prozessor zu verwenden, damit Benutzereingaben nicht als Attacken benutzt werden können
- ✓ Versichere dich, dass Cookies das „HttpOnly“ Flag besitzen, somit können diese nicht mittels Javascript ausgelesen werden (XSS-Angriffe)
- ✓ Schütze Login, Registrierung und Passwort zurücksetzen Methoden vor Brute-Force Attacken
  (Benutze zum Beispiel bei ASP.NET das Attribut [AllowXRequestsEveryXSecondsAttribute])
- ✓ Benutze bei fehlerhaften Eingaben von Login-Daten eine allgemeine Fehlermeldung „Benutzer oder Passwort stimmen nicht überein“, ansonsten könntest du preisgeben wer die Benutzer deines Systems sind.
- ✓ Verwende Autorisierung auf Controller und Methoden-Ebene.
- ✓ Vergewissere dich, dass „Debug“ und „Trace“ im Produktivsystem abgeschalten sind.
- ✓ Verwende und validiere AntiforgeryToken um gegen XSS Attacken geschützt zu sein
- ✓ Logge und Monitore fehlerhafte Loginversuche und unautorisierte Versuche in das System zu gelangen

2 Verwendete Ressourcen

Microsoft Security .NET Standard

https://docs.microsoft.com/en-us/dotnet/standard/security/

OWASP – Security Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/DotNet_Security_Cheat_Sheet.html