Sicherheitsanalyse Webanwendung

Wechseln zu:Navigation, Suche

Dieser Artikel erfüllt die GlossarWiki-Qualitätsanforderungen **nur teilweise**:

Korrektheit: 2 (teilweise überprüft)	**Umfang**: 2 (wichtige Fakten fehlen)	**Quellenangaben**: 1 (fehlen großteils)	**Quellenarten**: 3 (gut)	**Konformität**: 3 (gut)

Inhaltsverzeichnis

1 Definition
2 Verfahren
3 Beispiele
- 3.1 M100 Data Validation: Filterung
- 3.2 M110 Data Validation: Whitelisting statt Blacklisting
- 3.3 M140 Data Validation: SQL-Injection verhindern
- 3.4 M150 Data Validation: Diverse Maßnahmen
- 3.5 M170 Session Management

4 Quellen

1 Definition

Schwachstellen in Internetanwendungen sind heutzutage allgegenwärtig. Die große Vielfalt an Technologien in Verbindung mit der Struktur für eine Softwareanwendung im Internet bieten daher viele verschiedene Angriffsmöglichkeiten und Schwachstellen. Um das Potenzial so gering wie möglich zu halten, ist es wichtig keine offensichtlichen oder bekannten Fehler in den Algorithmen und Strukturen der Anwendung zu haben.

2 Verfahren

Basierend auf dem Maßnahmenkatalog „Sicherheit von Webanwendungen[2]“ des „Bundesamt für Sicherheit in der Informationstechnik“ (kurz BSI) kann eine Analyse der typischen Schwachstellen durchgeführt. Das Dokument beinhaltet einen sehr übersichtlichen Maßnahmenkatalog. Die einzelnen Maßnahmen müssen für die Analyse nach ihrer Relevanz für die Anwendung sortiert und gewertet.

3 Beispiele

Im folgenden sind ein paar Beispielmaßnahmen aus dem Katalog aufgelistet.

3.1 M100 Data Validation: Filterung

Alle Daten, die von außen in die Anwendung gelangen, sind zu validieren und zu fil tern. Neben den offensichtlichen Eingabedaten in Form-Variablen existieren eine Reihe weiterer Quellen. Ebenso sind Ausgaben an den Browser zu filtern, wenn dies nicht bereits durch die Inputfilterung mit abgedeckt ist. Der Abschnitt "Output Filte rung" geht auf diese Frage näher ein.

3.2 M110 Data Validation: Whitelisting statt Blacklisting

Eine grundsätzliche Entscheidung ist die Wahl des Validierungs- bzw. Filterprinzips: Blacklisting oder Whitelisting. Beim Blacklisting werden die Muster definiert, die aus dem Eingabestrom herausgefiltert werden, alles andere wird durchgelassen. Beim Whitelisting ist es umgekehrt: was nicht ausdrücklich erlaubt ist, ist verboten. Black listing ist dann problematisch, wenn als "nicht kritisch" erkannte Zeichen im Verlaufe der weiteren Verarbeitung zu einem ungewollten Systemverhalten führen.

3.3 M140 Data Validation: SQL-Injection verhindern

Durch Verwendung von Prepared Statements anstatt dynamisch zusammengebauter SQL-Statements kann dem Problem der SQL-Injection aus dem Weg gegangen wer den. Software-Entwickler müssen sich im Rahmen des Designs einer Funktion oder hat bzw. was die Menge aller möglichen Abfragestrukturen ist, die in Abhängigkeit vom Input zur Ausführung gelangen können.

3.4 M150 Data Validation: Diverse Maßnahmen

Namen von Formularen überprüfen
Länge/Größe der Eingabedaten begrenzen
Serverseitig validieren
Datentypen der Eingabedaten
Ausgabedaten filtern

3.5 M170 Session Management

SessionID nach Authentisierung erneuern
Cookies einschränken
Session beenden

4 Quellen

Sicherheit von Webanwendungen
Gesundes Misstrauen

Kategorie:
Web-Programmierung

Diese Seite wurde zuletzt am 3. August 2019 um 13:51 Uhr bearbeitet.
Inhalt verfügbar unter CC BY-SA 4.0.