Bericht des Penetrationstests

von Webapplikation XYZ

für Beispiel XYZ GmbH

Version: 1.0
Datum: 20.08.2024
Klassifikation: VERTRAULICH

1 Dokumentenhistorie

Version	Autor	Bemerkung	Datum
0.1	Pen Tester	Projektversion	13.08.2024
0.2	Pen Tester	QS-Version	19.08.2024
1.0	Pen Tester	Kundenversion	20.08.2024

Inhaltsverzeichnis

2 Management Summary

Es wurde eine Sicherheitsüberprüfung der Webapplikation XYZ der Beispiel XYZ GmbH in Form eines Penetrationstests durchgeführt. Dabei wurde versucht, mit den Mitteln realer Angreifer technische Sicherheitsschwachstellen auszunutzen und mögliche Risiken für die Beispiel XYZ GmbH aufzuzeigen.

Die Erkenntnisse aus dieser Sicherheitsüberprüfung wurden in Form von risikobewerteten Schwachstellen im vorliegenden technischen Detailbericht dokumentiert.

Im Zuge des Penetrationstests wurden **4 Schwachstellen** ermittelt.

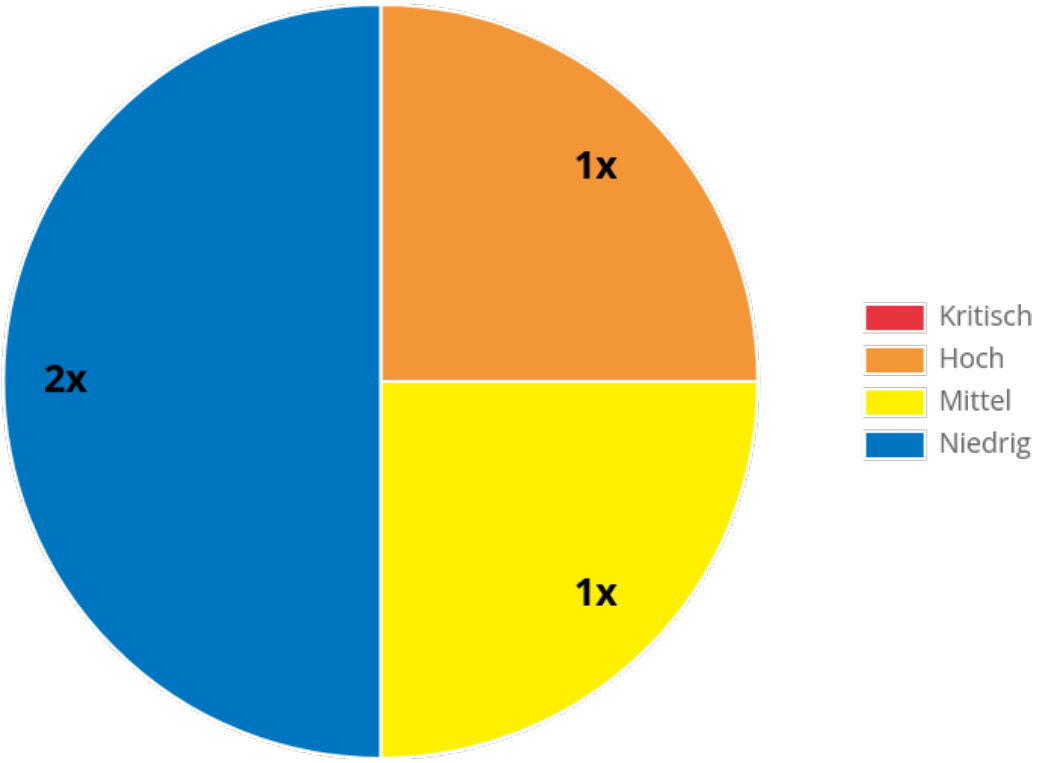


Abbildung 1 - Schwachstellenverteilung

In der Webanwendung konnten Benutzereingaben ungeprüft gespeichert wodurch, welche sogenannte "Stored Cross-Site-Scripting" - Angriffe (XSS) ermöglicht wurden. Die Ausnutzung von gespeicherten XSS-Schwachstellen erfordert keine Benutzerinteraktion, was sie gefährlicher macht als reflektierte XSS-Schwachstellen. Durch solche XSS-Angriffe könnten Session-Cookies gestohlen werden, um Aktionen als der eingeloggte Benutzer durchzuführen, oder die Funktionalität der Seite einzuschränken.

Zudem wurde festgestellt, dass die Webanwendung veraltete TLS-Versionen (1.0 und 1.1) unterstützt. Dies stellt ein erhebliches Sicherheitsrisiko dar, da diese Protokolle bekannte Schwachstellen aufweisen. Es wird dringend empfohlen, auf TLS 1.2 oder höher umzusteigen, um die Sicherheit der Datenübertragung gewährleisten zu können.

Während der Sicherheitsprüfung wurde festgestellt, dass der Webserver seine Version in der HTTP-Antwort preisgab. Die Offenlegung der Webserver-Version kann Angreifern wertvolle Informationen liefern, die sie für gezielte Angriffe nutzen können. Es ist ratsam, diese Informationen zu verbergen, um die Angriffsfläche zu minimieren.

Weiters wurde ebenso festgestellt, dass ein veraltetes Softwarepaket mit bestehenden Schwachstellen in der Anwendung verwendet wurden. Es wird empfohlen, die betroffene Software auf die neueste Version zu aktualisieren, um ein höheres Sicherheitsniveau zu erreichen.

3 Rahmenbedingungen

3.1 Umfang

Folgende Tabelle beschreibt die in der Vorbesprechung abgestimmten Komponente(n), welche die Sicherheitsüberprüfung umfasste:

Komponente	IP Adresse	Beschreibung
https://www.beispiel-xyz-gmbh.at/	127.0.0.1	Webapplikation XYZ

3.2 Verwendete Benutzerkonten

Folgende Tabelle gibt einen Überblick über die während des Penetrationstests verwendeten Benutzerkonten:

Benutzerkonto	Rolle(n) / Berechtigung(en)	Beschreibung
pentester1	Benutzer der Webapplikation mit administrativen Berechtigungen	Testbenutzer für den Login der Webapplikation

3.3 Tester

Folgende Mitarbeiter der Raiffeisen Informatik GmbH & Co KG waren an der Planung und Ausführung der Sicherheitsüberprüfung beteiligt:

Mitarbeiter	Kontaktdaten
Pen Tester	+43 123 456789
Pen Tester	+43 123 456789

3.4 Tätigkeiten

Folgende Tabelle gibt den chronologischen Ablauf der Sicherheitsüberprüfung wieder:

Zeit	Aktivität
06.08.2024	Kick-Off-Besprechung
07.08.2024 - 08.08.2024	Vorbereitung des Penetrationstests
09.08.2024 - 19.08.2024	Durchführung des Penetrationstests & Dokumentation der Ergebnisse
20.08.2024	Durchführung der Qualitätssicherung & Übermittlung Kundenversion

3.5 Methodologie Schwachstellenbewertung

Zur Sicherstellung, dass die im Zuge der Sicherheitsüberprüfung identifizierten Schwachstellen transparent, nachvollziehbar und vor allem vergleichbar sind, werden diese mittels CVSS (*Common Vulnerability Scoring System*), einem offenen Industriestandard der FIRST.Org, Inc. (FIRST) in der Version 3.1 bewertet.
Für jede Schwachstelle wird der entsprechende ***CVSS-Score*** ermittelt, welcher den Schweregrad der möglichen Auswirkungen bei Ausnutzung der Schwachstelle angibt.

Bei Schwachstellen, für welche sich kein CVSS-Score berechnen lässt, behalten sich die Tester vor, eine qualitative Bewertung ohne *CVSS-Score* vorzunehmen. Die Interpretation dieser Schwachstellen erfolgt analog zu Schwachstellen mit CVSS-Bewertung, jedoch ist dabei die Spalte "*CVSS-Score*" zu ignorieren.

Zur quantitativen Berechnung gemäß CVSS-Standard werden die Schwachstellen zusätzlich zur Vereinfachung qualitativ bewertet und wie nachfolgend dargestellt in Klassen kategorisiert.

Schweregrad	CVSS-Score	Beschreibung
Kritisch	9.0 – 10.0	Die Ausnutzung dieser Schwachstelle kann die Kompromittierung von Systemen oder Zugriff auf kritische Anwendungsbereiche zur Folge haben. Da dies meist ohne zusätzliche Information über das System oder durch automatisierte Scripts ausgenutzt werden kann, sollte das Risiko einer Schwachstelle mit kritischem Schweregrad auf keinen Fall akzeptiert werden.
Hoch	7.0 – 8.9	Die Auswirkung solch einer Schwachstelle kann ähnlich zu einer kritischen Schwachstelle sein. Der Unterschied liegt jedoch meist in der Eintrittswahrscheinlichkeit, da zur Ausnutzung zusätzliche Voraussetzungen, wie vorher bereits (berechtigte) Zugänge oder die Unterstützung eines (dazu verleiteten) berechtigten Benutzers, benötigt wird.

Schweregrad	CVSS-Score	Beschreibung
Mittel	4.0 – 6.9	Die Ausnutzung einer Schwachstelle mit mittlerem Schweregrad gewährt meist nur eingeschränkten Zugriff oder hat nur eingeschränkte Auswirkungen auf die Nutzbarkeit. In Kombination mit anderen Schwachstellen können diese Schwachstellen dennoch höhere Schweregrade erreichen und sollten immer im Kontext mit anderen Schwachstellen berücksichtigt werden.
Niedrig	0.1 – 3.9	Für die Ausnutzung einer niedrig bewerteten Schwachstelle ist meist der Zugriff auf ein lokales oder physisches System erforderlich oder es bedarf bereits umfangreicher vorher vorhandener Zugänge. Oft existieren auch keine öffentlich bekannten Angriffswerkzeuge.
Info	0.0 – 0.0	Bei diesem Schweregrad handelt es sich um Maßnahmen, um eventuelle Schwachstellen zu verhindern oder die Auswirkung zu reduzieren, stellt alleine jedoch keine Schwachstelle dar.

4 Risikoüberblick

4.1 Ergebnisübersicht

Nachfolgend beschriebene Risiken, Schwachstellen und Empfehlungen konnten im Zuge der Sicherheitsüberprüfung identifiziert werden. Diese werden geordnet nach deren Kritikalität aufgelistet. Eine detaillierte Beschreibung jeder Maßnahme erfolgt im technischen Detailbericht.

4.1.1 Webapplikation XYZ

Bewertung	Name
Hoch (7.6)	Stored Cross-Site-Scripting (XSS) möglich
Mittel (qual.)	Unterstützung von TLS 1.0 und TLS 1.1
Niedrig (qual.)	Offenlegung der verwendeten Webserver-Version
Niedrig (qual.)	Verwendung von Software mit Schwachstellen

5 Technischer Detailbericht

In diesem Abschnitt finden sich detaillierte Informationen zu den identifizierten Schwachstellen und Sicherheitsproblemen mit entsprechenden Maßnahmenempfehlungen. Diese Maßnahmenempfehlungen stellen den von einem Sicherheitsexperten der Raiffeisen Informatik GmbH & Co KG im Kontext als „Security Best Practice“ oder Stand der Technik erachteten adäquaten Lösungsweg zur Bereinigung des identifizierten Sicherheitsproblems dar.

Natürlich kann unter Rücksichtnahme auf entsprechende Spezifika des jeweiligen Systems, der Applikation oder aufgrund organisatorischer oder technischer Unverhältnismäßigkeit des zur Umsetzung erachteten Aufwands von dieser Empfehlung Abstand genommen werden, diese Entscheidung muss jedoch für die Sicherheitsverantwortlichen des jeweiligen Systems getroffen werden und kann von den Sicherheitsexperten der Raiffeisen Informatik GmbH & Co KG nicht vorweggenommen werden. Sollte dieser Fall eintreten, können alternative Maßnahmen, soweit nicht ohnehin schon vorgeschlagen, mit ähnlichem Schutzniveau dargestellt werden.

Soweit vorhanden, werden in diesem Abschnitt auch Proof-of-Concept (PoC) Attacken bzw. Code-Teile oder skizzierte Angriffswege dargestellt. Diese dienen zum Beweis der Ausnutzbarkeit einzelner Schwachstellen oder zurIllustration des Sicherheitsrisikos durch die Kombination unterschiedlicher Schwachstellen und der daraus resultierenden Angriffsvektoren.

Sollten zu einzelnen Schwachstellen keine Angriffsvektoren oder PoC-Attacken explizit dargestellt sein, ist dies nicht direkt mit der Unausnützbarkeit dieser Schwachstellen gleichzusetzen, sondern muss auf die gesetzten Limits hinsichtlich der Zeit der Durchführung der Sicherheitsüberprüfung oder der nicht-öffentlichen Verfügbarkeit dafür notwendiger Programme und Werkzeuge zurückgeführt werden. Dies wird in der Risikoklassifizierung entsprechend berücksichtigt.

5.1 Webapplikation XYZ

5.1.1 Stored Cross-Site-Scripting (XSS) möglich

Bewertung	**Hoch (7.6)**
CVSS-Vektor	CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N
Beschreibung	Während der Sicherheitsprüfung wurde festgestellt, dass die Webanwendung Benutzereingaben ungeprüft speicherte. Durch die Speicherung von ungeprüften JavaScript-Code war die Webapplikation anfällig für *Stored Cross-Site-Scripting*-Angriffe (XSS). Die Ausnutzung von gespeicherten XSS-Schwachstellen erfordert keine Benutzerinteraktion, was sie gefährlicher macht als reflektierte XSS-Schwachstellen.
Betroffenes Objekt	https://www.beispiel-xyz-gmbh.at/

Details

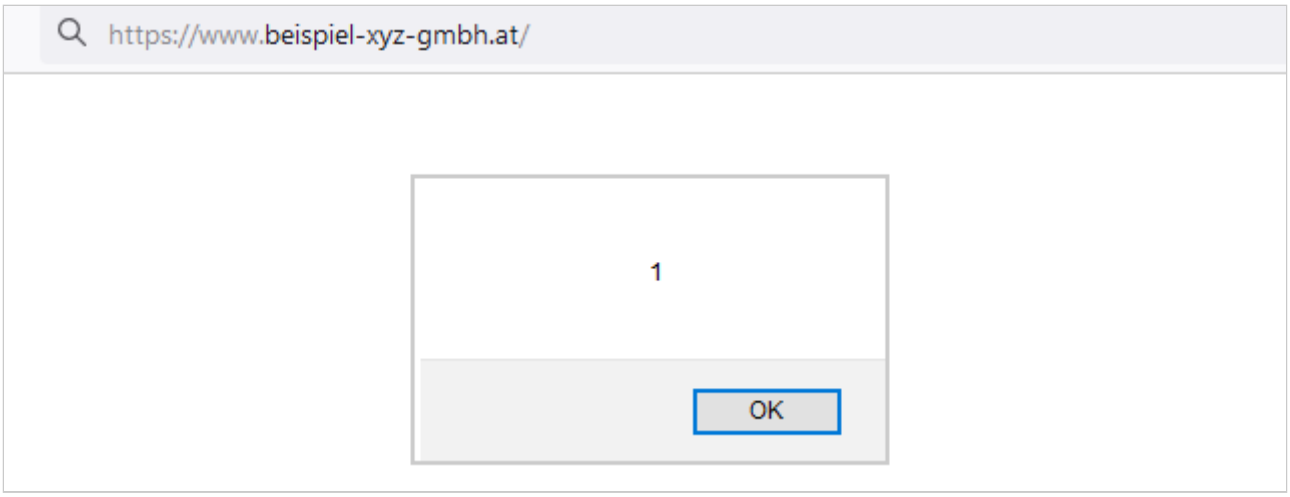
Im Zuge der Sicherheitsüberprüfung gelang es, eine *Stored Cross-Site-Scripting*-Schwachstelle in der Funktion `Blogposts()` zu identifizieren.

Cross-Site-Scripting (XSS) ist eine Sicherheitslücke bei Webseiten, bei der über ungeprüfte Benutzereingaben bösartige Skripte eingeschleust werden können. Bei *Stored Cross-Site-Scripting* (XSS)-Angriffen wird JavaScript-Code auf Seiten platziert, die von anderen Benutzern besucht werden. Der Code wird im Webbrowser des Opfers ausgeführt und kann gegebenenfalls auf sensible Informationen wie Session-Cookies zugreifen.

Um die Schwachstelle auszunutzen, musste beim Bearbeiten eines Blogposts in den Feldern `Titel` und `Kurzbeschreibung` ein JavaScript-Code eingegeben werden. Nur diese Felder waren gegen den XSS-Angriff verwundbar.

In diesen Feldern wurde – wie in folgendem Screenshot gezeigt – ein einfacher JavaScript-Code eingegeben, um festzustellen, ob dieser ausgeführt wird:

```
<script>alert(1)</script>
```



Maßnahmenempfehlungen

Grundsätzlich müssen Eingabefelder auf eingebettete HTML-Elemente und JavaScript-Code geprüft werden, bevor deren Inhalt gespeichert wird.

Alle vom Benutzer bzw. vom Browser des Benutzers übergebenen Parameter müssen geprüft werden und bei der Ausgabe „aktive“ Zeichen (z.B. `<`) richtig kodiert und damit unschädlich gemacht werden.

Es sollten entsprechende Ein- und Ausgabefilter implementiert werden, um nur zulässige Zeichenketten vom Benutzer zu übernehmen und als normalen Text wieder zurückzusenden. Dabei sollte nach dem Whitelist-Prinzip vorgegangen werden, d. h. der eingesetzte Filter erlaubt nur ungefährliche Zeichenketten. Außerdem wird zur Erleichterung der Filterung im Allgemeinen im ersten Schritt eine Normalisierung der Eingabedaten durchgeführt.

Weitere Möglichkeiten können den weiterführenden Informationen entnommen werden.

Weiterführende Informationen

- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- https://cwe.mitre.org/data/definitions/79.html

5.1.2 Unterstützung von TLS 1.0 und TLS 1.1	
Bewertung	**Mittel (qualitativ)**
Beschreibung	Der getestete Server unterstütze zum Zeitpunkt der Sicherheitsprüfung die TLS Protokolle der Version 1.0 und 1.1, welche seit März 2021 nicht mehr verwendet werden sollten.
Betroffenes Objekt	https://www.beispiel-xyz-gmbh.at/

Details

Die beiden Protokolle TLS 1.0 und TLS 1.1 gelten seit Jahren als unsicher und unterstützen demnach keine modernen bzw. sicheren kryptografischen Algorithmen mehr. Bekannte Schwachstellen wie **BEAST** ("Browser Exploit Against SSL/TLS") oder **FREAK** ("Factoring Attack on RSA-Export Keys") sind mögliche Angriffsvektoren der beiden Protokolle. Im März 2021 wurden die TLS Protokolle v1.0 und v1.1 offiziell von der IETF (Internet Engineering Task Force) als veraltet erklärt.

Die folgende Kommandozeilenausgabe zeigt, dass die betroffenen Server die veralteten TLS Protokolle unterstützten:

```
SCAN RESULTS FOR [redacted]
------------------------------------------------

* TLSV1_1 Cipher Suites:
      Forward Secrecy                    OK - Supported
      RC4                                OK - Not Supported

    Preferred:
       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA              ECDH-521 bits  256 bits
    Accepted:
       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA              ECDH-521 bits  256 bits
       TLS_DHE_RSA_WITH_AES_256_CBC_SHA                DH-2048 bits   256 bits
       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA           DH-2048 bits   256 bits
       TLS_RSA_WITH_AES_256_CBC_SHA                    -              256 bits
       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA              ECDH-521 bits  128 bits
       TLS_DHE_RSA_WITH_AES_128_CBC_SHA                DH-2048 bits   128 bits
       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA           DH-2048 bits   128 bits
       TLS_RSA_WITH_AES_128_CBC_SHA                    -              128 bits

* TLSV1 Cipher Suites:
      Forward Secrecy                    OK - Supported
      RC4                                OK - Not Supported

    Preferred:
       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA              ECDH-521 bits  256 bits
    Accepted:
       TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA              ECDH-521 bits  256 bits
       TLS_DHE_RSA_WITH_AES_256_CBC_SHA                DH-2048 bits   256 bits
       TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA           DH-2048 bits   256 bits
       TLS_RSA_WITH_AES_256_CBC_SHA                    -              256 bits
       TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA              ECDH-521 bits  128 bits
       TLS_DHE_RSA_WITH_AES_128_CBC_SHA                DH-2048 bits   128 bits
       TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA           DH-2048 bits   128 bits
       TLS_RSA_WITH_AES_128_CBC_SHA                    -              128 bits
```

Seit Ende 2020 unterstützen Browser wie Google Chrome, Firefox oder Microsoft Edge die beiden Protokolle TLS 1.0 und 1.1 nicht mehr.

Aus diesem Grund stellen die Websites, die sich auf die unsicheren Versionen verlassen, ein Sicherheitsrisiko für eine sichere Verbindung dar.

Maßnahmenempfehlungen

Es wird empfohlen, die aktuelle TLS-Version 1.3, oder zumindest TLS 1.2, zu unterstützen und TLS 1.0 und TLS 1.1 sowie alle älteren zu deaktivieren.

Weiterführende Informationen

- https://ssl-config.mozilla.org/
- https://datatracker.ietf.org/doc/html/rfc8996

5.1.3 Offenlegung der verwendeten Webserver-Version

Bewertung	**Niedrig (qualitativ)**
Beschreibung	Während der Sicherheitsprüfung wurde festgestellt, dass der Webserver seine Version in der HTTP-Antwort preisgab.
Betroffenes Objekt	https://www.beispiel-xyz-gmbh.at/

Details

Es wurde festgestellt, dass der Webserver seine Version `Microsoft IIS/10.0` und die Nutzung des Frameworks `ASP.NET` in HTTP-Antworten preisgab.

Jede Information über verwendete Software, Typ, Version, usw. bietet einem Angreifer die Möglichkeit, öffentlich verfügbare Schwachstelleninformationen zu durchsuchen, um damit gezielte Angriffe durchführen zu können. Er kann auch Angriffsvektoren, die bei der speziellen Software bzw. in der spezifischen Konfiguration nicht erfolgreich sind, von vornherein ausschließen.

Maßnahmenempfehlungen

Generell sollte die Konfiguration so verändert werden, dass keine technischen Informationen über eingesetzte Technologien und Versionsnummern preisgegeben werden.

Dies bedeutet, dass für alle oben genannten Dienste der `Server`-, `X-Powered-By`-Header, etc. deaktiviert werden sollte.

Weiterführende Informationen

https://cwe.mitre.org/data/definitions/497.html

5.1.4 Verwendung von Software mit Schwachstellen

Bewertung	**Niedrig (qualitativ)**
Beschreibung	Die betroffene Anwendung verwendete Software, die zahlreiche Schwachstellen aufwies bzw. nicht mehr unterstützt wurde.
Betroffenes Objekt	https://www.beispiel-xyz-gmbh.at/

Details

Im Zuge der Sicherheitsüberprüfung wurde festgestellt, dass ein Softwarepaket verwendet wurde, das veraltet ist und bekannte Schwachstellen aufweist.

Konkret betraf dies folgende Softwarepaket:

- jQuery v2.1.4

Das Softwarepaket `jQuery v2.1.4` wurde beim Öffnen eines Blogeintrages (`https://www.beispiel-xyz-gmbh.at/js/jquery-2.1.4.js`) geladen.

Diese Version des Softwarepakets ist verwundbar gegen Cross-Site-Scripting (XSS) und "Prototype Pollution".

Maßnahmenempfehlungen

Die oben genannten Software-Versionen sollten so schnell wie möglich durch Versionen ersetzt werden, die frei von bekannten Schwachstellen sind.

Generell sollte sichergestellt werden, dass die Systeme immer aktuell gehalten werden.

Es wird empfohlen, eine periodische Überprüfung der Aktualität der eingesetzten Software zu etablieren. Dies kann entweder durch einen entsprechenden Prozess oder im Rahmen einer periodischen Sicherheitsüberprüfung umgesetzt werden.

Weiterführende Informationen

- https://cwe.mitre.org/data/definitions/1395.html
- https://security.snyk.io/package/npm/jquery/2.1.4