Bundesamt
für Sicherheit in der
Informationstechnik

BSI-Veröffentlichungen zur Cyber-Sicherheit

EMPFEHLUNG: INTERNET-DIENSTLEISTER

Sicherheitsmaßnahmen beim Einsatz aktiver Inhalte

Verwendung aktiver Inhalte durch Anbieter von Webanwendungen

Bei der Gestaltung moderner Internet-Angebote greifen Entwickler von Webseiten regelmäßig auf aktive Inhalte zurück, die gegenüber statischem HTML insbesondere einen höheren Grad an dynamischer Interaktion mit dem Nutzer erlauben. Mittels aktueller Webtechniken, wie HTML5 und JavaScript, werden im Browser ausgeführte Anwendungen möglich, die zuvor nur mit lokal installierten Programmen umsetzbar waren.

Die Verwendung aktiver Inhalte, sowohl für Anbieter von Webseiten als auch deren Nutzer, erfordert die Umsetzung von umfangreichen Maßnahmen auf Seiten der Anbieter zu deren technischer Absicherung, um bestehenden Risiken hinreichend zu begegnen. Bei der Planung der Sicherheitsmaßnahmen ist es wichtig, dass der Anbieter sich seiner Verantwortung für die Sicherheit seiner Nutzer jederzeit bewusst ist, indem er darauf achtet, sie keinen unnötigen Risiken auszusetzen. Aus diesem Grund dienen die im Folgenden vorgestellten Maßnahmen nicht nur dem Schutz der Anbieter, sondern auch dem Schutz der Nutzer. Unabhängig vom Schutzbedarf des Anbieters kann daher auf die Umsetzung dieser Maßnahmen nicht verzichtet werden.

1 Aktive Inhalte

Die Nutzung von aktiven Inhalten jedweder Art ist durch den Dienstanbieter auf das Notwendige zu beschränken. Hierzu zählen in erster Linie Funktionalitäten, welche der Nutzerführung und der einfachen Zugänglichkeit von Inhalten (Accessibility) förderlich sind. Sofern aktive Inhalte zum Einsatz kommen, sollte der Dienstanbieter ausschließlich verbreitete Webtechniken, deren Unterstützung bereits in modernen Browsern in Verbindung mit geeigneten Sicherheitsmaßnahmen integriert ist, verwenden; dazu gehören etwa HTML5-Elemente sowie JavaScript. Andere aktive, über das Internet verteilte und dann im Browser ausgeführte Techniken sollten aus sicherheitstechnischen Gründen *nicht* verwendet werden. So wird eine mit diesen Techniken verbundene mangelnde Interoperabilität vermieden und die Angriffsfläche wirksam minimiert.

Die Ausführung von aktiven Inhalten kann zur Sicherstellung der Barrierefreiheit und durch Filter an Sicherheitsgateways oder Netzübergängen sowie durch den Nutzer selbst blockiert werden. Daher ist stets die vollständige Nutzbarkeit der über die Webseite angebotenen Inhalte und Dienstleistungen auch ohne die Verwendung aktiver Inhal-

te zu ermöglichen. Aktive Inhalte sollen die Nutzung der Webanwendung vereinfachen oder effizienter gestalten, jedoch keine zwingende Voraussetzung für die grundsätzliche Nutzung des Angebots bilden. Dies gilt insbesondere auch zur Gewährleistung einer barrierefreien Nutzung der Webseite.

Aufgrund der hier beschriebenen Eingrenzung auf wenige, sicherheitstechnisch akzeptable aktive Inhalte, beschränken sich im Folgenden auch die empfohlenen Maßnahmen auf die Verwendung von JavaScript und HTML5. Obwohl die beschriebenen Maßnahmen auch bei anderen aktiven Inhalten ggf. eine Schutzwirkung entfalten können, stellen sie keine ausreichende Grundlage zu deren Absicherung dar. JavaScript im Kontext dieser Empfehlungen bezieht sich auf die ECMAScript® Language Specification[1] sowie die JavaScript Reference von Mozilla[2], HTML5 auf die entsprechenden Spezifikationen des World Wide Web Consortiums (W3C)[3].

2 Risiken

Folgende Risiken können die Anbieter von Webseiten mit aktiven Inhalten und/oder auch die Nutzer betreffen:

- Erhöhung der Angriffsfläche bei vorhandenen Schwachstellen im Browser durch erweiterte Möglichkeiten der Speicherkontrolle für den Angreifer mit dem Ziel des Ausbruchs aus der schützenden Sandbox des Browsers.
- Verletzung der Same-Origin-Policy des Browsers mit dem Ziel der Manipulation oder des Abhörens von Inhalten anderer Webseiten, ebenso wie die Manipulation oder das Abhören der eigentlich besuchten Seiten durch den Angreifer.
- eine unwissentliche Beteiligung des Nutzers an Distributed-Denial-of-Service-Angriffen (DDoS) mittels von Angreifern manipulierten und dann im Browser ausgeführten aktiven Inhalten, wie JavaScript-Code.
- Gefahr durch Malware, die über ansonsten vertrauenswürdige Server auf den Rechner der Nutzer gelangt, indem durch Angreifer serverseitige Lücken ausgenutzt werden, um eigenen Code auf dem Server abzulegen.
- Man-in-the-Middle-Angriffe gegen den von ihnen ausgelieferten JavaScript-Code.
- Injektionsangriffe gegen die oder mittels der über aktive Inhalte ausgelieferten Funktionen und Schnittstellen zu Server-seitig gehaltenen Daten.
- Erhöhung der Angriffsfläche für Defacement-Angriffe mit gleichzeitig höherem Aufwand für die regelmäßigen Maßnahmen zu deren Erkennung.
- Erfordernis für im Vergleich zu statischen Inhalten umfangreicheren Tests vor einer öffentlichen Bereitstellung neuer Inhalte, verbunden mit dem Risiko unentdeckt bleibender Schwachstellen aufgrund der höheren Komplexität des Webangebots.
- die Client-seitige Verletzung der Same-Origin-Policy und damit Angriffe gegen die vom Anbieter bereitgestellten Inhalte bis hin zur Manipulation von auf dem Server gehaltenen Daten.

1 https://www.ecma-international.org/ecma-262/11.0/
2 https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
3 http://www.w3.org/TR/html5/

- Angriffe gegen die oder mittels der vom Anbieter verwendeten JavaScript-Bibliotheken, die dynamisch oder statisch in das eigene Angebot eingebunden werden und nicht vollständig unter Kontrolle des Anbieters stehen. Es ist zu beachten, dass bei einem externen Hosting die IP-Adresse des Benutzers dem externen Hoster bekannt wird.
- Eine naturgemäß signifikant größere Bandbreite an Möglichkeiten, die Angreifern zur Kompromittierung von Client- und Serversystemen zur Verfügung steht und der damit mit differenzierten Maßnahmen wirksam begegnet werden muss.

3 Maßnahmen für Anbieter

3.1 Abhängigkeiten zum verwendeten Browser

Anbieter von Webseiten müssen sicherstellen, dass das Angebot mit gängigen Browsern vollständig dargestellt und genutzt werden kann. Browser- und Herstellerspezifische Funktionen in HTML und JavaScript sind daher zu vermeiden - insbesondere dann, wenn sie Nutzer zur Verwendung unsicherer Browser, Plugins oder anderer Komponenten zwingen. Es sollten ausschließlich Techniken die auf offenen Standards[4] basieren eingesetzt werden, um eine ausreichende Interoperabilität und sicherheitstechnische Überprüfbarkeit der Techniken zu ermöglichen. Zur Orientierung, welche Browser in welchen Konfigurationen von Nutzern verwendet werden, können u.a. die BSI-Empfehlung für sichere Web-Browser[5], der IT-Grundschutz zu Web-Browsern[6] und der Mindeststandard des BSI für Web-Browser[7] herangezogen werden.

Die Messung der aktuellen Verteilung der auf Nutzerseite eingesetzten Web-Browser ist – abhängig von der konkreten Messmethode – mit systematischen Fehlern verbunden. Die Daten der Anbieter entsprechender Analysekomponenten oder Hosting-Infrastrukturen, wie z. B. StatCounter[8], W3Counter[9] oder Akamai[10], zeigen jedoch, dass der Einsatz von Web-Browsern, die in der Lage sind, dynamische Webangebote mit standardkonformen aktiven Inhalten zu verarbeiten, vorausgesetzt werden kann.

Insbesondere die automatischen Mechanismen zur Installation von Sicherheitsaktualisierungen in den am meisten verbreiteten Browsern zeigen hier Wirkung: Wenn der Nutzer den Browser in seiner aktuellen Version einsetzt, kann der Anbieter von Webanwendungen davon ausgehen, dass die Browser-seitigen Schutzmechanismen umgesetzt werden.

Es ist jedoch darauf zu achten, dass besonders ältere Smartphones/Tablets oft nur noch eingeschränkt durch Updates unterstützt werden.

3.2 Sichere Entwicklung

Bereits bei der Entwicklung von Webseiten, aktiven Inhalten und Diensten, die aktive Inhalte beim Betreiber bereit stellen, muss auf eine sichere Programmierung und Entwicklung von robustem Quellcode geachtet werden. Viele Angriffe werden erst möglich, weil grundlegende Prinzipien bei der Entwicklung, wie Eingabedatenüberprüfung (Input Validation), korrekte Kodierung von Daten, sichere Authentisierung oder die Nutzung von Verschlüsselung nicht beachtet werden.

4 https://ec.europa.eu/digital-agenda/en/open-standards
5 https://www.bsi.bund.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_071.pdf?__blob=publicationFile
6 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/06_APP_Anwendungen/APP_1_2_Webbrowser_Edition_2021.html
7 https://www.bsi.bund.de/DE/Themen/Oeffentliche-Verwaltung/Mindeststandards/Web-Browser/Web-Browser_node.html
8 https://gs.statcounter.com/
9 https://www.w3counter.com/globalstats.php
10 https://www.akamai.com/de/de/resources/visualizing-akamai/

Informationen, welche Verfahren beachtet werden sollten und wie eine sichere Entwicklung realisiert werden kann, kann den Empfehlungen des Open Web Application Security Project (OWASP)[11] aus dem Secure Coding Practices-Quick Reference Guide[12] entnommen werden.

3.3 Absicherung des Webservers

Zur sicheren Bereitstellung von JavaScript auf Webservern ist die Umsetzung der umfangreichen Empfehlungen des OWASP von zentraler Bedeutung. Dabei ist die Top-10-Liste der größten Sicherheitsrisiken webbasierter Anwendungen[13] in der jeweils aktuellen Fassung zu beachten, denen mit geeigneten Maßnahmen dauerhaft begegnet werden muss.

3.3.1 Cross-Site-Scripting und Cross-Site-Request-Forgery

Beim Einsatz von JavaScript sind insbesondere verschiedene Vorkehrungen zur Vermeidung von Cross-Site-Scripting (XSS)[14,15] und Cross-Site-Request-Forgery (CSRF)[16] zu treffen. Die Wirksamkeit der Maßnahmen sollte regelmäßig durch die probeweise Anwendung aktueller Angriffstechniken, wie sie z. B. im XSS Filter Evasion Cheat Sheet[17] beschrieben werden, im Rahmen von Penetrationstests getestet werden.

3.3.2 Unsicherer Einsatz von JavaScript und Content Security Policy (CSP)

Des Weiteren sollte der JavaScript-Code ausschließlich in separaten Dateien und nicht direkt im HTML-Code platziert werden. Darüber hinaus sollte auf die Nutzung von „eval()“ verzichtet werden, da diese Funktion JavaScript-Code interpretiert und ausführt.
Nur dann kann ein geeigneter CSP-Header insbesondere ohne die sicherheitskritischen Attribute „unsafe-inline“ und „unsafe-eval“ gesetzt werden.

3.3.3 Asynchrone Datenübertragung mittels JavaScript

Mit der Verwendung von asynchronem JavaScript (Asynchronous JavaScript and XML, AJAX)[18] sind weitere Risiken verbunden, denen ebenfalls mit spezifischen Maßnahmen begegnet werden muss. Präventive Maßnahmen für die Client- und Server-seitige Absicherung von AJAX-Code liefern hierbei die AJAX Security Cheat Sheet[19] des OWASP.

Weitere BSI-Empfehlungen für die Bereitstellung von Diensten im Internet bzw. Web-Angeboten geben der IT-Grundschutz[20,21] sowie die ISi-Reihe (ISi-Web-Server)[22]. Insbesondere sollte die regelmäßige und kurzfristige Installation von Sicherheitsaktualisierungen für sämtliche verwendeten Produkte, Techniken und Bibliotheken umgesetzt werden.

3.3.4 Cookies

Zur Identifikation der Besucher einer Webseite mit aktiven Inhalten werden häufig Cookies verwendet. Cookies stellen damit für Angreifer ein besonders interessantes Ziel dar, da mit einem Cookie die Identität eines Besuchers komplett übernommen werden kann.

11 https://owasp.org/about/
12 https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/migrated_content
13 https://owasp.org/www-project-top-ten/
14 https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
15 https://cheatsheetseries.owasp.org/cheatsheets/DOM_based_XSS_Prevention_Cheat_Sheet.html
16 https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
17 https://owasp.org/www-community/xss-filter-evasion-cheatsheet
18 https://developer.mozilla.org/de/docs/Web/Guide/AJAX
19 https://cheatsheetseries.owasp.org/cheatsheets/AJAX_Security_Cheat_Sheet.html
20 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/06_APP_Anwendungen/APP_3_1_Webanwendungen_Edition_2021.pdf?__blob=publicationFile&v=2
21 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Kompendium_Einzel_PDFs_2021/06_APP_Anwendungen/APP_3_2_Webserver_Edition_2021.pdf?__blob=publicationFile&v=2
22 https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/ISI-Reihe/isi-reihe_node.html#doc453736bodyText4

Anbieter sollten daher die von Web-Browsern für Cookies verfügbaren Schutzmechanismen nutzen und insbesondere die Attribute „Secure“, „HttpOnly“ und „SameSite=Strict“ verwenden, sofern dies für die Anwendung kein Hindernis darstellt. Es sei hier noch einmal darauf hingewiesen, dass aktive Inhalte auf ein Minimum begrenzt und somit auch auf den Einsatz von Cookies weitestgehend verzichtet werden sollte.

Weitere Informationen zur Sicherheit von Cookies und dem Sitzungsmanagement können aus einem entsprechenden Dokument[23] der OWASP bezogen werden.

3.3.5 Sichere Datenübertragung

Zum Schutz vor diversen Angriffen, sollten Web-Inhalte heutzutage ausschließlich verschlüsselt angeboten werden. Im Zweifel sollte eine automatische Weiterleitung des Web-Browsers von einer ungesicherten Verbindung zu einer verschlüsselten Verbindung erfolgen. Zur weiteren Absicherung sollten Anbieter HTTP Strict Transport Security (HSTS)[24] einsetzen.

Darüber hinaus sollte der Anbieter ein gültiges Zertifikat nutzen, das von einer allgemein anerkannten vertrauenswürdigen Stelle signiert worden ist. Die Bereitstellung von Inhalten über eine wie oben beschriebene gesicherte Verbindung sollte den Anforderungen der Technischen Richtlinie BSI TR-02102-2 „Verwendung von Transport Layer Security (TLS)“[25] genügen.

Zur Überprüfung der Verschlüsselung eines Web-Servers und damit verbundener häufiger Angriffe, kann auf Angebote mit automatisierten Tests[26] zurückgegriffen werden.

3.4 Bibliotheken

Bei der Einbindung externer JavaScript-Bibliotheken, wie z. B. jQuery[27], ist sicherzustellen, dass externe Scripte stets aus einer vertrauenswürdigen Quelle über eine sichere Verbindung geladen werden. Dabei ist zusätzlich mit der Hilfe der Subresource Integrity[28] sicherzustellen, dass Veränderungen an der eingebundenen Bibliothek im Browser des Benutzers nicht ausgeführt werden. Kann dagegen die Vertrauenswürdigkeit der Quelle nicht zweifelsfrei und dauerhaft garantiert werden, müssen solche Bibliotheken vollständig vom Anbieter bezogen, abhängig vom individuellen Schutzbedarf der eigenen Web-Anwendung hinreichend auditiert und anschließend auf dem unter eigener Kontrolle stehenden Web-Server selbst gehostet werden. Die Verfügbarkeit der Bibliotheken unter einer freien Lizenz[29] versetzt den Anbieter prinzipiell in die Lage, diese Anforderungen geeignet umzusetzen. Bei Lösungen unter anderen Lizenzen müssen ggf. vorab gesonderte Vereinbarungen mit dem jeweiligen Hersteller getroffen werden.

Es sollten Maßnahmen etabliert werden, mit denen der Anbieter über eventuell auftretende Schwachstellen von im Code genutzten Funktionen informiert wird.

- Es sollten regelmäßige Qualitätssicherungen/Audits des Codes der externen Bibliotheken durchgeführt werden.
- Zusätzlich sollten gängige Quellen wie z. B. die Seiten der jeweiligen Hersteller oder auch Mailinglisten abonniert werden, um zeitnah über Sicherheitslücken informiert zu werden.
- Werden Sicherheitslücken bekannt, sollten diese zeitnah mitigiert werden, bis eine Aktualisierung verfügbar ist. Dies kann z. B. dadurch erreicht werden, dass die Auslieferung

23 https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
24 https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
25 https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.pdf?__blob=publicationFile
26 https://ssllabs.com/ssltest/
27 https://jquery.com/
28 https://developer.mozilla.org/en-US/docs/Web/Security/Subresource_Integrity
29 http://opensource.org/licenses

aktiver Inhalte solange unterbunden wird, bis die Sicherheitslücken geschlossen sind.

- Sicherheitsaktualisierungen sind umgehend auf ihre Kompatibilität mit der eigenen Web-Anwendung zu testen und dann zu übernehmen.

Die Aktualisierung der Bibliotheken ist immer dann selbst sicherzustellen, wenn mit lokalen Instanzen der Bibliotheken auf dem eigenen Web-Server gearbeitet wird.

4 Fazit

Unter den mit dieser BSI-Veröffentlichung beschriebenen Randbedingungen und bei Umsetzung der sicherheitstechnisch erforderlichen Maßnahmen bleibt das Risiko der Bereitstellung von aktiven Inhalten mit einem normalen Schutzbedarf beherrschbar. Technisch sollten nur verbreitete aktive Webtechniken, deren Unterstützung bereits in modernen Browsern in Verbindung mit geeigneten Sicherheitsmaßnahmen integriert ist, genutzt werden. Auf Grundlage der obigen Ausführungen wird geschlussfolgert, dass dies aktuell bei Einsatz von HTML5-Elementen und JavaScript der Fall ist.

Die Ausführung von aktiven Inhalten kann zur Sicherstellung der Barrierefreiheit und durch Filter an Sicherheitsgateways oder Netzübergängen sowie durch den Nutzer selbst blockiert werden. Daher ist stets die vollständige Nutzbarkeit der über die Webseite angebotenen Inhalte und Dienstleistungen auch ohne die Verwendung aktiver Inhalte zu ermöglichen.

5 Weitere Quellen/Hilfestellungen

Von der Allianz für Cyber-Sicherheit veröffentlichte Absicherungsmöglichkeiten beim Einsatz von Web-Browsern[30] sind in der Lage, abhängig vom individuellen Schutzbedarf auf Clientseite, eine sicherheitstechnisch hinreichende Ausführungsumgebung für einen Browser unter Nutzung aktiver Inhalte bereitzustellen. Zudem sind die Empfehlungen des IT-Grundschutz[31] (hier seien besonders die Bausteine APP.1.2 (Web-Browser) und Bausteine aus dem Bereich SYS (IT-Systeme, hier z. B. Clients unter Windows/Linux/iOS) genannt) und der ISi-Reihe des BSI[32], dort insbesondere zur sicheren Nutzung von Web-Angeboten (ISi-Web-Client)[33], zu beachten.

Mit den BSI-Veröffentlichungen publiziert das Bundesamt für Sicherheit in der Informationstechnik (BSI) Dokumente zu aktuellen Themen der Cyber-Sicherheit. Kommentare und Hinweise können von Lesern an **info@cyber-allianz.de** gesendet werden.

30 https://www.bsi.bund.de/SharedDocs/Downloads/Webs/ACS/DE/BSI-CS/BSI-CS_047.pdf?__blob=publicationFile
31 https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html
32 https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/ISI-Reihe/isi-reihe_node.html
33 https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/ISI-Reihe/isi-reihe_node.html#doc453736bodyText7