Kai Jendrian

Web-Schutz-Baukasten für Firefox

Sicheres Surfen trotz aktiver Inhalte

Dieselbe Technik, die durch die Ausführung so genannter „aktiver Inhalte“ im Web-Browser attraktive, leistungsfähige und interaktive Web-Anwendungen ermöglicht, eröffnet auch Schadsoftware den Zugriff auf das System des Nutzers. Der Beitrag stellt einige Möglichkeiten vor, wie man sich beim Surfen mit dem Mozilla Firefox vor zahlreichen Angriffen schützen kann.

1 Im Web-Dschungel

Die Web-Welt wird immer bunter und (inter-)aktiver. Getrieben durch die Forderung nach mehr Dynamik und Interaktivität haben sich die Browser von reinen Werkzeugen zur Darstellung von (statischem) HTML hin zu uneingeschränkt programmierbaren Werkzeugen entwickelt.

Moderne Web-Anwendungen versuchen, den Leistungsumfang von eigenständigen Programmen auf dem eigenen PC auf Anwendungen im Internet zu übertragen. Dabei wird reichlich von unterschiedlichen Skriptsprachen Gebrauch gemacht. So werden eingebaute Möglichkeiten wie Javascript verwendet, aber auch Skripting-Lösungen von Drittanbietern, wie z. B. Adobe Flash oder Microsoft Silverlight eingebunden.

Die Flexibilität moderner Browser wird durch eine hohe Komplexität und damit verbunden auch vielfältigen Angriffsmöglichkeiten erkauft. Auch wenn für die verschiedenen Browser eingebaute grundlegende Sicherheitsmechanismen existieren,[1] sorgen diese Mechanismen durch einen kompletten Verzicht auf aktive Inhalte für einen Schutz vor Angriffen wie Cross-Site-Scripting[2] (XSS), der Ausspähung von Daten oder Cross-Site-Request-Forgery[3] (CSRF).

Ein Ansatz, der eine sichere Grundkonfiguration der Browser – nach dem Prinzip „Security by default“ – ermöglicht und dem Anwender die Nutzung aktiver Inhalte für einige, vom Anwender als vertrauenswürdig eingestufte Websites erlaubt, existiert heute mit Bordmitteln für keinen Browser.

Im Folgenden wird eine Auswahl von Add-Ons für den Mozilla-Browser Firefox vorgestellt, mit der sich eine sichere Grundkonfiguration erreichen lässt, ohne grundsätzlich auf die Möglichkeiten moderner Web-Anwendungen verzichten zu müssen.

Kai Jendrian

Security Consultant bei der Secorvo Security Consulting GmbH, lizenzierter Auditor und OWASP-Mitglied.
Beratungsschwerpunkte: Information Security Management und Anwendungssicherheit.
E-Mail: kai.jendrian@secorvo.de

1 Siehe: Google Browser Security Handbook

2 Konkrete Angriffe

Die Möglichkeiten aktiver Inhalte (z. B. Flash oder Javascript) erlauben einem Angreifer eine umfangreiche Kontrolle des Browsers und sogar des Computers seines Opfers. Dazu muss es nur gelingen, das Opfer dazu zu bringen, aktive Inhalte im Kontext einer Website auszuführen, der es vertraut.

Hierzu werden beispielsweise die bereits erwähnten XSS- oder CSRF-Angriffe genutzt. Aber auch durch eine Kompromittierung von Websites mit guter Reputation und der dortigen Einbettung von aktiven Inhalten, z. B. durch die Nutzung unsichtbarer Rahmen (so genannter „IFrames“), erreicht ein Angreifer sein Ziel.

Neben so unspektakulären, aber umso schwerwiegenderen Angriffen wie dem Stehlen von Cookies, die dem Angreifer bei vielen Web-Anwendungen den vollen Zugriff im Kontext des angemeldeten Benutzers erlauben, gibt es eindrucksvolle veröffentlichte Beispiele wie das „Browser Exploitation Framework (BeEF)“[4], den „XSS Proxy“[5] und den „Javascript Port Scanner“[6], die verdeutlichen, welche Kontrolle ein Angreifer ausüben kann.

3 Schutzmaßnahmen

Vor allen anderen Maßnahmen ist es für sicheres Surfen im Web unabdingbar, dass immer alle Software auf dem aktuellen Stand ist. Nicht nur die Browser und Betriebssysteme haben Schwachstellen, die von Angreifern ausgenutzt werden können, sondern gerade auch die Plugins zur Darstellung alternativer Inhalte (z. B. Flash oder PDF) weisen immer wieder schwer wiegende Sicherheitsprobleme auf. Moderne Angriffe durch bösartige Websites werten die vom Browser übermittelten Informationen über die Umgebung des Benutzers aus, um dem Opfer eine möglichst passgenaue Schadsoftware zur Kompromittierung unterzuschieben.

Um einigermaßen komfortabel mit vertrauenswürdigen Websites arbeiten zu können, ohne gleich der ersten Attacke

2 Siehe http://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

3 Siehe http://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29

4 Siehe http://www.bindshell.net/tools/beef/

5 Siehe http://xss-proxy.sourceforge.net/

6 Siehe http://www.gnucitizen.org/blog/javascript-port-scanner/

Abb. 1 | NoScript

zum Opfer zu fallen, sollte ein Browser zudem in der Grundkonfiguration sicher sein und dem Benutzer die bewusste Entscheidung über die Aufweichung der sicheren Grundkonfiguration für vertrauenswürdige Seiten ermöglichen. Konkret bedeutet das, dass

- eine unkontrollierte Einbettung nicht vertrauenswürdiger Inhalte verhindert wird,
- nicht vertrauenswürdiger Code nicht ausgeführt wird und
- die unerwünschte Speicherung oder Übermittlung privater Daten unterbunden wird.

Die integrierten Sicherheitsmechanismen der Browser sind allerdings ungeeignet, da die Sperrung aller aktiven Inhalte und die pauschale Blockierung von Cookies die Nutzung sehr vieler Webseiten und praktisch aller Web-Anwendungen ausschließen – und das Web so unbrauchbar machen.

4 Sicherheit in Firefox

Mozilla Firefox ermöglicht durch die Nutzung geeigneter Add-Ons, den Browser nach dem Prinzip „Secure by default" zu schützen. Jeder Benutzer kann dabei granular entscheiden, welchen Inhalten er vertraut und welche Risiken er einzugehen bereit ist.

Diese Strategie verlangt vom Nutzer ein tiefer gehendes Verständnis der Hintergründe von Web-Anwendungen. In der Einarbeitungsphase werden viele Web-Anwendungen auf Anhieb nicht wie gewohnt funktionieren, bis die Werkzeuge mit der Zeit so konfiguriert sind, dass sie dem Surf-Verhalten des Nutzers entsprechen und dieses nicht mehr signifikant behindern.

Leider ist es bisher noch nicht so einfach, die Add-Ons in Unternehmensumgebungen zentral zu verteilen und einzurichten. Trotzdem stellen die vorgestellten Zusatzmodule bei entsprechender Bereitschaft zur Einarbeitung einen echten Sicherheitsgewinn dar, auch wenn der Mozilla-Browser selbst Schwachstellen aufweisen sollte.

Die im Folgenden vorgestellten Module können einzeln oder im Zusammenspiel das Sicherheitsniveau beim Surfen signifikant erhöhen, wenn sie korrekt und gewissenhaft genutzt werden. Sie zielen auf ein vom Benutzer gesteuertes „abgestuftes Schutzkonzept", das mit dem Vertrauen in eine Webseite wachsen kann und zugleich die verbleibenden Risiken und das Schadensausmaß minimiert.

4.1 PwdHash

Der nachlässige Umgang mit Passwörtern im Internet bietet Angreifern eine große Angriffsfläche. Entweder gelingt es einem Angreifer, sein Opfer dazu zu bewegen, seine Zugangsdaten in einer Website einzugeben, die zwar aussieht wie die eigentliche Seite, aber nur dazu dient, diese Daten auszuspähen (Phishing[7]), oder aber ein Angreifer bekommt Zugriff auf Zugangsdaten, die bei einem erfolgreichen Angriff auf eine Website entwendet wurden, um diese systematisch bei verschiedenen anderen Sites auszuprobieren. Die Erfolgswahrscheinlichkeit für einen Angriff der zweiten Art ist relativ hoch, da viele Benutzer die gleichen Zugangsdaten für unterschiedliche Web-Accounts verwenden.

Zum Schutz gegen beide Angriffsarten verknüpft die Firefox-Erweiterung PwdHash[8] auf Wunsch des Nutzers ein Masterpasswort mit dem Domain-Namen der Website, an die das Passwort übermittelt werden soll, via Hashfunktion zu einem quasi zufälligen Passwort. Dadurch wird für jede Website ein individueller Zugangsschutz eingerichtet, so dass der Benutzer gut gegen Angriff der zweiten Art geschützt ist. Aber auch gegen Angriffe der ersten Art bietet PwdHash einen guten Schutz, denn wenn der Domain-Name der besuchten Webseite nicht stimmt, stimmt auch das dem Phisher übermittelte Passwort nicht.

Die Wahl eines ausreichend langen Masterpassworts ist auch bei der Verwendung von PwdHash zwingend, damit es einem Angreifer nicht möglich ist, aus einem ausgespähten Passwort durch gutes Raten oder Ausprobieren das verwendete Masterpasswort zu ermitteln.

4.2 NoScript

Die meisten gefährlichen Angriffe gegen Websurfer werden durch aktive Inhalte im Browser ausgeführt. Die Firefox-Erweiterung NoScript[9] sperrt Skriptsprachen und erlaubt es dem Benutzer, die Ausführung aktiver Inhalte gezielt frei zu schalten. Dabei kann jede Quelle auch nur temporär frei geschaltet oder auch stufenweise blockiert werden – die Konfigurationsmöglichkeiten von NoScript sind sehr umfangreich.

Anfangs wird NoScript leicht als arbeitsbehindernd wahrgenommen, denn es ist häufig dafür verantwortlich, dass Webseiten nicht wie erwartet funktionieren. Aber schon nach kurzer Zeit sind alle häufig genutzten und als vertrauenswürdig angesehenen Webseiten frei geschaltet, und NoScript meldet sich nur noch, wenn neue, unbekannte Seiten besucht werden.

4.3 Request Policy

Viele Websites bedienen sich bei anderen Websites mit Inhalten, um diese in die eigenen Seiten einzubetten. Diese Vermaschung von Websites ermöglicht erst viele moderne Techniken, wie sie z. B. im Bereich der sozialen Netzwerke Anwendung finden. Allerdings bietet gerade diese Vermaschung Angreifern die Möglichkeit, schädliche Inhalte in scheinbar vertrauenswürdige Webseiten einzubetten. Dass kann durch kompromittierte Webseiten geschehen (z. B. durch Nutzung von IFrames), aber auch durch „bösartiger" Inhalt von Werbeeinblendungen, die in vertrauenswürdige Seiten eingebunden sind. Auch die verschiedenen Dienstleister, die

7 Siehe Fox, Gateway, DuD 6/2005, S. 365.
8 Siehe http://www.pwdhash.com
9 Siehe http://www.noscript.net

das Nutzerverhalten von Websurfern analysieren, nutzen die Einbettung fremder Inhalte zur Übermittlung von Nutzerdaten an den jeweiligen Dienst.

Mit der Firefox-Erweiterung RequestPolicy[10] hat der Nutzer die Möglichkeit, individuell zu steuern, ob und welche fremden Inhalte von einer Webseite eingebunden werden dürfen. RequestPolicy arbeitet zudem mit Regeln, die festlegen, wie solche Inhalte eingebunden werden. Dabei können entweder ganze Websites als Quelle oder Ziel, aber auch einzelne Quell-/Zielkombinationen frei geschaltet werden.

Wie NoScript erfordert auch die Arbeit mit RequestPolicy sowohl ein gewisses Verständnis von den Abläufen beim Surfen im Internet als auch ein gewisses Durchhaltevermögen in der Anfangsphase, bis die Regelbasis an das eigene Surfverhalten und die Vertrauensbewertungen angepasst ist.

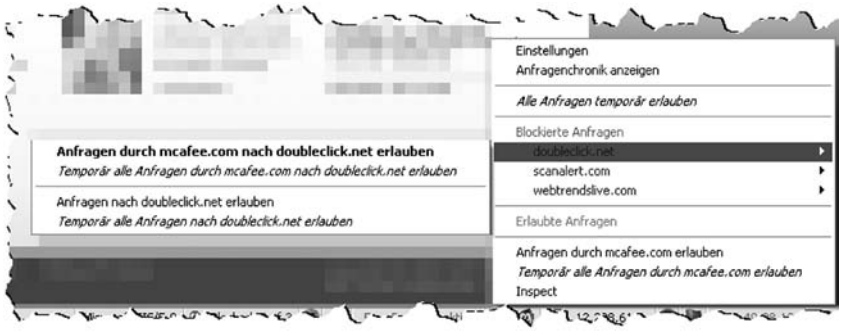
4.4 Cookie Manager – CS Lite

HTTP ist ein zustandsloses Übermittlungsprotokoll. Das heisst, dass eine Webanwendung keine im Protokoll implementierten Mechanismen zur Verfolgung von Anwendungsabläufen und der Zuordnung von Anfragen nutzen kann. Daher wird in vielen Anwendungen mit Zusatzdaten gearbeitet, die diese Zuordnung ermöglichen. Hierbei spricht man von Session-Informationen. Alle gängigen Browser bieten dafür so genannte Cookies[11] an. Web-Anwendungen schicken bestimmte Daten zur Speicherung an den Browser. Dieser legt diese in Cookies ab und übermittelt die Cookies bei jedem Seitenaufruf wieder zurück an die jeweilige Webanwendung.

Auch Cookies können für die harmlose Speicherung von Session-Informationen, aber auch zur Nachverfolgung der Aktivitäten von Benutzern eingesetzt werden. Daher ist es auch für Cookies wünschenswert, einen einfachen Mechanismus zur Verfügung zu haben, mit dem gezielt gesteuert werden kann, welche Cookies erlaubt und welche blockiert werden sollen.

Die beiden Firefox-Erweiterungen CookieSafe[12] und CS Lite[13] ermöglichen eine solche granulare Kontrolle über die Nutzung von Cookies beim Surfen.

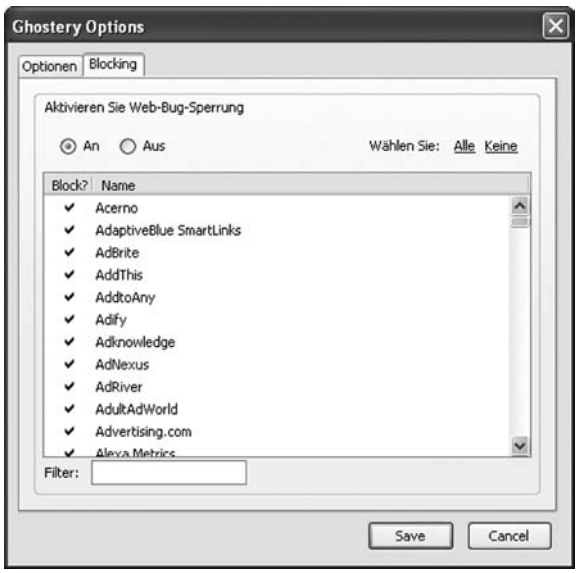
Abb. 2 | Request Policy



4.5 Better Privacy

Unbekannter als die Verwendung von Cookies ist die Verwendung von Local Shared Objects[14] durch Adobe Flash oder DOM Storage[15] Mechanismen. Damit können Informationen gespeichert werden, die mit den Standardwerkzeugen der gängigen Browser nicht verwaltet werden können. Die Firefox-Erweiterung BetterPrivacy[16] bietet ein integriertes Werkzeug zur Kontrolle und Verwaltung solcher so genannter Super-Cookies.

Abb. 3 | Ghostery



4.6 Ghostery

Das Ausspähen des Nutzerverhaltens durch verschiedenste Tracking-Sites ist aus der Marketing-Perspektive für viele Dienstanbieter sehr verlockend. Aus Benutzersicht ist die dauerhafte Überwachung durch Dienste wie Google-Analytics hingegen nicht immer erwünscht.[17] Während einige dieser Dienste, wie z. B. etracker, dem Nutzer technisch eine Datenschutz konforme Widerspruchsmöglichkeit einräumen, gilt das nicht für alle Dienstleister.

Daher bietet die Firefox-Erweiterung Ghostery[18] allen auf ihre Privatsphäre be-

Abb. 4 | Cookie Manager

10 Siehe http://www.requestpolicy.com
11 Siehe Bizer, Gateway, DuD 10/2003, S. 644.
12 Siehe https://addons.mozilla.org/en-US/firefox/addon/2497
13 Siehe https://addons.mozilla.org/en-US/firefox/addon/5207
14 Siehe http://www.adobe.com/products/flashplayer/articles/lso/
15 Siehe http://en.wikipedia.org/wiki/DOM_storage
16 Siehe http://netticat.ath.cx/BetterPrivacy/BetterPrivacy.htm
17 Siehe z. B. Hansen, Google Analytics auf dem Prüfstand, DuD 8/2008, S. 506.
18 Siehe http://www.ghostery.com

Abb. 5 | Web of Trust (WoT)

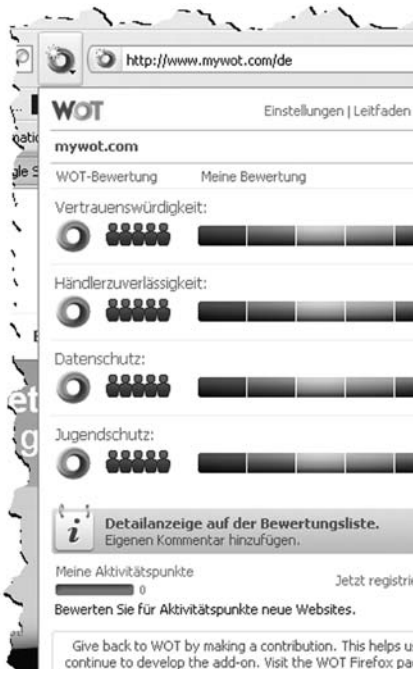
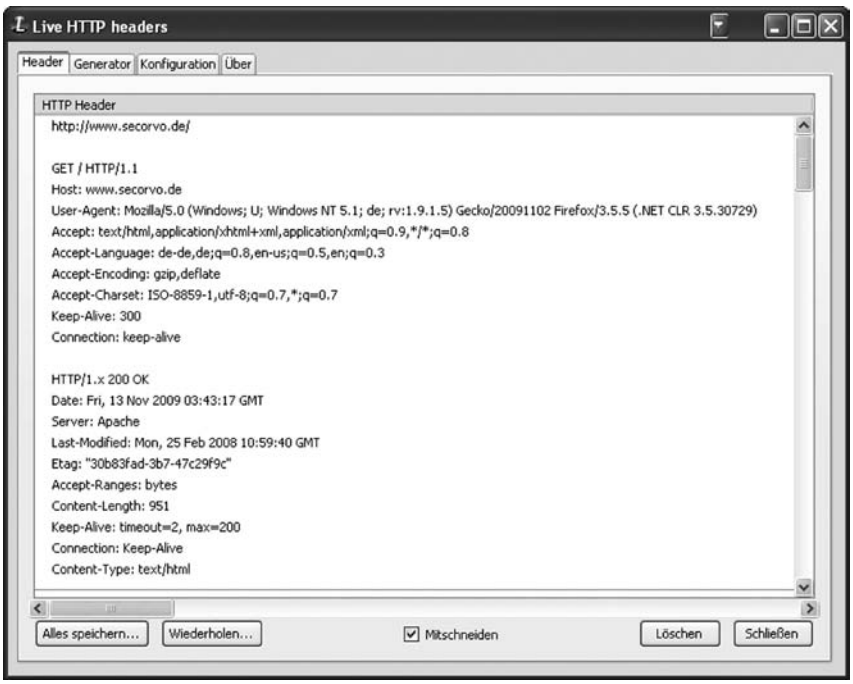


Abb. 6 | Live HTTP Headers



dachten Benutzern die Möglichkeit, die Nutzung von Tracking-Dienstleistern gezielt zu blockieren oder zuzulassen.

4.7 Long URL Please

Vertrauen in die Seiten, die aufgerufen werden sollen, spielt beim Surfen im Web eine große Rolle. Mit der aktuellen Verbreitung der mobilen Nutzung des Internes greift ein Trend zur Verkürzung von URLs (durch Dienstleister wie bit.ly o. ä.) um sich. Dienste wie Twitter machen ausgiebigen Gebrauch von dieser Technologie. Aus Sicherheitssicht sind verkürzte Links ein „Alptraum", da nicht einmal an der URL zu erkennen ist, welches das eigentliche Ziel eines solchen Links ist.

Eine gefahrlose Auflösung solcher Links ermöglicht das Add-On LongURLPlease[19]. Hierdurch werden in Firefox verkürzte Links von z. Z. 75 Dienstleistern aufgelöst und komplett im Browser dargestellt, so dass ein Nutzer sich vor dem Aufruf dieser Links entscheiden kann, ob er dem Ziel vertraut oder nicht.

4.8 Web of Trust

Vielen der vorgestellten Ansätze liegt die Annahme zu Grunde, dass ein Benutzer die Vertrauenswürdigkeit einer Website gut einschätzen kann. Diese Annahme ist aber in der Praxis selten realistisch. Daher existieren verschiedene Ansätze, um einem Benutzer die Einschätzung über die von ihm besuchte Seite zu erleichtern.

Die Firefox-Erweiterung Web of Trust[20] (WOT) verarbeitet dazu die gesammelten Einschätzungen der Community und visualisiert eine Einschätzung der Vertrauenswürdigkeit durch ein Symbol in der Navigations-Symbolleiste von Firefox. Durch Klicken auf das Symbol stellt die Erweiterung eine detaillierte Analyse der Einschätzung dar. Registrierte Benutzer können hierüber auch ihre eigene Einschätzung dem Erfahrungsschatz der Community hinzufügen.

4.9 LiveHTTPHeaders

Mehrfach war zuvor die Rede davon, dass eine tiefer gehende Kenntnis der Vorgänge beim Browsen zur besseren Kontrolle der Sicherheit hilfreich sind. Wer sich diese Kenntnis verschaffen möchte, dem leistet die Firefox-Erweiterung LiveHTTPHeaders[21] gute Dienste. Mit ihr kann man dem Datenaustausch zwischen Browser und Website „auf die Finger schauen" und dabei verstehen, wie genau das Internet funktioniert.

5 Fazit

Die vorgestellten Mechanismen erlauben es, das Surfen im Web mit dem Mozilla-Browser Firefox sicherer zu gestalten. Leider ist die Einbindung der vorgestellten Erweiterungen heute noch Handarbeit und bedarf einer aufwändigen, nicht immer einfachen und intuitiven Konfiguration. Demjenigen, der sich davon nicht abschrecken lässt, ist damit schon heute gut geholfen. Für alle anderen Nutzer bleibt zu wünschen, dass die einzelnen Ansätze ihren Weg in den Browser finden und sich so zentral konfigurieren lassen, dass auch in Unternehmen ein kontrollierter Einsatz solcher Schutzkonzepte möglich wird.

Ein Restrisiko bleibt allerdings auch beim Einsatz der vorgenannten Add-Ons: Gibt der Nutzer – sei es aus Nachlässigkeit, sei es aufgrund einer Fehleinschätzung – einer mit Schadsoftware verseuchten Webseite den Zugriff frei, verhindert keines der Add-Ons den Angriff. Vor dem Nutzer selbst kann keine Zusatzsoftware schützen. In diesem Fall bleibt nur noch der Rückgriff auf eine Surf-CD[22]: Beim nächsten Booten ist alle Malware vom System gelöscht.

19 Siehe http://www.longurlplease.com/

20 Siehe http://www.mywot.com

21 Siehe http://livehttpheaders.mozdev.org/

22 Beispielsweise die Surf-CD des Bundesamtes für Sicherheit in der Informationstechnik (BSI) https://www.bsi.bund.de/DE/Themen/ProdukteTools/SecuritySurfCD/securitysurfcd_node.html oder die der Fachzeitschrift c't („Bankix"): http://www.heise.de/software/download/ct_bankix/57557.