Web 2.0 Sicherheit - Herausforderungen und Trends

Steffen Ullrich, GeNUA mbH, Padiofire Projekt

1 Abstract

Das heutige Web ist geprägt von interaktiven Applikationen mit nutzerspezifischen und nutzergenerierten Inhalten. Es wird sowohl privat wie auch bei der Arbeit, sowohl zu Hause, wie auch unterwegs und am Arbeitsplatz genutzt. Zunehmend vertrauen wir privat und geschäftlich darauf, dass im Web sensible Daten sicher aufgehoben sind und die Dienste rund um die Uhr erreichbar sind.

Mit steigendem Wert der im Web befindlichen Daten und Identitäten (Soziale Netzwerke, Mail..) steigt auch die Begehrlichkeit, diese Daten zu stehlen oder zu manipulieren, Inhalte unter falscher Identität zu publizieren oder auch Diensteanbieter und -nutzer durch Angriff auf Verfügbarkeit des Dienstes zu schädigen.

Im weiteren wird untersucht, welche Komponenten im Web 2.0 interagieren. Es werden Entscheidungen bei Design und Implementation von Komponenten und deren Zusammenwirken bzgl. resultierender Sicherheit hinterfragt. Dabei wird ein Augenmerk auf die zunehmende Komplexität der sicherheitsrelevanten Entscheidungen in Firewalls gelegt. Abschließend werden verschiedene Trends in der weiteren Entwicklung des Web 2.0 aufgezeigt, bei denen einige die Sicherheit verbessern, andere weitere Probleme produzieren.

2 Komponenten des Web2.0

Webanwendungen bestehen aus dem Client (i.A. der Browser), welcher über eine Verbindung von Servern Daten erfragt und diese darstellt. In der Realität ist dieses scheinbar sehr einfache Vorgehen deutlich komplizierter - sowohl Client wie auch Server wie auch die Middleware bestehen aus einer Vielzahl von Komponenten, die zueinander in verschiedenen Vertrauensstellungen stehen und komplex interagieren.

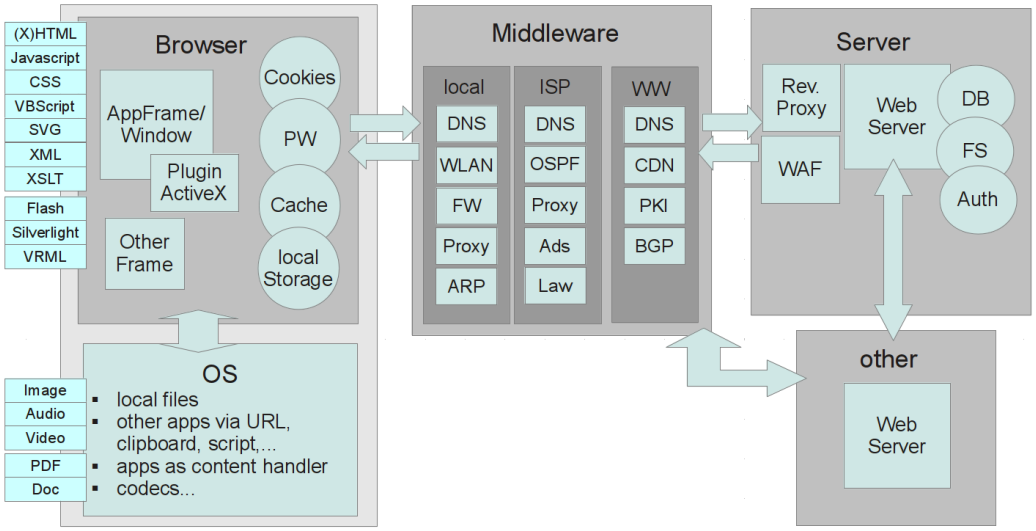

Abbildung 1: Komponenten Web 2.0

Insbesondere können innerhalb eines einzelnen Clients Quellen verschiedener Herkunft und Vertrauensstellung zusammen dargestellt werden, d.h. neben dem primär gewünschten Inhalt bekommt man noch die Anzeigen von Anzeigennetzwerken, Buttons von sozialen Netzwerken und oft unsichtbare Verlinkungen zu Trackingnetzwerken. Oftmals sind diese sekundären Quellen in einer Art eingebunden, dass sie mit den primären Inhalten interagieren können.

3 Probleme bei Funktion und Zusammenwirken der Komponenten

Die Probleme, die in den Komponenten bzw. bei deren Interaktionen auftreten, lassen sich grob in drei Klassen unterteilen.

Zum einen haben wir Fehler im Design von Protokollen und Standards. Hierzu gehören insbesondere Ungenauigkeiten der Spezifikation, die zu Unterschieden der Implementation und damit unterschiedlichem Verhalten führen.

Weiterhin werden die Standards teilweise nicht korrekt umgesetzt. Das ist oft aus dem Druck entstanden, nicht standardkonforme Webseiten besser anzuzeigen als die Konkurrenz. Leider führte das dazu, dass keine Notwendigkeit mehr bestand die Webseiten an die Standards anzupassen, d.h. die Designfehler der Implementation müssen wegen der Abwärtskompatibilität bestehen bleiben.

Und schließlich haben wir das ungerechtfertigte Vertrauen in Funktion von Komponenten und ein mangelndes Verständnis für deren Zusammenspiel. Das betrifft die Verifikation von Ein- und Ausgaben aber umfasst auch das Vertrauen in die Gutartigkeit von offene WLANs, DNS, PKI Strukturen oder ISPs unter staatlicher Fürsorge.

Zusammen ergeben diese Problem ein Vielfalt an Angriffsmöglichkeiten, gegen die aktuelle Browser, Virenscanner oder Firewalls nur unzureichend schützen.

4 Analyse in Firewalls und IDS

Die Analyse der Inhalte wird erschwert durch tief verschachtelte Containerstrukturen in den verwendeten Protokollen und Daten, d.h. mehrere HTTP-Requests in einer TCP-Verbindung, eventuell mehrere Chunks im HTTP-Body, komprimierte Bodies, Multipart-Daten (MHTML, JAR...), in HTML eingebettetes Javascript, CSS oder data-URLs usw. .

Dazu kommt noch fehlender Kontext, d.h. bei script und style interpretiert der Browser die Daten unabhängig vom übermittelten Content-type, ähnlich ist es mit der Ermittlung des Charsets. Auch sieht der Firewall keine Inhalte, die bereits im Browsercache liegen oder über andere Wege übertragen wurden. Probleme gibt es ebenfalls wenn die Daten stückweise über mehrere Requests geholt werden. Dem Firewall fehlen in diesen Fällen wichtige Informationen zur Risikoeinschätzung.

Per HTTPS übertragene Daten zu analysieren ist für ein IDS unmöglich, Application Level Gateways wiederum können das nur unter Veränderung der Zertifikate, was zu Problemen mit der Zertifikatverifikation führt.

Eine zuverlässige und realistische Risikoabschätzung ist unter diesen Bedingungen in real-time derzeit unmöglich. Erfolgversprechender scheinen Normalisierung der Daten und White- bzw. Blacklisting sein. Allerdings muss bei letzteren der Administrator das Gefahrenpotential selber einschätzen oder sich auf externe Listen verlassen.

Im vom BMBF geförderten Projekt Padiofire verfolgen wir daher Ideen für eine tiefgehende skalierbare Analyse, gekoppelt mit einer überwiegend automatischen Risikoabschätzung.

5 Trends

Die führenden Browserhersteller haben die Problematik erkannt und arbeiten an einer Verbesserung der Situation. Allerdings führt der Zwang zur Abwärtskompatibilität oft nur zu (durchaus nützlichen) Workarounds wie Blacklists und XSS-Filter, der Wunsch zum Bruch mit fehlerhaften Designentscheidungen muss i.A. von jedem Webserver explizit bekanntgegeben werden.

Der lebende Standard HTML5 beseitigt einige Ungenauigkeiten der Vorgänger und führt sicherheitsrelevante Ergänzungen wie Sandboxing und Content-Security-Policy ein, bietet auf der anderen Seite mit Websockets und Local Storage aber zusätzliche Angriffspunkte.

Auf Serverseite sind weniger klare Bemühungen um Sicherheit zu erkennen. Hier muss derzeit wohl primär das Bewusstsein von Entwicklung und Management für Sicherheit deutlicher ausgeprägt werden, damit die verfügbaren Sicherheitsmöglichkeiten auch angewandt werden.

Letztendlich ist es fraglich, ob die Sicherheitsbemühungen ausreichen, um die Angriffsfläche spürbar zu verkleinern, da ja gleichzeitig die Nutzung des Web zunimmt.