



Entwicklung einer Web 2.0-Firewall

Parallelisierte Anwendungserkennung in Overlay-Netzen für Firewalls

Autor: Alexander von Gernler, GeNUA mbH

Abstract:

Das Web 2.0 und die damit verbundenen Protokolle und Frameworks, wie JavaScript, AJAX, JSON, XML-RPC, Google API, sind heute kaum noch aus modernen Webseiten wegzudenken. Durch die Web 2.0-Technologie steht den Seiten ein mächtiger Satz an Werkzeugen zur grafischen Gestaltung und zur Bereitstellung von datenbankgestützten Inhalten zur Verfügung, der bisher allerdings kaum kontrolliert werden kann.

Im Rahmen des vom Bundesministerium für Bildung und Forschung geförderten Sicherheitsforschungsprogramms läuft seit Juli 2011 das Verbundprojekt PADIOFIRE, das sich mit der Absicherung und Inspektion von Web 2.0-Verkehr beim Passieren von Firewalls beschäftigt.

Problemstellung

Waren Webseiten in den 90er Jahren noch statische Objekte zur reinen Informationsdarstellung, so besteht die Landschaft des heutigen Web 2.0 aus durchwegs bunten, interaktiven Webpräsenzen, die standardmäßig auch eine Bereitstellung von Informationen durch ihre Nutzer vorsehen, etwa mittels Kommentaren oder Uploads.

Diese veränderte Umgebung macht vor dem typischen Bildschirmarbeitsplatz in Organisationen der öffentlichen Hand nicht halt: Auch wenn während der Arbeitszeit im Browser hauptsächlich Anwendungen des eigenen Intranets aufgerufen werden, so ist doch ein kurzer Seitenblick auf eine der Google-Seiten zur Beschaffung weiterer Informationen inzwischen der Normalfall.

Genauso haben Personalere für sich entdeckt, während der Dienstzeit Facebook und andere soziale Netzwerke zum routinemäßigen Abklopfen von Bewerbern zu nutzen. Ganz zu schweigen von der Nutzung von Web 2.0-Seiten während der Mittagspause oder bei kurzen, fachfremden Zugriffen während des Tages:

Der Zugriff auf Web 2.0-Inhalte am Arbeitsplatz ist, wird er nicht durch Filterung einer Firmen-Firewall geblockt, heutzutage üblich.



Facebook: Jetzt auch als Firmenanwendung für Social Marketing und im Personalbereich

Problematisch sind hierbei nicht die vordergründigen Inhalte der aufgerufenen Seiten. Denn in Zeiten, in denen Firmen verstärkt mit Blogs, Twitter-Meldungen und YouTube-Kanälen auf sich aufmerksam machen, ist es teilweise sogar erwünscht, dass Mitarbeiter neben ihrer normalen Tätigkeiten beispielsweise durch Networking die Werbung neuer Kollegen vorantreiben.

Zur Gefahr werden kann aber ein Missbrauch der überaus mächtigen Frameworks, mit denen die diversen Seiten realisiert sind: War beim „alten“ Web noch alles statisch, so lebt das neue, unter dem unscharfen Begriff Web 2.0 zusammengefasste Netz davon, dass auch auf Client-Seite, also im Web-Browser des Kunden oder Interessenten, einiges an Programmlogik abläuft.

Dies kombiniert mit der Fähigkeit, ohne komplettes Neuladen der Seite eine schnelle Kommunikation zwischen Web-Browser und Server darzustellen und einzelne Inhalte schnell nachzuladen, gibt einem Angreifer allerhand Werkzeug an die Hand. Und damit lässt sich schon einiges anstellen: Ausspähen von Informationen aus dem lokalen Verlauf des Browsers, Beobachten von Sitzungen oder Nutzung des Clients als unwissende Plattform für weitere Angriffe. Der Kreativität sind hier keine Grenzen gesetzt.

Möglich wird dies, weil die Darstellung aktiver Inhalte – angefangen bei JavaScript – heute in Browsern unverzichtbar aktiviert ist, denn sonst bleiben die meisten Seiten „dunkel“.



The screenshot shows the homepage of muenchen.de. At the top, there are navigation links for 'Stadtplan', 'MVV', 'Kino', 'Tickets', 'Hotel', 'Webcam', and 'Wetter'. Below this is a main navigation bar with categories like 'Rathaus', 'Veranstaltungen', 'Essen, Trinken', 'Shopping', 'Hotel', 'Sehenswürdigkeiten', 'Freizeit', 'Verkehr', 'Wirtschaft', and 'Themen'. A secondary navigation bar includes 'Stadtpolitik', 'Stadtverwaltung', 'Stadtinfos', 'Themen', 'Lebenslagen', and 'Kontakt'. The main content area features the 'Landeshauptstadt München' logo and a section titled 'Aktuelles'. A red circle highlights a JavaScript error message: 'Ihr Browser unterstützt kein Javascript. Subscribe to RSS headline updates from: Powered by FeedBurner'. Below this, there are two bullet points: 'Veranstaltungen' and 'Social Media'. The right sidebar contains a search bar, a section for 'Ihre Frage an OB Ude' with a 'direktzu' button and a photo of Christian Ude, and other sections like 'Der Münchner Stadtrat', 'Stadtverwaltung', and 'Engagiert Leben'.

Meldung: „Ihr Browser unterstützt kein JavaScript“

Im Sicherheitsbereich, also in Unternehmen mit Schutzbedarf, in Behörden, aber auch in Konzernen, die einem allzu leichten Abfluss von Informationen aus dem internen Netz vorbeugen möchten, gab es bisher nur eine Abhilfe für das beschriebene Problem: Per Richtlinie wurden in den Browsern des Unternehmens, besser aber auf der zentralen Firewall, JavaScript und andere aktive Inhalte so gut wie möglich herausgefiltert. Dieser 99-prozentige Schutz kann aber aufgrund der vielen subtilen Kanäle, die durch verschiedene Kodierungsverfahren, Implementierungsunterschiede, Auslegung von Standards etc. entstehen, zwar hin und wieder umgangen werden, ist aber meist vollkommen ausreichend. Nun lässt es sich zwar sicher, aber nicht mehr besonders angenehm mit dem Netz arbeiten, weil wichtige Seiten entweder sehr abgespeckt oder aber gar nicht mehr funktionieren.

Angestrebtes Ergebnis

Hier wäre es also wünschenswert, dass man nicht nur „den großen Hebel umlegen“ und die Web 2.0-Technologien damit komplett verbannen kann, sondern dass auch eine feingranularere Aufstellung von Richtlinien möglich ist.

So könnte ein Administrator festlegen wollen, dass zwar generell keine Web 2.0-Seiten aufgerufen werden dürfen, aber dass die interaktive Ansicht von Google



Maps z. B. zur Planung von Dienstreisen schon funktionieren soll – auch im ansonsten streng abgesicherten internen Firmennetz.

Nun ist es für das Projekt aber nicht genug, eine herkömmliche URL-Filterung vorzunehmen, etwa in der Hinsicht, dass jeglicher Inhalt von maps.google.de die Firewall einfach passieren darf. Denn erstens möchte man vielleicht nur eine von vielen Seiten auf einem bestimmten Host erlauben. Zweitens sind, gerade durch die von Sicherheitsexperten wie Dan Kaminsky gezeigten Angriffe auf das Internet-Namenssystem, DNS-Täuschungen möglich, so dass ein Angreifer theoretisch für kurze Zeit selbst für eine Firma als maps.google.de erscheinen kann – genug Zeit, um schadhafte Code zu platzieren. Und drittens sind selbst die SSL-Sicherheitszertifikate, die die Zugehörigkeit von bestimmtem Inhalt zu einer Internet-Domain sicherstellen sollen, nach neuerlichen Analysen von Experten wie Moxie Marlinspike und anderen in Verruf geraten.

GeNUGate 7.0 Patchlevel 4
Soviel ist sicher.
Account: genua
Hostname:

System Verbindungen Benutzer Paketfilter Statistik Logging Suche Go

Policy: WWW-Filter-Vscan-Weeder-Extern

<< Allgemein Module Logging Scanner Transferstatus Weeder Filter Optionen >>

Einfaches Weeding

ActiveX
Java
JavaScript
Plugins

Detailliertes Weeding
Detaillierte Weeder-Einstellungen

Weeder
Informationenachricht beim Weeden
Kein Weeding für Domains

- *.maps.google.de
- *.de-de.facebook.com
- *.adobe.com
- *.xing.com/de

Firewall-GUI: Verbotenes JavaScript mit Ausnahmeliste

Lösungsweg

Es ist also klar, dass man sich in Puncto Sicherheit nicht auf „Umverpackungen“ verlassen, sondern die transportierten Inhalte und damit auch die transportierten Web 2.0-Logiken selbst analysieren können möchte. Die transportierten Logiken werden im Rahmen des Projekts übrigens als Overlays betrachtet, also als zusätzliche Transportschichten, die durch Umfunktionieren von Anwendungsschichten (sog. Overlaying) entstehen.



So wird etwa gerade das Hypertext-Transportprotokoll HTTP heute zu deutlich mehr Zwecken verwendet als eben nur zum Transport von Texten mit Markup-Language (HTML). Gerade HTTP 1.1 mit seinen persistenten Verbindungen und HTML5 mit der Erweiterung der Websockets spannen hier den Bogen weit.



Overlaying – vom Protokoll im Protokoll im Protokoll

Um aber trotzdem wieder Kenntnis von echten Inhalten nehmen zu können, muss zunächst eine Entwirrung der vielen geschichteten Transportprotokolle vorgenommen werden. Diese Auftrennung erlaubt erst danach wieder eine semantische Analyse des eigentlich transportierten Inhalts. Zur schnellen Klassifizierung der vorbeirauschenden Inhalte wird eine Anleihe bei einem verwandten Gebiet genommen: Netzwerk-basierte Intrusion Detection Systeme (NIDS) stellen sich vom Prinzip her auch die ganze Zeit lauschend an den Verkehr etwa eines ganzen Netzsegmentes und versuchen, aus den vorbeifliegenden Mustern diejenigen zu extrahieren, die auf Anomalien (Abweichungen vom Normalzustand) oder sogar speziell auf laufende Angriffe innerhalb des Netzwerks hindeuten.

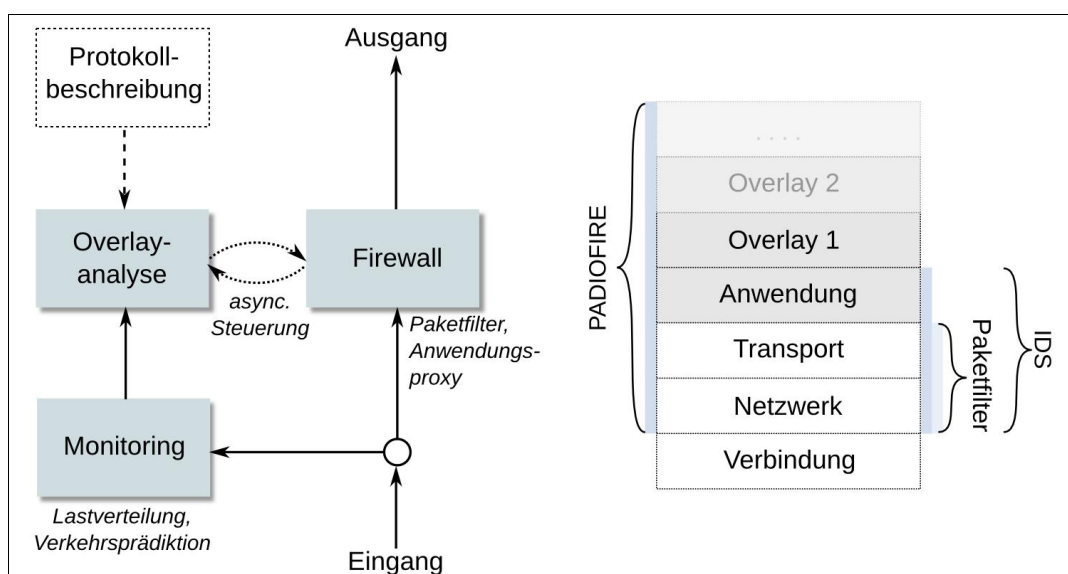
Um die verwendeten NIDS leicht auf das jeweilige Problem anpassen zu können, soll im Projekt PADIOFIRE auch noch eine Sprache entwickelt werden, mit der vom IDS zu erkennende Inhalte der Web 2.0-Kommunikation in Regelsätzen beschrieben werden können. Mit der speziell für die Overlay-Analyse angepassten NIDS-Technologie ist das erste Teilziel von PADIOFIRE erfüllt, nämlich die Overlay- und Anwendungsprotokoll-Analyse durch IDS-basierte Regelsprachen.

Nun ist nur noch das Problem der Leistungsfähigkeit des Gesamtsystems in den Griff zu bekommen: Heutige Netzwerkverbindungen arbeiten im Bereich von einem bis 10 GBit/s. Eine aussagekräftige Analyse der eintreffenden Datenströme muss also von vornherein mit der Anzahl der eingesetzten Rechnersysteme oder mindestens der Prozessorkerne innerhalb eines eingesetzten Systems skalieren, sonst ist das Unterfangen mit steigender Bandbreite schnell unmöglich.



Um die Arbeit sinnvoll auf mehrere Systeme oder Kerne zu verteilen, ist ein strombasierter Ansatz empfehlenswert. Das bedeutet, dass alle Datenpakete, die zu einer einzelnen logischen Verbindung gehören, auch vom selben Analyse-system bearbeitet werden müssen – sonst fehlt wichtiger Kontext für die Analyse.

Das Aufspalten des kompletten Netzwerkverkehrs in separate, den einzelnen Kommunikationsbeziehungen zugeordnete Ströme, soll durch das bereits bestehende, im Forschungsprojekt HISTORY entwickelte Toolkit „Vermont“ erledigt werden. Damit ist als Teilziel des Projekts das effiziente Monitoring und die parallele Erkennung auf Multicore-Systemen erfüllt.



Schematischer Aufbau von PADIOFIRE

Mit dem Erkennen alleine ist es allerdings nicht getan: Hat die parallelisierte Anwendungserkennung nun einen interessanten Datenstrom ausgemacht, auf den hin gehandelt werden muss, so muss immer noch die Firewall signalisiert und zu einer Verhaltensänderung bewegt werden. Hier kommt das zweite Ziel des Projekts ins Spiel: Die Entwicklung einer Firewall-Architektur mit asynchron gekoppelter Overlay-Erkennung.



Förderung und beteiligte Partner

Das Projekt PADIOFIRE wird vom Bundesministerium für Bildung und Forschung gefördert.



Koordinierender Partner des Verbundvorhabens ist die Brandenburgische Technische Universität Cottbus. Konsortialpartner sind weiterhin die Friedrich-Alexander-Universität Erlangen-Nürnberg und die Gesellschaft für Netzwerk- und Unix-Administration GeNUA mbH aus Kirchheim bei München. Als assoziierte Partner des Projekts gelten die Universität Innsbruck und Ixia, vertreten durch die deutsche Niederlassung in Gilching bei München.

Übersicht: Partner mit Kurzbeschreibung

BTU Cottbus, Lehrstuhl Rechnernetze und Kommunikationssysteme

- **Kompetenzen:**
Host- und netzbasierte Signaturerkennung, Signature Engineering, Intrusion Detection
- **Rolle im Projekt:**
Projektkoordination, Arbeit an der IDS-Komponente und der regelbasierten Policy-Sprache (speziell: SGML-Einbindung, Signaturen für JavaScript-Anwendungen)





FAU Erlangen, Lehrstuhl für Sicherheitsinfrastrukturen

- **Kompetenzen:**

Expertise auf dem Gebiet praktischer Computersicherheit: Cold Boot Attacks, Malware, Timing und Seitenkanalangriffe, Angriffsmodelle bei drahtlosen Netzwerken

- **Rolle im Projekt:**

Aufstellung eines Systems zur Einschätzung der Gefährlichkeit von vorliegendem JavaScript-Code; Arbeit an dem Kompromiss zwischen Sicherheit und Performance für den Demonstrator des Projekts



GeNUA

Gesellschaft für Netzwerk- und Unix-Administration mbH

- **Kompetenzen:**

Langjährige praktische Erfahrung bei der Implementierung von spezifischen Anwendungsproxies auf dem Application Level Gateway der zweistufigen Hochsicherheits-Firewall GeNUGate; Erfahrungen mit Sicherheitsanforderungen durch mehrfache Zertifizierungs- und Zulassungsverfahren für die hauseigenen Produkte

- **Rolle im Projekt:**

HTTP-Einbindung für das IDS (speziell: Kompression, De-Chunking); Asynchrone Kopplung der eigenen Firewalls in das Gesamtsystem





Universität Innsbruck, Lehrstuhl für Technische Informatik

- **Kompetenzen:**
Themenverwandte Vorarbeiten durch verschiedene Masterarbeiten am Lehrstuhl
- **Rolle im Projekt:**
Aktivierung von IDS auf Basis von Anomalie-Erkennung, Performance-Optimierung durch Monitoring-Toolkit Vermont, evtl. erweiterte Erkennung von HTTP-Sitzungen



Ixia Technologies Europe Limited

- **Kompetenzen:**
Ausgeprägte praktische Erfahrung beim Testen von Komponenten der IT-Infrastruktur im Hochlastbereich als Hersteller eigener Lösungen
- **Rolle im Projekt:**
Bereitstellen von Testequipment für den Demonstrator, Anpassen von Testszenarien für das Projekt



Über den Autor

Dipl.-Inf. Alexander von Gernler ist bei der GeNUA mbH für Forschungsprojekte zuständig und hier stets auf der Suche nach neuen, interessanten Betätigungsfeldern im Bereich Computer- und Netzwerksicherheit. Davor hat er bei GeNUA auch schon als Software- und Systementwickler sowie Scrum-Master gearbeitet und hat daher ein großes Praxiswissen, von dem er für seine jetzige Tätigkeit profitieren kann.