

# XWT Foundation Advisory: Firewall circumvention possible with all browsers

*Source:* <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2002-07/0363.html>

---

*From:* Adam Megacz ([adam@xwt.org](mailto:adam@xwt.org))

*Date:* 07/29/02

To: [bugtraq@securityfocus.com](mailto:bugtraq@securityfocus.com)

From: Adam Megacz <[adam@xwt.org](mailto:adam@xwt.org)>

Date: 29 Jul 2002 10:57:11 -0700

---

XWT Foundation Security Advisory

Adam Megacz <[adam@xwt.org](mailto:adam@xwt.org)>

<http://www.xwt.org/sop.txt>

29-Jul-2002 [Public Release]

---

## Abstract

The following exploit constitutes a security flaw in JavaScript's "Same Origin Policy" (SOP) [1]. Please note that this is *\*not\** the IE-specific flaw reported in February [2].

The exploit allows an attacker to use any JavaScript-enabled web browser behind a firewall to retrieve content from (HTTP GET) and interact with (HTTP <form/> POST) any HTTP server behind the firewall. If the client in use is Microsoft Internet Explorer 5.0+, Mozilla, or Netscape 6.2+, the attacker can also make calls to SOAP or XML-RPC web services deployed behind the firewall.

---

## Status

This advisory is being released in accordance with the Responsible Disclosure Draft RFC [3]. See the last section of this advisory for a timeline. Vendors were notified on 28-Jun-2002, 30 days prior to the public release.

As of 29-Jul-2002, *\*no vendor\** has implemented a fix that will protect clients behind proxies (without external DNS) from the attack variant outlined in the section "Quick-Swap DNS".

Further vendor status can be found in the section "Vendor Responses".

## Exploit

- 1) Attacker controls DNS zone \*.baz.com, configuring it as follows:
  - a) foo.bar.baz.com -> some web server operated by the attacker
  - b) bar.baz.com -> 10.0.0.9 (some address behind BigCo's firewall)
- 2) The attacker induces unsuspecting user at BigCo to visit <http://foo.bar.baz.com/>.
- 3) A JavaScript on said page sets document.domain to "baz.bar.com" (this is valid since baz.bar.com is a parent domain of foo.bar.baz.com). See [1]. Also note that this step is not strictly necessary, but substantially improves the performance of the exploit and the ease of implementation.
- 4) JavaScript on the page then loads a page from <http://bar.baz.com/somePrivatePage.html> into a hidden frame. This page will be retrieved from 10.0.0.9, a machine behind the firewall.
- 5) The JavaScript then extracts the contents of the other frame (it can do this since the two frames' document.domain matches), url-encodes it into a link and loads \*that\* link in another hidden frame, thereby transmitting the contents of the intranet page back to the attacker as part of the HTTP GET request. Large pages could use <form>s and an HTTP POST.

---

## Moving beyond a single server

By adding an entry X.Y.Z.baz.com for each address 10.X.Y.Z, this script could iteratively scan the entire 10.0.0.0/8 netblock. A pop-under could be used to keep a window open (with the JavaScript probe running) long enough to get substantial coverage.

---

## Attacking Web Services

If the client in use is Microsoft Internet Explorer, this technique can be used to access arbitrary SOAP or XML-RPC based web services behind the firewall. Microsoft Internet Explorer 5.0 and later ship with an ActiveX control called "XMLHTTP", which allows JavaScripts to POST XML content to the server they originated from. Although XMLHTTP does not respect changes to document.domain, it is still vulnerable to this Quick-Swap DNS. Credit goes to Jared Smith-Mickelson for suggesting this possibility.

A similar attack should be feasible with Mozilla's XMLHttpRequest object [4].

### Increased sophistication

An even more sophisticated attack would involve the JavaScript querying the attacker's server for a list of IPs/URLs to fetch using this exploit. If the attacker can induce enough users within BigCo to visit the malicious script (by spamming them?), the attacker could construct a proxy server that would route her requests to a cluster of slave javascripts. The attacker would effectively be able to open up a web browser and saunter around the company's intranet as if she were sitting right on it.

---

### Quick-Swap DNS

This variation of the attack will still work even if browser vendors change their policy to prohibit changes to document.domain.

In this situation, the attacker would need a DNS server with the refresh/expire ttl set to zero (no caching allowed). Once the user loads the page from the attacker's web server, the attacker would change her DNS records so that foo.bar.baz.com now points to 10.0.0.9. The exploit would proceed normally. A custom DNS server could be used to automate this process. By allocating a single hostname to each slave JavaScript, an arbitrary number of

Clients can be modified to "lock in" the IP for a given hostname once a page is loaded, although this approach will fail for clients behind a proxy without DNS access.

---

### Short Term Solution (by Dave Ahmad of SecurityFocus)

Web servers behind firewalls should be configured to reject any HTTP requests with an unrecognized "Host" header, rather than serving pages from the "default" virtual host. This can be accomplished without patches by creating a "default" virtual host with no content, and creating a name-based virtual server for each hostname which the server is intended to serve as.

---

### Long Term Solution

Many products have embedded HTTP servers which entirely ignore the Host header since they do not support name-based virtual hosts. The notion of a "default" virtual server is also very useful; it is doubtful that web server vendors will be willing to remove this feature simply to work around a flaw in popular web browsers.

Clearly, a long-term solution to this problem must involve a refinement of the SOP policy.

SOP should be enforced on an IP-by-IP basis, rather than a hostname-by-hostname basis, since the hostname-to-IP mapping is under the control of the attacker, while the IP-to-physical-server mapping is not.

Since some clients behind HTTP proxies do not have access to a DNS server which they can use for name-to-IP resolution, HTTP Proxies should return an additional header in the HTTP reply "Origin-Server-Address:", whose value is the network-layer address of the origin server. A web browser without DNS access which receives a script from a proxy which does not support this header must not be allowed to access content in any other frame, iframe, window, or layer.

---

## Vendor Responses

### Netscape:

Netscape/Mozilla has included a patch in the CVS repository [5] which implements the following two refinements:

- 1) A change to document.domain is only honored if both the source and target frame altered document.domain.
- 2) If the client has access to external DNS, the hostname-to-IP mapping is "pinned" for the lifetime of the page.

These refinements defend against this vulnerability if the client has access to DNS. Clients behind proxies who lack DNS access are still vulnerable to the attack outlined in the section "Quick-Swap DNS".

### Microsoft:

Unsurprisingly, Microsoft's response to this issue came from their Public Relations department, rather than their Engineering department. The statement indicated that Microsoft \*would not\* issue a patch or hotfix, but would prefer to downplay the severity of the vulnerability instead.

---

## Responsible Disclosure Timeline

25-Jun Vulnerability discovered by Adam Megacz, submitted to bugtraq [Discovery Phase begun]

26-Jun Bugtraq moderator (Dave Ahmad) withholds posting, confers with Adam Megacz, proposes short-term solution.

28-Jun Vendor disclosure [Notification Phase begun]

SecurityFocus Bugtraq: XWT Foundation Advisory: Firewall circumvention possible with all browsers

Microsoft Notified: [secure@microsoft.com](mailto:secure@microsoft.com)  
Apache Foundation Notified: [security@apache.org](mailto:security@apache.org)  
Netscape Notified: <http://help.netscape.com/forms/bug-security.html>  
Mozilla Project Notified: [security@mozilla.org](mailto:security@mozilla.org)  
CERT Notified: [cert@cert.org](mailto:cert@cert.org)

01-Jul Advisory updated; SOAP/XML-RPC also vulnerable if client is Microsoft Internet Explorer.

Microsoft Notified: [secure@microsoft.com](mailto:secure@microsoft.com)  
Apache Foundation Notified: [security@apache.org](mailto:security@apache.org)  
Mozilla Project Notified: [security@mozilla.org](mailto:security@mozilla.org)  
CERT Notified: [cert@cert.org](mailto:cert@cert.org)

08-Jul Advisory updated; SOAP/XML-RPC also vulnerable if client is Mozilla.

29-Jul Advisory publicly released on bugtraq.

---

#### Footnotes

[1] <http://www.mozilla.org/projects/security/components/same-origin.html>  
<http://developer.netscape.com/docs/manuals/communicator/jsguide4/sec.htm>

[2] <http://online.securityfocus.com/bid/3721>

[3] <http://www.ietf.org/internet-drafts/draft-christey-wysopal-vuln-disclosure-00.txt>

[4] <http://unstable.elemental.com/mozilla/build/latest/mozilla/extensions/dox/interfacensIXMLHttpRequest.html>

[5] [http://bugzilla.mozilla.org/show\\_bug.cgi?id=154930](http://bugzilla.mozilla.org/show_bug.cgi?id=154930)

--  
Sick of HTML user interfaces?  
[www.xwt.org](http://www.xwt.org)

---

- **Previous message:** [kokane: "KDE 2/3 artsd 1.0.0 local root exploit"](#)
- **Next in thread:** [Peter Watkins: "Re: XWT Foundation Advisory: Firewall circumvention possible with all browsers"](#)
- **Reply:** [Peter Watkins: "Re: XWT Foundation Advisory: Firewall circumvention possible with all browsers"](#)
- **Reply:** [GreyMagic Software: "RE: XWT Foundation Advisory: Firewall circumvention possible with all browsers"](#)
- **Reply:** [Jason Coombs: "RE: XWT Foundation Advisory: Firewall circumvention possible with all browsers"](#)
- **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)