

RE: Is this as bad as it seems?

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/security-basics/2002-07/0798.html>

From: Jason Coombs (jasonc@science.org)

Date: 07/30/02

From: "Jason Coombs" <jasonc@science.org>
To: "Stefan Osterlitz" <osterlitz@p-p.de>, <security-basics@securityfocus.com>
Date: Tue, 30 Jul 2002 09:38:00 -1000

The network being protected by the router or firewall is still vulnerable to attacks like the one described yesterday by XWT Foundation. See below.

-----Original Message-----

From: Jason Coombs [<mailto:jasonc@science.org>]
Sent: Tuesday, July 30, 2002 9:32 AM
To: Thor Larholm; Microsoft Security Response Center;
bugtraq@securityfocus.com
Cc: Adam Megacz
Subject: RE: XWT Foundation Advisory

Aloha, Thor.

- > *I still quite fail to see the relevance to firewalls, as nothing is*
- > *circumvented – the administrator has explicitly allowed HTTP traffic on*
- > *(most often) port 80.*

Outbound HTTP traffic is allowed by the firewall administrator, yes, but this exploit has the effect of allowing the attacker to send *INBOUND* HTTP requests to any HTTP server inside the firewalled network from a remote location outside the firewall. The HTTP servers behind the firewall are otherwise normally protected from remote access by the existence of the firewall. The HTTP servers that are at-risk are not the ones that service inbound requests from outside the network but rather those HTTP servers that are supposed to be accessible only to users on the LAN.

The remote attacker uses the browser as a JavaScript-based HTTP "proxy by force".

The two most important conditions for vulnerability are:

- 1) The HTTP server (located on the internal network or anywhere else that is accessible to the client browser) must be configured to respond to HTTP/1.0-style requests that do not supply a Host: header. Even though the browser sends Host: header along with its HTTP/1.1-compliant request, the "default" Web site explicitly ignores the Host: header in order to maintain

SecurityFocus BASICS: RE: Is this as bad as it seems?

compatibility with HTTP/1.0 clients.

2) The HTTP server must not require manual authentication or a cookie as a condition for access to the requested URL. In some scenarios the attacker may be able to compel the victim browser to send its cached HTTP Basic Authentication credentials along with the request in order to authenticate automatically with realms that same browser has previously authenticated with through user-supplied login credentials.

The reason that a Host: header defeats the JavaScript-based proxy by force is that the client thinks its sending the request to a host inside the remote DNS domain that triggered the exploit. The Host: header contains that remote (malicious) DNS domain, and the Web server in question (on the internal network, for example) won't have that Host: header configured.

Sincerely,

Jason Coombs
jasonc@science.org

-----Original Message-----

From: adam@megacz.com [mailto:adam@megacz.com] On Behalf Of Adam Megacz
Sent: Monday, July 29, 2002 7:57 AM
To: bugtraq@securityfocus.com
Subject: XWT Foundation Advisory: Firewall circumvention possible with all browsers

=====
==

XWT Foundation Security Advisory

Adam Megacz <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

SecurityFocus BASICS: RE: Is this as bad as it seems?

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.

SecurityFocus BASICS: RE: Is this as bad as it seems?

- 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]

Microsoft Notified: secure@microsoft.com

Apache Foundation Notified: security@apache.org

Netscape Notified:

<http://help.netscape.com/forms/bug-security.html>

Mozilla Project Notified: security@mozilla.org

CERT Notified: cert@cert.org

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

Microsoft Notified: secure@microsoft.com

Apache Foundation Notified: security@apache.org

Mozilla Project Notified: security@mozilla.org

CERT Notified: cert@cert.org

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

29-Jul Advisory publicly released on bugtraq.

Footnotes

SecurityFocus BASICS: RE: Is this as bad as it seems?

[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/intefacensIXMLHttpRequest.html>

[5] http://bugzilla.mozilla.org/show_bug.cgi?id=154930

--
Sick of HTML user interfaces?
www.xwt.org

-----Original Message----- From: Stefan Osterlitz [mailto:osterlitz@p-p.de] Sent: Monday, July 29, 2002 6:07 AM To: security-basics@securityfocus.com Subject: Re: Is this as bad as it seems?

On Sun, 28 Jul 2002 12:21:49 -0700 (PDT), Jay wrote:

>I just inherited a network with what I believe are >numerous security holes. Here is an overview. >>My questions are, (1) how effective is a router-based >access list that blocks ports, compared to a firewall? >Pros? Cons?

you remain more vulnerable to DoS attacks and spoofed traffic. a stateful firewall (as opposed to your router) keeps track of the connections your server has initiated. data is checked whether it is the answer to an legitimate request your server made.

(2) Is it correct that putting public >and private hosts on different subnets is nothing more >than minimal security by obscurity, and a major risk?

as long as there are no routing restrictions between the subnets, there is not even the obscurity ;-))

>(3) Is it as crazy as it seems, to put your domain >controllers on public hosts? My thought is, a hacker >who "owns" a PDC will own the entire network's >security. >

exactly right. external servers should not even be in the same domain as the workstations. best practice ist to set up one domain per external server or no domain on the servers. explicitly set trust relations if you have to. also, create a dmz (put a firewall between external and internal servers)

>Management believes this configuration is safe enough >because (1) malicious traffic is "stopped at the >router";

not at all. let us say that 90% of (directed, not worm) attacks bypass a cheap firewall. for example. almost every workstation functions as a dns client. any incoming udp packet coming from port 53 is allowed in your router. the latest sql server 2k exploit needed exactly one such packet to own your server....

RE: Is this as bad as it seems?

SecurityFocus BASICS: RE: Is this as bad as it seems?

(2) there is no risk from malicious web >hosting clients because their accounts are User-level >accounts with FTP-only access, and therefore cannot >run malicious programs;

there are usually two grades of exploits: local and remote. as soon as ftp access is allowed, local exploits become feasible. (if php or perl are allowed (or any active content), it's as good as a shell account for that purpose)

and (3) they aren't >particularly concerned with systems compromise via >DNS, DC, SQL, or other attacks aimed at >publicly-accessible services, again because of the >router access lists blocking most ports.

hmmm.. partially right.. if your sql server is not accessible from the outside, it can't be attacked. but security means just as secure as the weakest link. that could be any of your workstations. your sql server will be accessible to them at least. *zap – now they are vulnerable again.

> >Basically, they believe the access list at the border >is exceptionally effective because you can't get >attacked by what can't reach your hosts. My >background has primarily been desktop and application >support, so the responsibility of server/network >security is new to me.

draw them a simple diagram.. big red line from the router to the workstations (internet / email traffic) big red line from each workstation to the relevant servers (application traffic) then put the simple question: now which server is not accessible?

> >I believe this network is a disaster waiting to >happen, but I don't have enough knowledge on the >subject to create a detailed list of what's wrong for >my boss. I'm asking for any advice, URLs, etc., that >address what I believe are gaping holes mentioned >above, plus those which I may not have thought of.

get a good firewall. you get them for \$800 for nearly any size of internet connection. see cisco, nokia, sonicwall, checkpoint

get a good corporate antivirus for ca \$2000 upward as an smtp gateway. check all corporate email

configure your firewall with a dmz setup. separate external servers, internal servers, workstations into three nets

change your domain setup: one domain per server, one internal domain

put one cheap linux machine into your dmz with pop / smtp / http proxies.

this setup costs one week of work plus 3k – 4k \$\$\$ and places you in a better security league than 70 % of corporations worldwide.

Greetings, Stefan Osterlitz

-
- **Previous message:** [Alberta Book Bindery: "RE: OpenSSHd problem."](#)
 - **In reply to:** [Stefan Osterlitz: "Re: Is this as bad as it seems?"](#)
 - **Next in thread:** [Enquiries: "RE: Is this as bad as it seems?"](#)
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)