Rebind-Hijack : DNS Hijacking through DNS Rebinding

Abstract: *DNS Rebinding has been around for almost 15 years, Many security experts discussing method to emply it. This paper illustrates another attack which transfers the active control of a whole intranet Internet access feasibly once a single user of the domain visits a single corrupted web page; via DNS Hijacking. Possible attack scenarios, Demonstrations, defense and prevention strategies and weaknesses are discussed as well as a probabilistic approach to the success rate.*

Keywords: *DNS Rebinding, Same Origin Policy, Default Password, Man in the Middle, Pharming, DNS Hijacking, Phishing, Session Hijacking, Session Fixation, Sniffing*

Category: *Information Systems Security, Cross Site Scripting, DNS*

version 0.5 (May 2011)

Abbas Naderi
Shahid Beheshti University
abiusx@acm.org

1. Introduction

DNS is a very fundamental hierarchical protocol on the Internet - and also on every network - with the sole purpose of converting domain names (e.g **www.abiusx.com**) to IP addresses (and sometimes vice-versa), hence making it possible to conncet to hosts.

Since DNS relies on UDP transport layer protocol, and also quite old [16], Not much security was considered for and applied to it. Even DNSSEC which is going to be de-facto security standards for the DNS protocol, Would not solve many of the issues without breaking backward compatibility.

Web browsers keep user transactions on each web site private by following the Same-Origin Policy (SOP), Which indicates that dynamic contents of every web page (e.g Javascript, Flash Player, Applets) can only establish bidirectional connections to web addresses of their own origin, i.e a Javascript loaded form **www.abiusx.com** can not send a request to **www.not-abiusx.com** and read its response, but can (and usually does) send requests to **www.abiusx.com** and read the responses.

This mandatory step prevents many kinds of security flaw - namely Cross Site Scripting (XSS) attacks - to be interactive and is the sole mechanism that isolates web sites served on a web browser from each other's (malicious) reach. [7]

DNS Rebinding is a very effective attack, relatively easy to deploy and having a massive impact, which can simply bypass SOP in almost all modern web browsers, And has been around for approximately 15 years, but

some hands prevented it from getting out for almost 10 years (until 2006 [2]).

This paper tends to step furthur in DNS Rebinding possibilities introduced in [1] and [2], Developing new attacks which not even bypass SOP fully, but perform total DNS Hijacking, hence granting the attacker full control of private network Internet connections .

2. DNS Rebinding

To illustrate how DNS Rebinding actually works, consider this simplified scenario :

Once the Victim visits a web page on *www.attacker.com*, Victim's web browser performs a DNS Name Lookup operation on www.attacker.com, Which brings up the assigned IP of "4.4.4.4" , So the browser connects to the web server on the computer system at Internet address "4.4.4.4" and loads the attacker's web page.

The Same Origin Policy dictates that dynamic content on the loaded and served attacker's web page can only interact to www.attacker.com which is in fact the computer at IP address "4.4.4.4", So the attacker would not be able to access any other site-related information on the browser.

Now the dynamic content on attacker's webpage somehow tends to force expiry of the name lookup result for *www.attacker.com* (discussed in section 5, exploiting DNS TTL), So when the dynamic content sends another request to www.attacker.com, The browser needs to perform name lookup again.
This second name lookup, which is also received and processed by attacker.com's DNS server (located on attacker's computer) returns a different IP address, generally not even related to the attacker's website, such as "2.2.2.2"; instead of the real one which was "4.4.4.4".

This new IP "2.2.2.2" might belong to any server, deliberately targeted by the attacker, and attacker's customized DNS [17] can freely choose the server to return it's IP address (even with DNSSEC specifications).

Afterwards, SOP is bypassed and the dynamic content on attacker's web page (which now resides in the Victim's browser) can interact the server at "2.2.2.2" and read responses, since the browser thinks "2.2.2.2" is in fact **www.attacker.com** (the same site) and the operation is SOP compatible, but "2.2.2.2" actually belongs to another server such as **www.target.com**.

Bypassing SOP via DNS Rebinding brings up many possibilities, Explained briefly in section 3.

3. Possible Attacks

Two traditional category of attacks are possible considering DNS Rebinding (according to [1]) :

(a) Firewall Circumvention

Since now the dynamic content residing on attacker's web page can easily contact every IP address with the Victim as the source of connection, it can act as a proxy for the attacker to access the Victim's Intranet without the firewall noticing it.

(b) IP Hijacking

The dynamic content of the attacker's web page can easily connect to every server on the Internet and read its responses, and also connect to the attacker's backend (www.attacker-backend.com).

A duplex HTTP Streaming (aka Comet) could be employed to allow attacker to control Victim's browser to contact any desired HTTP web site and even connect to ports other than the default http port 80 tcp, thus performing malicious network operations (e.g sending e-mails); all framing the Victim as the source of connection.

Both these attack categories and their belonging attacks (BotNets, Click Fraud, Spam, Spidering, Abusing resources, Cracking unpatched systems, IP based authentication, IP/Port Scan and etc.) are covered in [1]. We shall now begin to introduce another attack, Which takes a few more steps further in both impact and risk.

4. DNS Hijacking

Most users use home/office routers with default configurations (IP/Port, interface Username/Password) since they consider the router inaccessible from the outside network (i.e Internet) and only accessible via physical link between personal system and the router.

Using the DNS Rebinding supplied with a comprehensive backend, An attacker can exploit these routers in a way that puts the whole network Internet connection in full active control of the attacker. The attack (namely Rebind-Hijack) is explained by a scenario in depth below, which for simplification, assumes only one victim at a time:

Rebind-Hijack Scenario. Attacker launches a server with two websites, **xrebind.com** on "4.4.4.4" and **xrebind-backend.com** on "5.5.5.5". The server also includes a customized DNS server on "4.4.4.4" (Manipulated BIND or PowerDNS with PipeBackend).

The **rebind-backend.com** website hosts a CGI script which receives three parameters, the explicit IP address and the explicit hostname and the implicit IP address (source IP) of the request. This script then updates the customized DNS tables and sets the input parameters so that Name Lookup of the hostname in parameters returns the explicit IP in parameters, whenever performed by Source IP.

Now the **xrebind.com** is served to a Victim to start DNS Rebinding attack. The dynamic script loads a whole set of router parameters from the **xrebind-backend.com** (interface address and types, model, default username and passwords, interface DNS configuration management requests and ... all available at [15]) and starts the Rebind-Hijack process, trying all the possible router configs.

The **xrebind.com** web page presents some amusing content to keep the Victim on the page as long as possible so that the script can complete the exploitation (e.g an interactive Flash Game). Once a match is made and dynamic client-side script is successfully logged in to the router's interface, It tries to set router's Primary and Alternate DNS to "5.5.5.5" (attacker's customized DNS).

Keep in mind that the dynamic script can easily interact with the original attacker's backend server via JSONP or iframe requests to **xrebind-backend.com** (or any equivalent), thus receiving instructions and even allowing the attacker to directly control the attack progress.

From now on, Not only attacker can bypass SOP (As explained in the previous section), He can do the following as well:

1. MITM : every name lookup could return attacker's MITM proxy server.

2. Session Hijacking & Session Fixation : if the attacker's DNS returns "4.4.4.4" for every website such as facebook.com or yahoo.com (on desired occasions, preferably not all the time) the attacker would easily acquire the session cookies of these web sites. Then a refresh is done and the DNS can simply return the correct IP address this time.

3. Pharming, Phishing and Fake Pages : attacker can return fake versions of famous web site login pages such as GMail, Yahoo! and facebook periodically with the original domain on the address bar, So that the user would enter credentials without even doubting.

4. Sniffing : Using a simple proxy server, The attacker can sniff all the traffic routed for

a domain on the Internet from the victim intranet.

5. IP/Port Scan : If the router is in fact the gateway to the Internet, the customized DNS can simply return local addresses thus performing an IP/Port scan.

If this scenario is applied carefully, The Victim would almost never know what has happened and the whole Victim intranet's Internet access would be in total control of the attacker for a durable period of time.

The attack was named Rebind-Hijack since it exhausts use of DNS Rebinding to hijack DNS and extensively bind/rebind every request from then on.

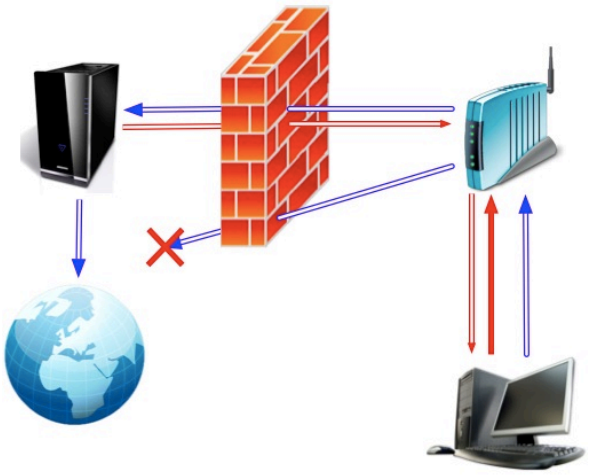

Figure 1: Rebind-Hijack Approach

5. DNS Pinning Problem

There's one issue left that we had postponed, And its the major defense applied widely against DNS Rebinding, known as DNS Pinning.

Browsers (and their common plug-ins) employ a mechanism which prevents basic DNS Rebinding, by pinning an IP address to a host name for a relatively long period of time, Ignoring the original TTL of the DNS name lookup result.

This feature, though effective, Can be bypassed by various means since some major load balancing technologies such as Dynamic DNS rely on short TTLs to change host IPs frequently so that clients would request load from different IPs (Although this happens rarely for a single client), And browsers are not willing to sacrifice performance and robustness for more security.

Most browsers, Namely MS IE 6-7, Mozilla Firefox 1-2, Apple Safari and Opera are subject to known DNS Pinning Bypass methods described and demonstrated thoroughly in [4] and [5].

These browsers pin host/IPs for no more than 120 seconds, Which might prevent brute-force and fraud DNS Rebinding uses, But the Rebind-Hijack requires just a few rebindings so a few minutes spent on the web page would do the trick.

6. Demonstration Results

The attack was successfully developed and deployed on [18], With a success rate of 40%, but due to the security reasons the demonstration details are omitted in this version of the paper.

To obtain thorough demonstration package, visit the website.

7. Where it fails

This attack would not work if a user uses a custom DNS service on the network such as OpenDNS.

It also fails if the user changes the default username/password of the router, But can succeed if user only changes the IP address of the router (via exhaustive search).

If user is cautious and only enters his/her password on authentic HTTPS connections, Fake pages and MITM attacks won't work as well.

On the websites where session hijacking is applied, secure implementations of Remem-

ber Me and Session cookies prevent session fixation.

If the user's intranet employs a DNS and/or web cache on the gateway, There's a possibility that some portion of attacks would fail, namely those dependent on low TTL DNS queries.

Using recursive DNS resolvers on the client side, Cause the attack to become much harder since the DNS server would have no way to determine the client, thus serving appropriate records.

Cumulatively, All of the above only exclude a tiny subset of all victims, Which is forgivable since any single victim on a subnet could easily infect all the subnet.

8. Defense & Precautions

As of defenses and precautions for DNS Rebinding, No solid fail-proof solution exists yet, And most probably would not exist in the near future.
This is due to the nature of the attack which abuses some DNS capabilities which are also used in many peaceful scenarios, And by disabling all those capabilities, We shall lose many beneficial DNS usages.

There are a few defenses described in [1], We shall explain here why these defenses are ineffective against Rebind-Hijack in general:

8.1 Fixing Firewall Circumvention

This defense employs a tool such as **dnswall** [12] that won't let external hostnames to be mapped to internal IP ranges. This precaution would disable VPNs, Since VPNs return their own websites IPs as internal IPs.

8.2 Fixing Plug-ins

8.2.1 Flash Player : Requiring a policy for every socket connection for Flash Player would require all Flash Movie servers to patch themselves, Which would not be backward compatible thus requiring a ver long time to propagate, As well as a much slower Flash interactivity.

8.2.2 Java : The Java defense requires all Java applet clients and servers to be patched, Which is - considering the Java nature - almost impossible. Many Java applets are really old and have not so clean codes to be patched.

8.3 Fixing Browsers

8.3.1 Host Headers : Host Headers makes web servers check for the Host in HTTP headers and validate it with their own (usually employed to enforce Virtual Hosts).

This is in fact a very intelligent approach, And is a de-facto standard now, But has nothing to do with Rebind-Hijack since the malicious server can just ignore it and the original server is contacted by its own IP.

8.3.2 Finer Grained Origins : As explained before, This would prevent many correct DNS usages currently under heavy traffic over the web, And the user would have no idea what is wrong with the web site.

8.3.3 Smarter Pinning : Could be bypassed by anti-pinning approaches, and would not impact the attack much, since Rebind-Hijack requires a few rebindings.

8.3.4 Policy Based Pinning : Almost the same as reverse DNS lookup, This is content to much debate both in Internet infrastructures and DNSSEC workgroups [13].

8.3.5 Trusted Policy Providers : This is also included in DNSSEC and is content to debate, But even when implemented, Provides backward compatibility for older DNS servers.

The generous people at [1] also stated that non of these defenses (or even the sum of them) are a 100% fail proof cost-effective way of defending against DNS Rebinding without losing much as well.

We suggest, As suggested before [14] the only effective defense against DNS Rebinding is general awareness. The fact that the authorities prevented [2] from general publication due to the high risk for as long as ten years, And all the obfuscation over this not so new threat, Only makes it harder and harder to guard against. Also general awareness is costly and requires a lot of security experts to put effort into the field, It is worth the cost since the original DNS Rebinding and the Rebind-Hijack attacks are very cost-effective with high risk and huge impact and should be prioritized as soon as possible.

9. Further Work

There are two area's of research open on Rebind-Hijack, making it highly more effective.

First, Finding a way to determine the IP belonging to the DNS query received at the server with certainty, Would ease the attack a lot, since The backend could provide client-specific records.

Second, Employing some heuristics dissimating the need to exhaustive search for the router, Instead determining it via headers and possible configurations of the network, based on an open database.

There's also the need to verify DNSSEC specs and implementations against Rebind-Hijack similar attacks, Defining the mandatory configuration of the protocl.

The Blackhat group, published a detailed guide on DNS Rebinding after this paper. The ideas provided here and there could be merged to achieve better results as well.

10. Conclusion

Rebind-Hijack, and DNS rebinding in general, have been around for too long now. Currently many well known attackers are using this approach to gather BotNets and Zombies [19] [20].

Due to the huge impact of this attack, General awareness is required to stop the general Internet/intranet user from getting affected.

11. References

[1] Protecting Browsers from DNS Rebinding Attacks http://crypto.stanford.edu/dns/dns-rebinding.pdf

[2] Using the Domain Name System for System Break-ins (Steven M. Bellovin AT&T Bell Laboratories)

[3] Breaking the Same-Origin Policy http://shampoo.antville.org/stories/1451301/

[4] HTTP based DNS Rebinding Demonstration http://www.jumperz.net/index.php?i=2&a=1&b=7

[5] Flash Player based DNS Rebinding Demonstration http://www.jumperz.net/index.php?i=2&a=1&b=8

[6] Flash Player 9,0,115,0 DNS Rebinding Fix http://www.adobe.com/devnet/flashplayer/articles/fplayer9_security.html#goal_dns

[7] Same Origin Policy, W3 Web Security http://www.w3.org/Security/wiki/Same_Origin_Policy

[8] DNS Wall http://crypto.stanford.edu/dns/

[9] DNS Pinning just got worse http://ha.ckers.org/blog/20060908/dns-pinning-just-got-worse/

[10] Craig Heffner : How to hack millions of routers https://www.blackhat.com/html/bh-us-10/bh-us-10-briefings.html#Heffner

[11] DNS rebinding attack : suggesting to change router configurations http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1308912,00.html

[12] dnswall http://code.google.com/p/google-dnswall/

[13] DNSSEC http://www.dnssec.net/ http://tools.ietf.org/html/rfc2535

[14] DNS Rebinding, How to defend http://www.abiusx.com/dns

[15] Default Passwords for routers http://www.phenoelit-us.org/dpl/dpl.html

[16] DNS RFC http://tools.ietf.org/html/rfc768

[17] PowerDNS Pipe-Backend http://doc.powerdns.com/pipebackend-dynamic-resolution.html

[18] rebind-hijack.abiusx.com , successful development, deployment and demonstration of Rebind-Hijack

[19] http://www.blackhat.com/presentations/bh-usa-07/Byrne/Presentation/bh-usa-07-byrne.pdf

[20] https://media.blackhat.com/bh-us-10/presentations/Heffner/BlackHat-USA-2010-Heffner-How-to-Hack-Millions-of-Routers-slides.pdf