

1 DNS重绑定调研

调研者：姜鹏辉

1.1 简介

1.1.1 概念






DNS重绑定(DNS Rebinding Attack) 是计算机攻击的一种形式。DNS重绑定通过滥用DNS来绕过同源策略的保护，当用户访问恶意网页时会运行指定脚本，将受害者的浏览器转换为开放的代理，攻击网络上其他的计算机。

在网页浏览过程中，用户访问一个指定的域名，浏览器通过DNS服务器将域名解析为IP地址，然后向对应的IP地址请求资源，最后展现给用户。而对于域名所有者，他可以设置域名所对应的IP地址。当用户第一次访问，解析域名获取一个IP地址；然后，域名持有者修改对应的IP地址；用户再次请求该域名，就会获取一个新的IP地址。对于浏览器来说，整个过程访问的都是同一域名，所以认为是安全的。这就造成了DNS重绑定攻击¹。

1.1.2 现状

斯坦福大学在2007年发表文章Protecting Browsers from DNS Rebinding Attacks²对DNS重绑定进行了系统的研究并提出了一些解决方法。

Speed Measured / Target Definition

Browser	OS	Strategy	Time to Exploit	Fetch Interval	Target Spec
	Windows 10	MA	3 seconds	1 second	127.0.0.1
	Ubuntu	MA	3 seconds	1 second	0.0.0.0
	macOS	MA	3 seconds	1 second	0.0.0.0
	macOS, Ubuntu, Windows	FS+Cache Flooding	15-40 seconds	1 second	Any
	iOS	FS+Cache Flooding	5 seconds	1 second	Any

上图是在使用singularity工具进行DNS重绑定攻击时，在各个系统和浏览器下的完成攻击所用的方法和时间³

1.2 原理

1.2.1 相关概念

1.2.1.1 同源策略(Same-Origin Policy)

是一项浏览器安全功能，同源策略限制了从同一个源加载的文档或脚本，与来自另一个源的资源进行交互。这是一个用于隔离潜在恶意文件的重要安全机制。

1.2.1.2 DNS TTL

即DNS服务器缓存此条DNS记录的时间，单位为秒。

1.2.1.3 Whonow 服务器

是一款能够帮助渗透测试人员实时执行DNS重绑定的DNS服务器，Whonow允许定义DNS响应并通过域名请求来实现规则的动态重绑定。

例如对于域名 `A.192.168.1.1.forever.rebind.network` 的DNS查询结果就是192.168.1.1(为简洁表达，查询结果省略无关信息)

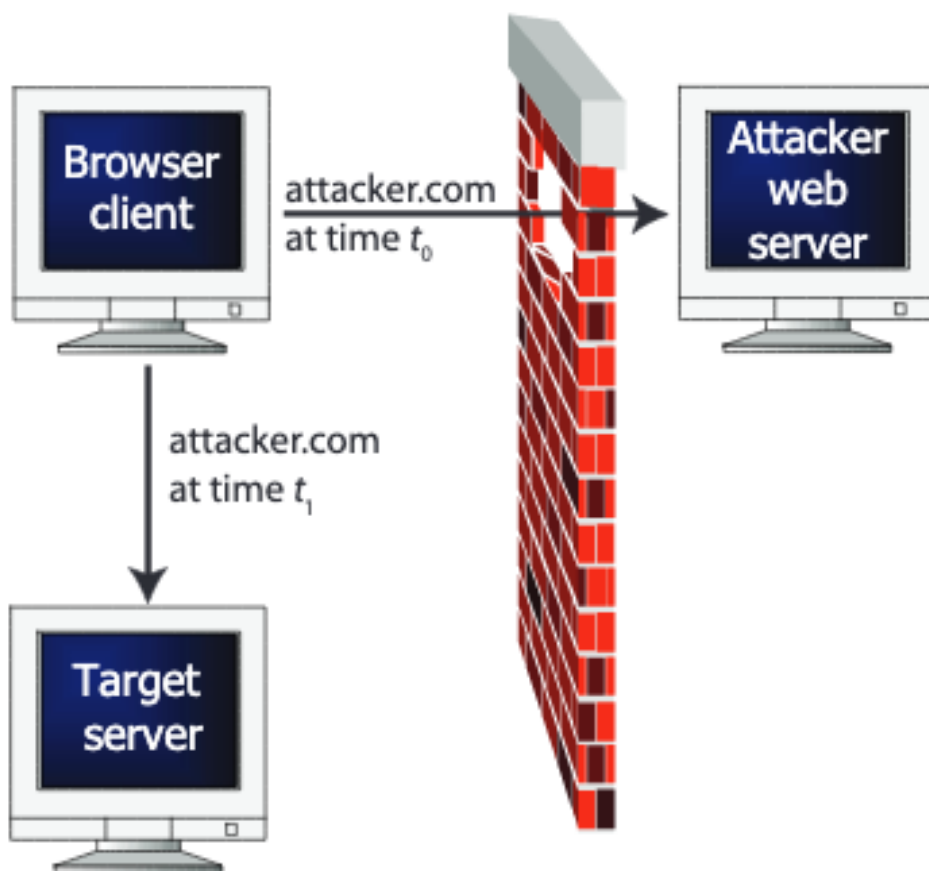
```
>>> dig A.192.168.1.1.forever.rebind.network
;; ANSWER SECTION:
A.192.168.1.1.forever.rebind.network. 1 IN A 192.168.1.1
```

对于域名 `A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network` 的查询则轮流返回 `127.0.0.1` 和 `10.0.0.1`

```
>>> dig A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network
;; ANSWER SECTION:
A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network. 1 IN A 127.0.0.1
```

```
>>> dig A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network
;; ANSWER SECTION:
A.127.0.0.1.1time.10.0.0.1.1time.repeat.rebind.network. 1 IN A 10.0.0.1
```

1.2.2 攻击过程



1. 攻击者注册一个恶意域名例如 `attacker.com`，通过多种方式吸引受害者访问此恶意域名
2. 当用户第一次访问恶意域名时，DNS服务器响应的A记录指向攻击者的服务器，并将TTL设为一个特别小的值（例如1s），以使受害者的机器不会长期缓存这条记录
3. 攻击者的服务器将包含恶意的JavaScript脚本或Java程序，当受害者的浏览器运行该脚本时，它会

为该域发送一个新的DNS请求和一些其他的请求。

4. 第二次DNS请求时，攻击者会使用新的IP地址进行回复，此IP地址可能是私有IP地址也可能是某个公有IP地址
5. 浏览器根据域名会认为两个服务器符合同源策略，所以允许脚本对第二个服务器发出请求和读取响应
6. 此时受害者的浏览器就变成了一个开放的代理，可以对指定的服务器发送请求。

1.2.3 Multi-Pin攻击

现代的浏览器使用多个插件（例如Flash，Java）来显示一个网页，攻击者可以维护多个独立的DNS Pin数据库，通过不同pin获得的DNS解析结果是不同的，通过这种方式可以在几百毫秒内完成攻击。

1.2.4 工具

1.2.4.1 rebind

Kali Linux提供了工具Rebind用于发起DNS重绑定攻击，例如用指定的域名 `kali.local` 发起dns重绑定攻击

```
root@kali:~# rebind -i eth0 -d kali.local

[+] Starting DNS server on port 53
[+] Starting attack Web server on port 80
[+] Starting callback Web server on port 81
[+] Starting proxy server on 192.168.1.202:664
[+] Services started and running!

> dns
[+] 192.168.1.202    kali.local.
[+] 192.168.1.202    www.kali.local.
[+] 192.168.1.202    ns1.kali.local.
[+] 192.168.1.202    ns2.kali.local.
```

具体使用方法见<https://tools.kali.org/sniffingspoofing/rebind>

1.2.4.2 DNS Rebind Toolkit⁴

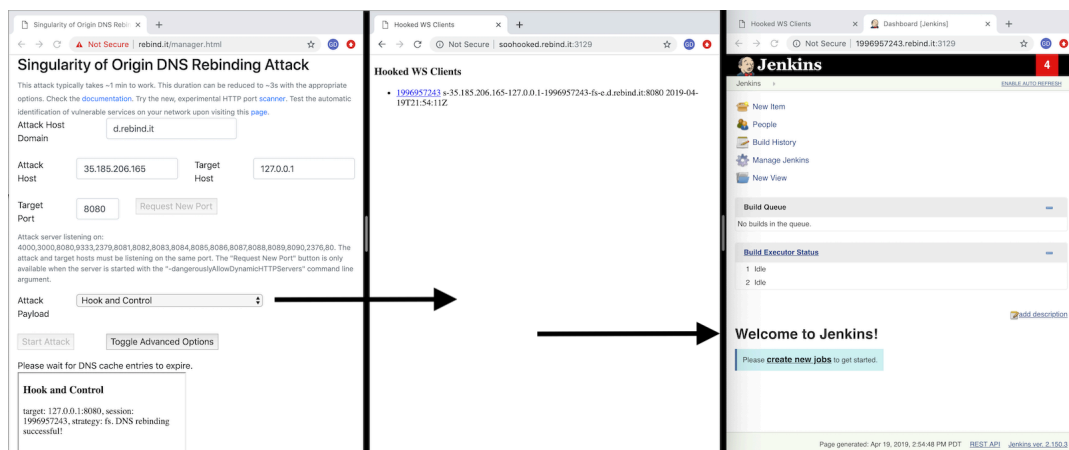
DNS Rebind Toolkit是一款前端JavaScript框架，可以对存在漏洞的主机或局域网进行重绑定攻击。在这款工具的帮助下，远程攻击者可绕过路由器的防火墙，然后直接与目标用户家庭网络内的设备进行交互，并提取出隐私信息，在某些情况下他们甚至还可以直接控制目标设备。

工具的安裝及使用方法见<https://github.com/brannondorsey/dns-rebind-toolkit>

1.2.4.3 singularity⁵

由ncc团队提出的一个发起DNS重绑定攻击的工具，位于<https://github.com/nccgroup/singularity>提供了一套完整的解决方案，包括：

- 一个自定义的DNS服务器来实现重绑定
- 提供了HTTP的管理员界面来对攻击进行管理
- 给出了一些现成的示例攻击载荷
- 除了IP地址外，还支持指定DNS的CNAME



1.2.5 应用

1.2.5.1 攻击路由器

攻击者使用DNS重绑定攻击用户局域网中的无线路由器：

1. 大部分路由器的管理员登录界面是 `http://192.168.1.1/login`，使用默认账户（如admin:admin）进行登录
2. 通过UPnP使用路由器的Internet网关设备（IGD）接口来配置永久端口转发连接，并将网络上的任意UDP和TCP端口暴露给公共Internet。

1.2.5.2 攻击物联网设备

许多常见的物联网设备都可能被DNS重绑进行攻击，例如Google Home系列产品提供了一些设备控制的API，例如播放内容或者重启、恢复出厂设置等，此类API无需身份验证，攻击者可以通过JavaScript脚本轻松调用这些API实现设备控制。

网络安全公司Armis分析了DNS重绑定对物联网设备的影响，调查显示，企业使用的近5亿智能设备易受DNS重新绑定攻击，包括智能电视、路由器、打印机、监控摄像头、IP电话、智能助手等等⁶。<http://rebind.network>是一个DNS重绑定攻击的演示网站，在访问时会尝试检测你当前网络中的Google Home、Roku、Sonos等物联网设备

1.2.5.3 僵尸网络

攻击者可以将DNS解析结果重绑定到公有ip上，将受害者的机器变成一个bot，进行广告点击、发送垃圾邮件等操作

<http://rebind.network/rebind/index.html>

1.3 防御

DNS重绑定的防御手段可以在浏览器、插件、DNS解析程序、防火墙和服务器上实现，在与外部域和内部域通信时的所有服务上使用TLS加密，可以比较有效的避免DNS重绑定。

1.3.1 防火墙

通过制定防火墙的规则，禁止将外部主机名解析为内部ip地址，可以保护局域网内的设备受到攻击。

1.3.2 修复浏览器插件

许多网页是通过各种插件来显示，例如Flash和Java。许多插件直接建立一个新的socket与服务器通信，插件建立socket时的DNS解析结果可能指向目标主机。可以通过实现认证策略，或通过ip而不是host来建立socket连接，来避免重绑定攻击。

1.3.3 浏览器

- **检查header字段:** HTTP/1.1要求在http请求中包含host字段来指定服务器的主机名，浏览器可以通过DNS反向查询等方法检查主机名与ip是否一致
- **DNS pinning:** 目前浏览器将域名解析为一个ip地址后，无论TTL值为多长，都将会缓存这个结果一段时间，这可以减少DNS重绑定的影响。
- **修改浏览器和插件的缓存机制:** 浏览器中缓存的脚本可能在DNS重绑定后在后台运行，所以需要检查原始脚本运行时存储的IP地址和URL与当前是否一致。

1.4 结论

目前DNS重绑定漏洞已经被发现并研究了很长一段时间，它通过修改DNS解析的结果，来绕过浏览器的同源策略，使用户向某个指定的服务器发送请求并获取响应。这种攻击的出现使得防火墙内部的局域网设备也容易受到攻击。已经有很多的设备及浏览器为DNS重绑定设定可安全策略，但还是存在许多漏洞使得DNS重绑定可以攻击成功。比较有效的防御方法是使用TLS加密和身份认证机制。

1.5 引用

-
1. <https://www.tripwire.com/state-of-security/vert/practical-attacks-dns-rebinding/> ↗
 2. JACKSON C, BARTH A, BORTZ A, 等. Protecting browsers from DNS rebinding attacks[J]. ACM Transactions on the Web, 2009, 3(1): 2:1–2:26. DOI:10.1145/1462148.1462150. ↗
 3. <https://docs.google.com/presentation/d/1O7MxvblfRcPSlbyZbFxD-fAR34XlquQSIRAHp2kR4E/edit#slide=id.p> ↗
 4. <https://www.freebuf.com/sectool/177299.html> ↗
 5. <https://github.com/nccgroup/singularity> ↗
 6. <https://www.freebuf.com/company-information/178622.html> ↗