DNS安全皮毛（一）

backcover7 / 2019-08-17 10:25:00 / 浏览数 5164 安全技术 WEB安全 顶(1) 踩(0)

背景介绍

本文基本上是今年六七月花挺大量的一段时间读了一些学术论文，也提及了一些常见的DNS利用技巧，有旧有新然后整合起来做个简单的总结。涉及的范围不是很广，但还是

DNS Data

Data Collections

一般来说，我们会通过收集数据来丰富我们的研究工作，同时这些数据也是可以被用作在安全测试当中。比较推荐的是Rapid7的开放DNS数据库，他们提供了存储DNS解析
DNS）和PTR信息的RDNS（Reserve DNS）。

https://opendata.rapid7.com/sonar.fdns_v2/

https://opendata.rapid7.com/sonar.rdns_v2/

同时Rapid7也提供了人性化的api接口供用户进行即时查询读取，接口地址为https://opendata.rapid7.com/apihelp/。类似的DNS数据库在互联网上也可以找到，类似有D

Passive DNS

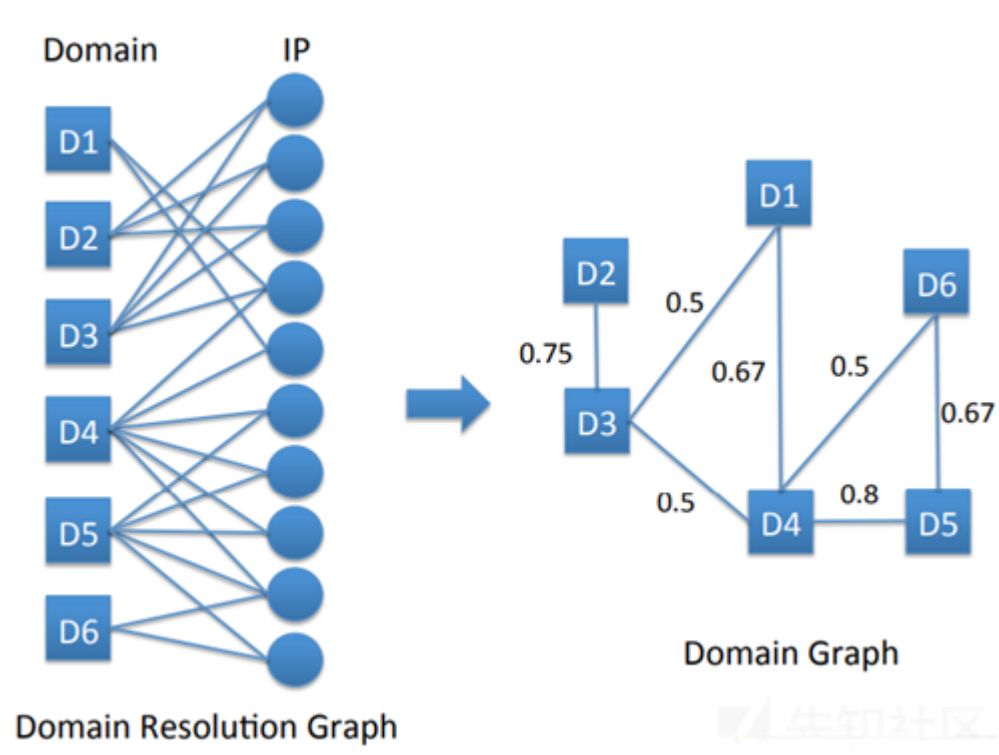
讲到DNS数据集，不得不提的是Passive DNS。我查阅了很多互联网博客资料，发现并没有哪个对Passive
DNS的概念描述得很清楚，导致大多数人会误以为Passive DNS和之前提供的DNS解析数据是差不多的，但是还是略有不同。Passive
DNS，顾名思义其实就是被动获取的DNS数据信息。它是由一些研究者放入互联网当中的流量抓取工具抓到的DNS数据信息，然后存储到一个中央数据库，形成了Passive
DNS
database。这种数据信息和一些本地DNS数据不同的是，它包含的不仅是当前的DNS数据（包括IP映射等），还包括了历史上所存在的一些DNS数据映射等。研究人员可以

一个域名在某个时间段内可以解析到多个IP地址上，一个IP地址也可以某个时间段内映射在多个域名，这些映射历史都是可以被Passive
DNS数据包含在内，这样研究者可以从这些关联数据中发现已知的恶意域名与未知恶意域名之间的关系，从而挖掘出未知的威胁风险。

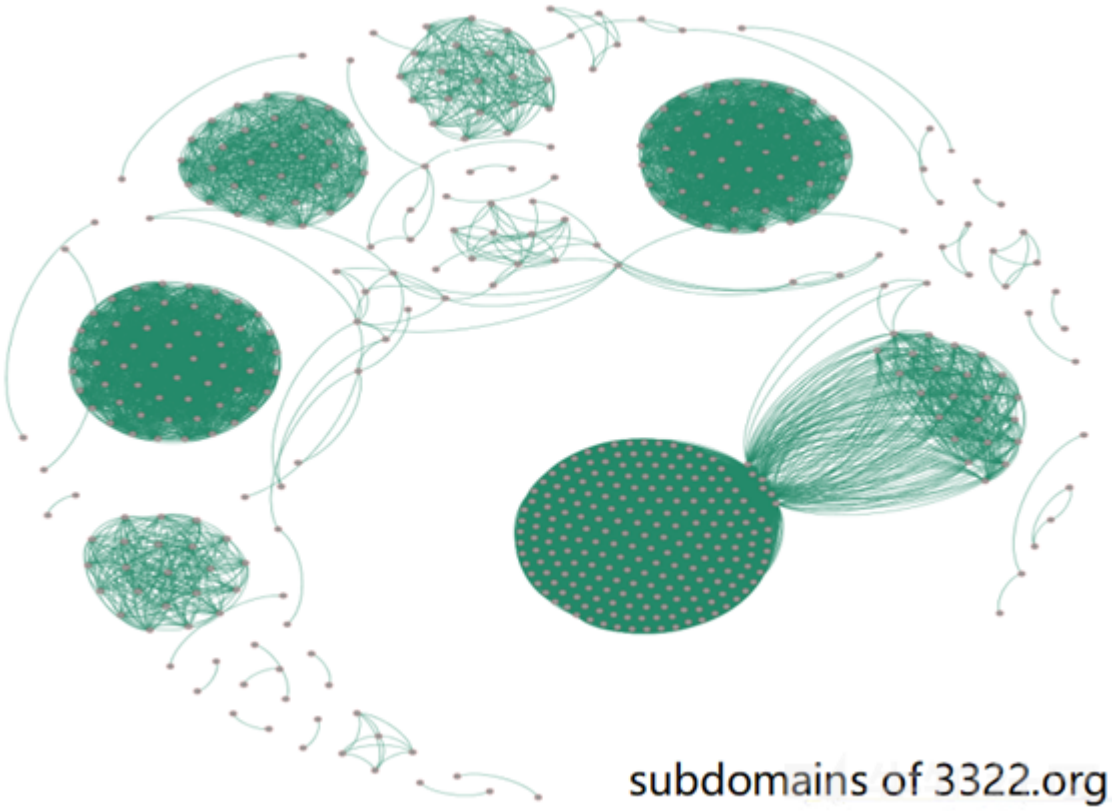
所以通过Passive DNS来构建域名间联系，可以通过定义如下公式来定义该联系的权重。

$$w(d_1,d_2)=\begin{cases}1-\frac{1}{1+|ip(d_1)\cap ip(d_2)|} & \text{if } d_1\neq d_2\\ 1 & \text{otherwise}\end{cases}$$

将该公式实验应用到下面这个域名解析图中，去计算每个域名节点之间的联系权重。举个例子，计算D1和D3之间的联系权重，D1和D3的IP映射当中有一个IP是重叠的，且得



将该公式用于计算3322.org这个著名的恶意域名的二级域名之间的联系关系，并将关系图绘制出来可以发现，二级域名间存在一个团体性的关系联系，这往往给我们对恶意域



利用上面的权重关系可以进一步分析一些未知域名与著名恶意域名之间的联系权重，从而判断未知域名是否具有恶意性。但是问题在于这些需要进行判断的未知域名与著名恶

1. D2-D3-D1-D4-D5-D6；
2. D2-D3-D1-D4-D6；
3. D2-D3-D4-D5-D6；
4. D2-D3-D4-D6；

如果从直观上选择一般选择路径最短，但是首先这里会定义一个联系关系函数assoc(P)，assoc(P)要求必须是每次经过路径的最大值，因为最大值路径代表了最强关联性，可

$$assoc(d_1, d_2) = max_{1 \le i \le k} w(P_i)$$

所以选择路径Path
3：D2-D3-D4-D5-D6。那么该公式中的w(P)，即路径权重是由edge的值相乘得到，但path越长，推定确定性越低，所以没经过一个hop需要打折一次。即如下公式：

$$w(P) = \prod_{1 \le i \le n-1} w(d_i, d_{i+1})$$

因为要通过著名恶意域名去判断未知域名，所以要设立一个关于已知恶意域名的对象seed，多个seed形成一个S集合。最终就是通过seed来计算未知域名的恶意性可能性值。

定义一个列表M(d)，这是一个seed与未知域名之间关系值assoc()函数的排序列表，即 (assoc(s1, d), . . . , assoc(sn,
d))，列表排序说明了在列表第一个的assoc(s1,d)说明d与s1的关联系最高，这里是通过与一个恶意域的强关联和与其他恶意域的弱关联从双面判定该域名与指定恶意域的强关

$$mal(d, S) = assoc(s_1, d) + \left(1 - assoc(s_1, d)\right) \sum_{i=2,\ldots,n} \frac{1}{2^{i-1}} assoc(s_i, d))$$

可是如上的公式推算忽略一个情况，就是公有云的问题。像阿里云这些可能由于云厂商拥有的是有限的IP地址池，所以可能有些主机域名因为厂商的缘故会共享使用或者不同

简单粗暴，直接将这类IP地址剔除不进行计算（不现实，且最后推算出的公式不通用）；

如果一个IP在一段时间内被映射到大量的域名，那么它很可能是一个公共IP。因此，如果IP在某个时间段内承载超过t个域名，我们将排除这些IP，其中t是一个可配置参数

重新定义域名间联系权重w(d1,d2)；

定义一个asn(I)，I为IP地址集合，那么asn(I)表示这个集合中IP地址归属的自治系统域（AS）的编号，这里默认认为一个服务提供商只会提供一个AS内的IP地址。但实际

$$w(d_1, d_2) = 1 - \frac{1}{1 + |asn(ip(d_1) \cap ip(d_2))|} \quad \text{if } d_1 \neq d_2$$

DNS Rebinding

DNS重绑定是一种比较常见的攻击形式，1996年该技术扰乱了JVM自身的安全措施。该技术的主要核心就是绕过SOP的限制来进行一些内网威胁攻击。基本过程如下：

1. 第一次DNS解析请求到外部合法域名绕过安全限制；
2. 第二次DNS解析请求发现数据变更请求到内网地址。

这种技术可以用于防火墙绕过或者IP劫持等。那么一般如果要实现这种技术利用，可以建立在好几种不同的基础原理上

1. 给同一个主机域名配置多个DNS的A记录（1996年攻击JVM的手法）；
2. 极短的TTL老化时间；
3. 不同浏览器中的Pinning设置时间不一样导致可以通过在部署弱安全措施的浏览器上实施攻击（DNS Pinning即浏览器会在自身设置好的缓存时间前使用自己缓存库中的IP地址或域名，这期间不信任不依赖TTL老化时间）；
4. Flash的crossdomain.xml不够严谨（由于Flash将死，不详述）；

针对这种攻击技术的防护可以从三个方面进行部署：

防火墙绕过层面：

1. 关闭企业对外的53端口禁止内网对外部DNS服务器进行请求，然后在本地DNS服务器上进行外部域名映射到内网IP地址；
2. 用户的个人PC防火墙可以加上dnswall的部署；
3. Windows的防火墙可以阻断对127回环IP地址的DNS解析请求；

插件安全

1. Flash Player需设置严谨的policy文件且需在socket连接前先通过policy的安全检查。
2. Java需使用CONNECT方法对外连接；
3. Java LiveConnect需使用和浏览器统一的缓存池以移除不同Pinning时间导致漏洞利用；

浏览器安全

如果插件默认拒绝socket连接

检查请求包Host字段
因为javascript标准不允许xhr自定义host字段，所以通过检查该字段可以防止在一个IP地址上承载多个主机域名。

A **forbidden header name** is a header name that is a byte-case-insensitive match for one of

- `Accept-Charset`
- `Accept-Encoding`
- `Access-Control-Request-Headers`
- `Access-Control-Request-Method`
- `Connection`
- `Content-Length`
- `Cookie`
- `Cookie2`
- `Date`
- `DNT`
- `Expect`
- `Host`
- `Keep-Alive`
- `Origin`
- `Referer`
- `TE`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade`
- `Via`

源信息细粒化
其实就是收集关于目标信息的一些细粒化的信息，包括其公钥信息等，再次请求时为了防止目标被篡改那么就会比对这些额外信息是否匹配，如果不匹配那么有可能遭受

智能化Pinning
服务商一般会降低TTL来提高应用健壮性，但牺牲了安全性，使其可以被DNS重绑定攻击。所以使用“C类固定”，只允许重绑定源目标的同C类地址。同时RFC1938标准文

基于协议的Pinning
浏览器在重绑定时参考目标服务器部署的一些policy措施（例如，crossdomain.xml和部署在reverse DNS上的policy）。

避免Pinning缺陷
浏览器和插件共享同一个pin数据库，防止出现缓存老化时间差异；且Cache中必须同时存储URL和IP地址以防篡改；类似document.domain=document.domain这种j

如果插件默认允许socket连接

- 主机名授权

服务器可以公布主机白名单，允许主机映射这个白名单内所有IP地址。例如授权www.example.com使用171.64.78.146。那么就会使用如下的DNS records。

```
auth.146.78.64.171.in-addr.arpa.
    IN A 171.64.78.146
www.example.com.auth.146.78.64.171.in-addr.arpa.
    IN A 171.64.78.146
```

具体使用过程如下：

1. 解析域名auth.ip.in-addr.arpa。
2. 如果域名存在，且IP地址是符合白名单策略的。那么就允许授权，反之不行。

最后，如果IP是符合白名单策略的话，那么根据www.example.com.auth.[ip].in-addr.arpa是否可以解析成功来确定域名（www.example.com）是否是被授权的。

可信任的策略提供商

使用可信任的策略提供商提供的接口进行主机名授权，便捷安全。

对抗升级

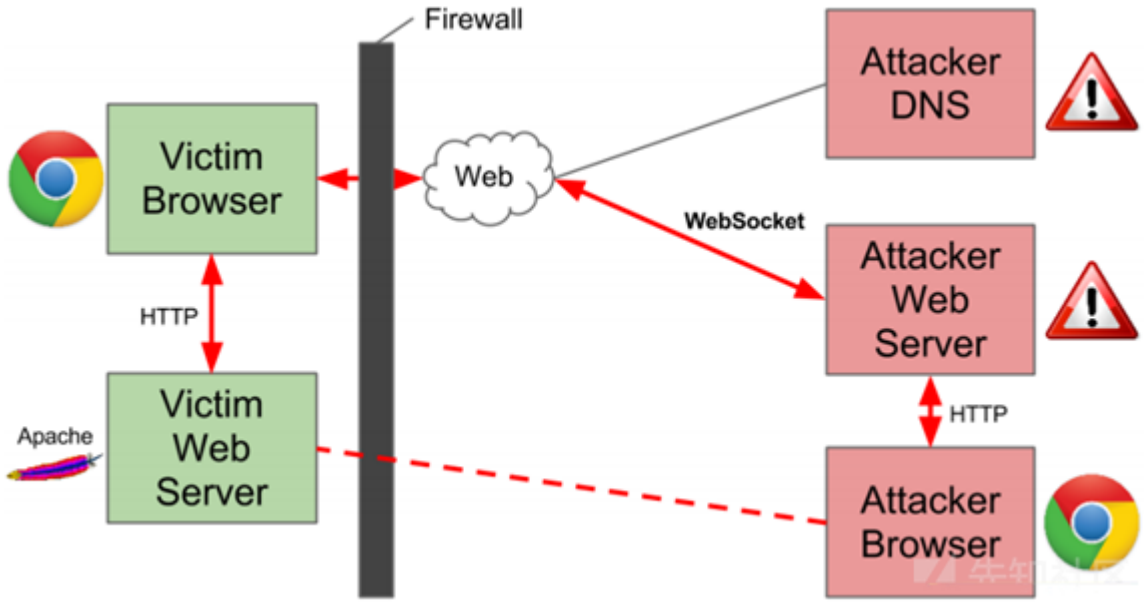
上文中所提到的安全防护技术中最为推崇的应该就是DNS Pinning技术了，其实针对这类Pinning技术也有一定的绕过方式。

Http DoS。

浏览器在第二次请求目标域名时去请求缓存中的IP地址，但此时只要攻击者使目标主机拒绝服务，即拒绝新的连接，那么浏览器就会放弃使用缓存中的数据而且建立新的Rebind后的新的IP地址（也就是内网IP地址）；

泛洪攻击DNS缓存表同时实现交互式DNS Rebinding

第二点中提到的方式主要可以用下图来表示。



攻击者控制一台Web服务器和DNS服务器，当用户访问attack.com时，DNS服务器意识到该主机是第一次请求解析这个域名，则它会返回攻击者控制的Web服务器的IP地址

且这个javascript脚本可以与攻击者的web服务器维持一个长期的WebSocket连接，并且从web服务器接收json格式的指令，指令包含三个字段：method（例如，POST方法
url:/form/login',
args:name='alice'}指的就是向/form/login地址post发送数据，数据内容为name=alice，而javascript脚本会通过xhr来执行这个命令。如果请求需要下载一个二进制文件，

由于目的是交互式攻击，那么还有一个难题就是使用户驻留，在这个攻击下其实就是让攻击者不关闭加载了javascript脚本的页面。那么攻击者可以构造一个诱惑页面使用户

针对这种交互式DNS重绑定攻击，有三种防护方式

头字段检测

浏览器应检测Host字段是否匹配（默认未开启或不阻断）。

增加缓存大小

不能完全阻止该攻击，且可能会影响性能。

智能化缓存清洗

在DNS缓存进行清洗时应该首先清洗只能识别出的无效无意义DNS条目。

带外数据攻击

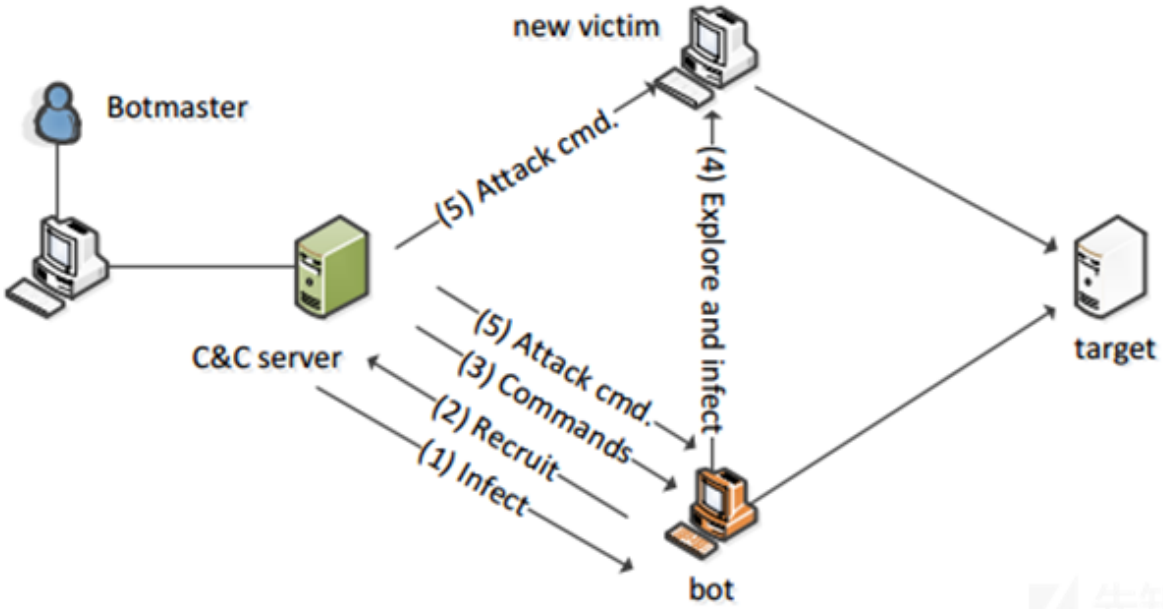
这个是当前一些渗透测试技术中比较常见的攻击形式，常用语SQL注入，XSS攻击，XXE等。也经常在利用一些新漏洞发现初期使用这种方式获得攻击回显，例如Weblogic的

DNS Tunnel

这是最近几年常被拿出来提的用于高级类攻击的一种技巧，一般用于后门连接C&C服务器时使用，近几年的wannacry，xshellghost等后门都使用了这种隧道攻击技术。这种
Tunnel当中。如果只是A记录或者AAAA记录，那么我们只可以从部署的后门获取到敏感数据，而部署在沦陷机上的后门或病毒并不能得到不断更新的指定命令，也就是说所

Host	record type	value
1.1.1.0	PTR	0x990xa50x330xd40xc90x310xbb0x750x000x000xff.1.com
1.1.1.1	PTR	0xe90xa50x310xd40xcb0x010xbb0x750xcc0x010xef.1.com
1.1.1.253	PTR	10min5delay.com
1.1.1.254	PTR	0min0delay.com
TimeforReconnect	A	1.1.10.5
1.1.1.0	PTR	0x990xa5.1.com
1.1.1.1	PTR	0x330xd4.1.com
1.1.1.2	PTR	0xc90x31.1.com
1.1.1.3	PTR	0xbb0x75.1.com
1.1.1.4	PTR	0x000x000xff.1.com

和PTR记录一样，TXT记录也可以用于交互攻击。攻击者可以使后门读取指定的域名的TXT记录，这些记录完整地写好了命令，执行即可，形式上和PTR没什么差别。所以交



这种隧道技术常见于恶意软件，例xshellghost就是用了DNS Tunnel技术。

如果把这种复杂的隧道技术运用到日常的渗透测试攻击当中。以XSS为例，在我们插入的XSS Payload当中，我们可以使用xhr去请求一个可控的web api，这个api可以读取对应域名的TXT记录从而读取最新的javascript脚本，然后也是通过xhr或者preload机制访问可控域名，将敏感数据放入二级域名，例如"[sesitiveDat
Tunnel技术帮助我们实现了增强型交互式XSS攻击。

```php
<?php
$target = $_GET["domain"];
$txt = dns_get_record($tagret, DNS_TXT);
echo $txt;
?>
```

【题外话】

除了DNS协议可以建立Tunnel来传输数据，ICMP协议同样也可以，如下提供了两例已开源的ICMP Tunnel工具：

1. MIT Ptunnel。http://www.mit.edu/afs.new/sipb/user/golem/tmp/ptunnel-0.61.orig/web/
2. inquish icmpsh。http://inquisb.github.io/icmpsh/

致谢

ourren@Secwiki

sshruoshui(warethink#gmail.com)@NUTD

本文首发于安全学术圈（SecQuan）公众号。本公众号分享安全方面的论文写作、会议发表、基金申请方面的资料。

参考文献

[1] Khalil, I., Yu, T., & Guan, B. (2016, May). Discovering malicious domains through passive DNS data graph analysis. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (pp. 663-674). ACM.

[2] Jackson, C., Barth, A., Bortz, A., Shao, W., & Boneh, D. (2009). Protecting browsers from DNS rebinding attacks. ACM Transactions on the Web (TWEB), 3(1), 2.

[3] Dai, Y., & Resig, R. (2013). FireDrill: Interactive {DNS} Rebinding. In Presented as part of the 7th {USENIX} Workshop on Offensive Technologies.

[4] Stødle, D. (2005, May 26). Ping Tunnel - Send TCP traffic over ICMP [Web log post]. Retrieved June 20, 2019, from http://www.mit.edu/afs.new/sipb/user/golem/tmp/ptunnel-0.61.orig/web/

[5] inquish. (2013, May 17). icmpsh - Simple reverse ICMP shell. Retrieved August 1, 2019, from http://inquisb.github.io/icmpsh/

上一篇：在互联网端口扫描过程中寻找速度和准... 下一篇：Apache Solr Injec...

1. 0 条回复
   - 动动手指，沙发就是你的了！