软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(10):4876−4911 [doi: 10.13328/j.cnki.jos.006987]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

递归侧 DNS 安全研究与分析*

张 宾[1], 张 宇[1,2], 张伟哲[1,2]

[1](鹏城实验室, 广东 深圳 518055)

[2](哈尔滨工业大学 网络空间安全学院, 黑龙江 哈尔滨 150001)

通信作者: 张宇, E-mail: yuzhang@hit.edu.cn

摘　要: 因特网用户在访问网络应用前都需要通过 DNS 进行解析, DNS 安全是保障网络正常运行的第 1 道门户, 如果 DNS 的安全不能得到有效保证, 即使网络其他系统安全防护措施级别再高, 攻击者也可以通过攻击 DNS 系统使网络无法正常使用. 目前 DNS 恶性事件仍有上升趋势, DNS 攻击检测和防御技术的发展仍不能满足现实需求. 从直接服务用户 DNS 请求的递归解析服务器视角出发, 将 DNS 安全事件通过两种分类方法, 全面梳理和总结 DNS 工作过程中面临的安全问题, 包括由攻击或系统漏洞等引起各类安全事件, 各类安全事件的具体检测方法, 各类防御保护技术. 在对各类安全事件、检测和防御保护技术总结的过程中, 对相应典型方法的特点进行分析和对比, 并对未来 DNS 安全领域的研究方向进行展望.

关键词: 域名系统 (DNS); 域名系统安全; 攻击检测; 系统防御

中图法分类号: TP309

中文引用格式: 张宾, 张宇, 张伟哲. 递归侧DNS安全研究与分析. 软件学报, 2024, 35(10): 4876–4911. http://www.jos.org.cn/1000-9825/6987.htm

英文引用格式: Zhang B, Zhang Y, Zhang WZ. Study and Analysis of Recursive Side DNS Security. Ruan Jian Xue Bao/Journal of Software, 2024, 35(10): 4876–4911 (in Chinese). http://www.jos.org.cn/1000-9825/6987.htm

Study and Analysis of Recursive Side DNS Security

ZHANG Bin[1], ZHANG Yu[1,2], ZHANG Wei-Zhe[1,2]

[1](Pengcheng Laboratory, Shenzhen 518055, China)

[2](School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Internet users need to resolve through DNS before accessing network applications. DNS security is the first portal to ensure the normal operation of the network. If the security of DNS cannot be effectively guaranteed, even if the level of security protection measures of other network systems is high, attackers can attack the DNS system to make the network unusable. At present, DNS malignant events still have an upward trend, and the development of DNS attack detection and defense technology still cannot meet practical needs. From the perspective of recursive servers that directly serve users' DNS requests, this study comprehensively summarizes the security problems faced in the DNS process through two classification methods, including various security events caused by attacks or system vulnerabilities, different detection methods for various security events, and various defense and protection technologies. When summarizing various security events, detection and defense protection technologies, the study analyzes the characteristics of the corresponding typical methods and prospects for the future research direction of the DNS security field.

Key words: domain name system (DNS); DNS security; attack detection; system protection

DNS 域名系统 (domain name system) 是一种互联网协议, 提供域名和 IP 转换机制, 它将易读的域名转换为 IP 地址, 将 IP 地址转换回域名, 从而保障网络应用的顺利执行. 随着互联网日新月起的快速发展, DNS 服务已经

* 基金项目: 鹏城实验室重大攻关项目 (PCL2023AS4-1); 国家自然科学基金青年科学基金 (62102202); 广东省重点领域研发计划 (2020B0101360001); 广东省基础与应用基础研究重大项目 (2019B030302002)
收稿时间: 2022-12-05; 修改时间: 2023-03-01, 2023-04-28, 2023-06-09; 采用时间: 2023-06-12; jos 在线出版时间: 2023-11-01
CNKI 网络首发时间: 2023-11-01

深入到互联网的各个角落, 成为互联网最为重要的基础服务和不可或缺的关键一环, 被称为互联网的“中枢神经系统”. 作为互联网的基础核心服务, DNS 渗透到网络的各个角落. 一旦遭受攻击, 会给整个互联网带来无法估量的损失. DNS 安全问题是保障目前网络安全亟待解决的首要问题.

DNS 主要由域名空间及资源记录、名字服务器、解析器 3 个部分组成, 如图 1[1]所示, DNS 通过层次树形结构的域名空间组织 DNS 域名, 具体的 DNS 请求和响应报文用资源记录表示, 解析器响应用户请求并从名称服务器获取查询结果, 完成用户 DNS 请求服务的称为 DNS 名字服务器, 分为递归解析服务器和权威名字服务器, 在整个域名解析过程中, 递归解析服务器起着承上启下至关重要的作用, 它直接面向用户, 接管用户查询请求, 并通过对不同层级的名字服务器的迭代查询完成对用户的 DNS 请求应答, 本文所述的“递归侧”是指递归解析服务器和用户间, 及递归解析服务器和权威服务器间的整个交互层面, 因此, 递归侧安全是用户能否正确得到 DNS 请求的关键, 目前的大部分 DNS 攻击事件都和递归侧安全有关, 递归侧安全越来越受到各国和各大 DNS 安全公司的重视[2].

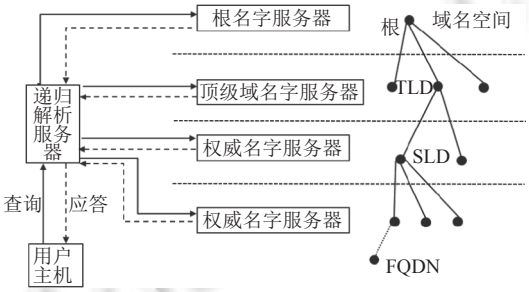

图 1 DNS 系统结构[1]

目前已经有很多工作对 DNS 相关技术进行了综述, 文献 [3] 从组件、结构、流量、安全 4 个方面对 DNS 测量研究工作进行了综述, 文献 [4] 侧重从 DNS 协议脆弱性出发, 对 DNS 协议设计、系统实现、检测监控和去中心化等方面进行了综述, 文献 [1] 从协议脆弱性、实现脆弱性和操作脆弱性 3 个方面对 DNS 系统面临的威胁进行了分类, 并对相应的改进方案进行了综述, 文献 [5] 从安全增强、行为监测、隐私保护等方面对 DNS 安全问题进行了总结和讨论. 国外也有很多工作[6−12]对 DNS 安全问题进行了总结和分析. 然而, 基于递归侧安全综述和分析仍是目前的空白.

本文从递归侧安全出发, 从两个不同的维度对 DNS 安全事件进行分类, 对目前 DNS 面临的安全问题进行深入的剖析, 并对相应的检测技术和 DNS 安全保护技术进行总结, 以帮助科研人员和工程人员更加透彻了解 DNS 工作原理, 熟悉各类 DNS 相关攻击方法和相应的检测方法, 为更好地改进 DNS 攻击检测方法和改进 DNS 设计提供帮助. 本文第 1 节对 DNS 工作原理进行阐述. 第 2 节对 DNS 的安全现状进行总结分析. 第 3 节对 DNS 的安全事件进行分类. 第 4 节按照分类具体介绍 DNS 安全事件, 并总结相应的检测技术. 第 5 节对不同的 DNS 增强保护技术进行分析总结. 第 6 节对 DNS 安全研究工作进行未来展望. 最后进行简要总结. 为方便读者阅读, 本文在表 1 中对文中出现的英文缩略词进行汇总.

表 1 文中的缩略词一览表

英文缩略词	英文	中文
DNS	domain name system	域名系统
AS	autonomous system	自治系统
ASN	autonomous system number	自治系统编号
PC	personal computer	个人计算机
HTTP	hyper text transfer protocol	超文本传输协议
HTML	hyper text mark-up language	超文本标记语言
TCP	transmission control protocol	传输控制协议
UDP	user datagram protocol	用户数据报协议

表 1 文中的缩略词一览表 (续)

英文缩略词	英文	中文
WWW	world wide web	万维网
BGP	border gateway protocol	边界网关协议
C&C	command and control	命令和控制
DGA	domain generation algorithm	域名生成算法
CDN	content delivery network	内容分发网络
DoS	denial of service	拒绝服务
DDoS	distributed denial of service	分布式拒绝服务
DRDoS	distributed reflection denial of service	分布式反射拒绝服务
SSDP	simple service discovery protocol	简单服务发现协议
IP	Internet protocol	因特网协议
DNSSEC	DNS security extensions	DNS安全扩展
RR	resource record	资源记录
A	address	FQDN–>IPv4地址记录
AAAA	address	FQDN–>IPv6地址记录
PTR	pointer	IP–>FQDN
CNAME	canonical name	别名记录
SOA	start of authority	起始授权记录
NS	name service	域名服务记录
MX	mail exchanger	邮件交换器记录
NXDOMAIN	none exist DOMAIN	不存在的域名
TLS	transport layer security	传输层安全
SLD	second level domain	二级域名
TTL	time to live	时间生存值
DoH	DNS-over-HTTPS	基于HTTPS的DNS
DoT	DNS-over-TLS	基于TLS的DNS
TLD	top level domain	顶级域名
IETF	Internet engineering task force	互联网工程任务组
ISP	Internet service provider	互联网服务提供商
NDN	named data networking	命名网络
RFC	request for comments	请求评议
RRL	response rate limiting	响应速率限制
FQDN	Fully Qualified Domain Name	全域名
IDC	Internet data center	互联网数据中心
ID	identify document	身份标识
SPF	sender policy framework	发送策略框架
DKIM	domain keys identified mail	域名关键字认证邮件
RF	random forest	随机森林
SVM	support vector machine	支持向量机
LSTM	long short-term memory	长短记忆网络
DT	decision tree	决策树
RPZ	response policy zones	响应策略区

1 DNS 工作原理

1.1 DNS 起源及功能

最初, 网络上的每个站点用 HOSTS.TXT 文件维护一个网络地址到网络设备名字的映射关系, 但随着网络规

模的发展, 用这样一个文件同步处理网络名字的爆炸扩展是非常困难的, 因此, 最初的探索者在 1983 年提出 DNS 来代替集中式管理的名字数据, 在 1987 年 DNS 进一步更新完善, 形成了目前在用的基本协议 RFC1034. 它的主要功能是将易读的域名转换为 IP 地址, 将 IP 地址转换回域名, 为互联网用户提供名字解析服务.

1.2 域名空间

DNS 域名空间负责分配不断增长的网络设备名称列表, 可以用一棵如图 2[12]所示层次树来表示, 其中根位于顶部, 用“.”表示, 第 2 层通常称为顶级域名 TLD, 第 3 层通常称为 SLD, 用来标明顶级域内的一个特定的组织, 如.cn 顶级域名下面设置的二级域名: .ac.cn、.edu.cn 等, 二级域下所创建的各级域统称为子域, 各个组织或用户可以自由申请注册自己的域名, 叶子位于底部, 叶子节点的域名为全域名 FQDN.

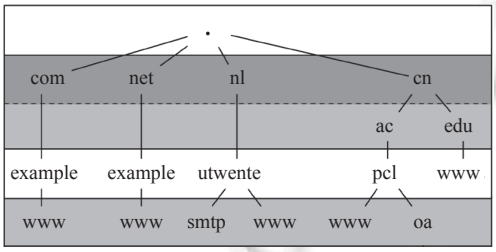

图 2 域名层次树示例图[12]

1.3 名字服务器

名字服务器负责将 IP 和域名进行转换, 通常分为递归解析服务器和权威服务器, 权威服务器的最顶层即为根服务器, 将 DNS 查询定向到每个顶级域的名称服务器, 这些顶级域是其正下方的主要分支 (例如, .com、.net). 根服务器是 DNS 根区的权威域名服务器, 对根区域具有权威解析. 其他权威服务器主要负责对所管辖域名或子域名的权威解析. 递归解析服务器直接面对用户, 负责将用户的 DNS 请求对权威服务器进行迭代查询, 并向用户返回最终解析结果.

权威域名服务器中使用区 (zone) 文件来管理具体的域名信息, 参与域名资源记录编写的服务器通常称为主名称服务器, 由于仅一台服务器不足以提供可靠的解决方案, 因此通过称为“域传输 (zone transfer)”的过程从主要名称服务器获取区域数据的副本. 这些附加服务器称为辅助域名服务器或从属域名服务器. 另外在域名服务过程中, 还存在一些只提供域名解析结果的缓存功能, 也存在一些只负责将域名转发的转发域名服务器.

1.4 DNS 报文

DNS 分为查询请求和查询响应, 请求和响应的报文结构基本相同. DNS 报文格式如表 2 所示.

表 2 DNS 报文格式

事物ID	标志 (QR, Opcode, AA, TC, RD, RA, Z, rcode)
问题计数	回答资源记录数
权威名称服务器计数	附加资源记录数
查询问题区域	
回答问题区域	
权威名称服务器区域	
附加信息区域	

如表 2 所示的 DNS 报文前面 6 个部分是基础结构部分, 第 7 个部分是问题部分, 最后 3 个部分是资源记录部分, 每个字段含义如下.

1) 事务 ID: DNS 报文的 ID 标识. 对于请求报文和其对应的应答报文, 该字段的值是相同的. 通过它可以区分 DNS 应答报文是对哪个请求进行响应的.

2) 标志: DNS 报文中的标志字段, 每个标志含义具体参见 RFC1034, 这里不再详述.

3) 问题计数: DNS 查询请求的数目.

4) 回答资源记录数: DNS 响应的数目.

5) 权威名称服务器计数: 权威名称服务器的数目.

6) 附加资源记录数: 额外的记录数目 (权威名称服务器对应 IP 地址的数目).

7) 问题部分指的是报文格式中查询问题区域部分. 该部分是用来显示 DNS 查询请求的问题, 通常只有一个问题. 该部分包含正在进行的查询信息, 包含查询名 (被查询主机名字)、查询类型、查询类.

8) 资源记录部分是指 DNS 报文格式中的最后 3 个字段, 将信息映射到公用域名. 包括回答问题区域字段、权威名称服务器区域字段、附加信息区域字段. 资源记录部分只有在 DNS 响应包中才会出现.

1.5 DNS 解析

如图 1 所示, 客户主机上的用户进行域名查询, 先向其内部的缓存进行查询, 如果没有, 则创建一个包含服务器名称的 DNS 递归查询消息, 并将其传输到配置中确定递归解析服务器, 递归解析服务器接收到查询后, 检查其资源记录, 以确定它是否是包含所请求服务器名称所在区域的权威源. 如果区域文件中有, 则返回结果, 该解析具有权威性. 如果没有, 则向缓存中查询, 如果没有, 则 DNS 服务器将生成一个迭代查询并将其提交到 13 台根服务器中的一台, 根服务器检查递归解析服务器请求的域名, 并查询其资源记录, 以确定该名称的顶级域名的地址. 然后根服务器向递归 DNS 服务器发送应答.

递归解析服务器生成一个新的迭代查询并将其传输到顶级域服务器, 顶级域服务器检查请求名称中的二级域, 并将包含该二级域的权威域名服务器地址的引用传送回. 递归解析服务器生成另一个迭代查询, 并将其传输到二级域服务器, 如果二级域服务器是包含请求名称的区域的权威服务器, 那么它会查询其资源记录以确定请求域名的 IP 地址, 并将其以应答消息的形式发送回该递归解析服务器. 递归解析服务器将 IP 地址传回客户机系统上的解析器, 同时缓存被请求系统的 IP 地址和特定域的权威服务器的地址. 解析器将地址转发给应用程序, 然后应用程序可以与用户指定的服务器进行 IP 通信.

从 DNS 解析过程中我们可以看到, 递归解析服务器在解析过程中占有至关重要的位置, 是用户进行解析请求的门户, 正常情况下, 由于缓存功能, 大约 80% 的域名解析在递归解析服务器处都可以完成了, 只有约 20% 左右的请求需要递归解析服务器继续向上迭代查询, 在 DNS 安全事件中, 除了对用户和权威服务器自身进行攻击外, 其他攻击基本和递归解析服务器相关, 因此, 递归侧的安全渗透到 DNS 安全的各个角落.

2 DNS 安全现状分析

近年来针对和利用 DNS 的攻击形势也愈加严峻. 根据 2021 年度的 IDC[2]统计, 2021 年度全球约有 87% 的组织遭受 DNS 攻击, 每个组织遭受 DNS 攻击的平均数量为 7.6, 单次攻击的平均损失为 95 万美元, 其中主要的攻击事件有: DNS 钓鱼 (49%), DNS 相关的恶意软件 (38%), DDoS (29%), DNS 劫持 (27%), DNS 隐秘隧道 (24%), 零日脆弱性 (23%), 错误配置滥用 (23%). 对用户造成信息泄露、生意亏损、云服务器下线、应用下线、商标破坏、网站服务破坏等损失. 尽管 DNS 安全是整个网络的核心环节, 但是仍有 42% 的公司没有特定的 DNS 安全解决途径来填补传统网络安全产品遗留的空白.

表 3 总结了到目前为止互联网发生的重大的 DNS 安全事件. 从表 3 中我们可以看到, 这些重大的 DNS 攻击事件, 不仅存在对全球根服务器的攻击, 还有对顶级域的攻击, 对大的 DNS 服务商的攻击及对全球重要知名网站的攻击, 攻击造成全球性或地区重要性事件.

目前已经有很多国家和企业重视 DNS 安全, 试图在一些层面解决 DNS 安全问题, 表 4 是一些发达国家 DNS 的安全系统, 美国最早完成了 DNS 安全系统的攻防布局, 量子 DNS[13]作为他们量子系统的重要一环, 很早就用于对网络流量实施劫持, 很容易对他国网络用户带来安全问题, 而针对.GOV 的重要域名[14], 他们实施了防护技术, 可以有效防护特征重要域名的劫持风险, 主要进行系统级别防护, 但项目并不对用户访问的恶意域名进行阻断, 只有当攻击威胁到特定重要域名时才进行保护.

表 3 DNS 重要安全事件一览表

时间	攻击类型	攻击目标	影响后果
2009.5.19	DDoS	dnspod	由游戏私服私斗打挂dnspod爆发的6省大规模断网的事故
2009.11.24	DoS	UltraDNS服务提供商	对DNS服务提供商UltraDNS发起DDoS攻击, 致使其服务的大型电子商务网站, 包括Amazon, Walmart, Expedia大约一个小时的服务拥塞
2010.1.12	DNS劫持	百度	百度的NS记录被伊朗网军劫持, 然后导百度无法访问达8 h
2011.9.5	DNS劫持	众多网站	DNS劫持了微软、宏碁、UPS在内的众多知名网站
2013.3.18 – 2013.3.19	DRDoS	Spamhaus网站	一系列攻击产生了大约300 Gb/s的网络流量对Spamhaus网站进行攻击, 造成服务拥塞
2013.8.25	DDoS	.cn	cn域DNS受到DDoS攻击而导致所有cn域名无法解析
2015.11.30– 2015.12.1	DDoS	根服务器	13个根服务器大都受到了攻击, 攻击者对根服务器发起了针对两个特定域名的数十亿次无效查询请求
2015.12.14	DDoS	土耳其国家域	黑客组织匿名者宣布自己是40 Gb/s DDoS的网络攻击发起人, 并表示该攻击跟反ISIS行动相关
2016.9.20	DRDoS	KrebsOnSecurity.com	攻击者利用DNS服务器反射产生了大约665 Gb/s的网络流量, 对KrebsOnSecurity.com网站进行了攻击, 造成网站瘫痪
2016.10.21	DDoS	Dyn	利用Mirai Botnet产生大约1.2 Tb/s网络流量对Dyn的Anycast的服务器进行DDoS攻击, 造成Twitter、 Tumblr、 Facebook、 CNN 在内的许多网站无法登陆, 是互联网上目前发生的最大的一次DDoS攻击
2017.5.12	WannaCry (永恒之蓝) 蠕虫病毒	150个国家200 000终端	WannaCry 蠕虫病毒有一个开关域名, 在蠕虫感染过程中, 有效载荷通过 445 端口上的漏洞投递并成功启动后, 会尝试访问特定域名的网页, 如果访问失败, 蠕虫会开始破坏动作, 并随后弹框勒索赎金
2021.10.4	错误配置滥用	Facebook	Facebook 及其旗下Instagram 和 WhatsApp 等应用全网宕机近7 h, 浏览器在尝试打开时显示 DNS 错误, 此次故障的根本原因是例行维护工作时发出的一条指令导致其DNS服务器不可使用.
2022.2.16	DDoS	域托管服务Point DNS	域托管服务Point DNS 负责全球超过 220 000 个域名, 在其所有DNS服务器遭到严重DDoS攻击后, 遭遇重大中断

表 4 各国重要 DNS 安全系统

各项目比较	DNS攻防系统	递归解析服务器	主要目的
美国量子DNS-QUANTUMDNS	攻击	包含递归解析服务的全体系攻击系统	对互联网流量实施劫持
美国DotGov项目	防御	包含递归解析服务基于特定域名的全体系防护系统	保护特定域名的流量劫持
俄罗斯RuNet	防御	包含递归解析服务的全体系防护系统	本国内网络的自主运行防护
欧盟 DNS4EU	防御	主要基于递归解析服务器的防御系统	防止区域外的DNS垄断, 同时保障区域内解析的安全性
加拿大CIRA Canadian Shield	防御	基于递归解析服务的防御系统	为本国内用户DNS安全解析服务, 阻断恶意域名

俄罗斯的 RuNet[15]主要为了完成国家网络独立自主的运行, 在切断和其他国家的连接时, 仍能为俄罗斯国内用户提供自主解析和自主网络的服务, RuNet 不仅在 DNS 层面, 在整个网络层面都为国家提供了安全防护, DNS 解析层面为互联网的域名系统提供了备份服务, 相当于在俄罗斯国内又建了一套独立的解析系统, 把整个互联网的解析系统备份下来, 从公开的资料看, RuNet 主要为了保障本国网络在切断外部网络后, 可以独立运转, 并不提供恶意域名阻断层面的安全措施.

欧盟 DNS4EU 提案[16]主要为了防止外部公司对欧盟国家的 DNS 垄断, 意在为欧盟成员国用户提供稳定安全的 DNS 递归解析服务, 防止用户被其他外部的 DNS 递归解析服务垄断, 本质上是提供一个安全稳定的 DNS 递归解析服务, 同时具有恶意域名阻断功能, 和违禁内容过滤功能, 在保障正确解析的同时提供一定的安全措施. 加拿大 CIRA Canadian Shield[17]和欧盟的 DNS4EU 类似, 不同之处在于, 加拿大 CIRA Canadian Shield 已稳定为加拿大

用户提供了 20 年服务, 而欧盟的 DNS4EU 还在提案阶段, 还未形成系统实施.

总体上讲欧盟的 DNS4EU 和加拿大的 CIRA Canadian Shield 在递归解析层面为区域用户提供解析服务, 同时提供 DNS 安全服务, 以免访问恶意域名和违禁内容; 俄罗斯的 RuNet 为本国提供独立的 DNS 解析 (含递归和各层权威解析) 和网络服务, 美国的量子 DNS 系统实施 DNS 劫持攻击手段, 可以对其他国家造成 DNS 安全问题, 而美国的 DotGov 项目同时保障了自己国家重点域名安全, 防止劫持. 从各国的 DNS 安全项目中我们可以看到, 在递归层面的安全布局已经成为各国的 DNS 安全防护布局方向.

3 递归侧 DNS 安全分类

鉴于本文所述的“递归侧安全”是指递归解析服务器和用户及和权威服务器间的整个交互层面的安全问题, 我们在对 DNS 安全问题进行分类时将仅涉及权威服务器的安全事件筛选掉, 而保留其他 DNS 安全事件, 比如权威服务器的域名撤销类等风险事件不在本文范围内. 对 DNS 安全问题一种分类方法是基于攻击目标的不同进行分类, 如图 3 所示, 具体有以下类型.

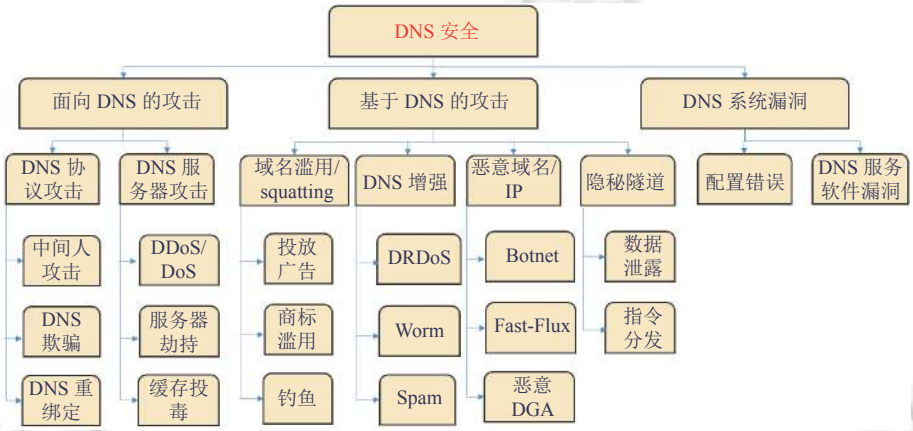

图 3 DNS 安全事件分类

(1) 面向 DNS 的攻击, 主要攻击特征为攻击 DNS 本身, 包括 DNS 协议和 DNS 服务器, 其主要思路是利用 DNS 协议或服务器的弱点, 通过攻击 DNS 服务或服务器的方式来达到攻击使用 DNS 服务的其他网络业务的目的. 针对 DNS 协议的攻击具体通常有中间人攻击、DNS 欺骗、DNS 重绑定等攻击方式, 针对 DNS 服务器的攻击通常有针对名字服务器的 DDoS/DoS 攻击、服务器劫持、缓存投毒等攻击方式.

(2) 基于 DNS 的攻击, 主要攻击特征为利用 DNS 进行攻击, 主要攻击对象为 DNS 服务系统以外的用户或网络目标, 其主要思路是通过恶意代码利用 DNS 通道, 实现与远程控制中心的通信, 从而执行传输窃取到的数据、获取攻击指令和执行攻击指令等恶意操作. 基于 DNS 的攻击可分为域名滥用、对其他网络目标的 DDoS/DoS 攻击、恶意域名/IP、隐秘隧道等攻击方式, 域名滥用通常指域名拼写等问题造成的误用, 从而造成投放广告、商标滥用、钓鱼等恶意攻击, 而对其他网络目标的 DNS 增强攻击通常指利用 DNS 系统对目标进行 DoS 反射放大攻击、蠕虫传播、邮件炸弹等, 恶意域名/IP 具体通常有 Botnet、快速变换 (Fast-Flux)、恶意 DGA 等攻击, 隐秘隧道是指通过 DNS 跨越防火墙进行数据泄露、指令分发等攻击.

(3) DNS 系统本身的漏洞及配置问题, 主要是指由于系统管理员配置错误区文件从而引起的解析错误, 或者部署的 DNS 系统软件本身的漏洞问题引发的 DNS 错误. 主要包括配置错误和 DNS 服务软件漏洞.

从信息安全的维度出发, 我们可以对 DNS 攻击分为 4 类.

(1) 对 DNS 信息保密性的攻击: 这类攻击主要针对 DNS 查询和应答信息保密性的攻击, 绝大多数的 DNS 信息在传输过程中都是明文传输, 很容易进行中间人攻击、DNS 欺骗、DNS 重绑定等攻击, 从而引起信息泄露和信

息篡改, 另外对服务器的劫持和缓存投毒也容易造成信息的篡改.

(2) 对 DNS 信息完整性的攻击: 这类攻击主要针对服务器返回信息的完整性进行攻击, 即使解决了信息传输的保密性, 但 DNS 响应的验证并不能保证, 攻击者可以通过缓存投毒、中间人攻击的报文拦截、DNS 欺骗进行的虚假响应等来改变响应中域名和 IP 的对应关系, 从而使 DNS 请求重定向, 也可以通过 DNS 服务器劫持来直接改变区文件或者将 DNS 请求劫持到另外一个恶意的服务器上, 另外, 用户配置导致的区文件错误和 DNS 服务软件本身的系统漏洞, 都可以造成对 DNS 信息完整性的攻击.

(3) 对 DNS 服务设施可达性的攻击: 这类攻击主要针对 DNS 服务设施进行攻击, 从而导致 DNS 服务的延缓或者不可用, 信息保密性和完整性只有当 DNS 服务可用时才有意义, 如果 DNS 服务设施不可达, 用户无法访问互联网的任何服务, 这类攻击主要是针对 DNS 服务器攻击的 DDoS/DoS, 蠕虫等造成服务器服务阻塞或者瘫痪的攻击.

(4) 对 DNS 可靠性的攻击: 这类攻击是利用 DNS 服务对其他网络设施和用户进行的攻击, 即 DNS 误用使得 DNS 违背了它自身的用途和功能, 从而对 DNS 的可靠性造成了攻击, 对应于第 1 种分类方法中基于 DNS 的攻击, 通过 DNS 服务进行秘密的恶意数据传送, 攻击源隐藏, 使得恶意攻击者隐藏在 DNS 服务背后, 更加难以检测, 无形中增加恶意事件传播的隐匿程度.

从上面对 DNS 安全的两种分类学中我们可以看到, 第 1 种分类中的所有 DNS 攻击也都可以对应到第 2 种分类中, 后续, 我们将按照第 1 种分类方法中的顺序详细阐述各类 DNS 攻击及对应的检测方案.

4 DNS 异常及检测技术

4.1 DNS 协议攻击

(1) 中间人攻击和 DNS 欺骗

主要针对 DNS 协议本身进行攻击, 比如中间人攻击, DNS 欺骗, 都是在用户和 DNS 服务器间对 DNS 查询进行抢先应答, 以达到攻击目标, 如图 4 中所示, 中间人攻击还可以达到信息窃取的目的. 这类攻击一种是在用户端, 一种是对 DNS 流量进行劫持. 用户端的有些是良性的 DNS 重绑定, 比如通知用户用新的 DNS 重绑定, 让用户主动选择, 另外一种是由恶意软件强制用户端将 DNS 绑定到恶意的 DNS 服务器上. DNS 流量劫持通常由 ISP 强制用户使用指定的 DNS 服务器, 或者由中间设备在解析路径上进行 DNS 拦截.

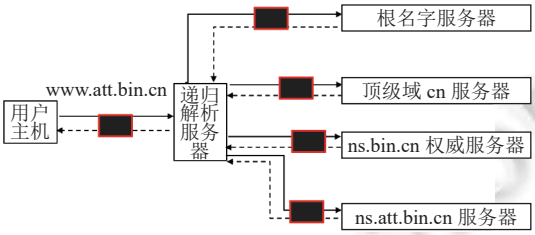

图 4 DNS 中间人攻击

线路上 DNS 拦截如图 4 方框所示位置, 分两种情况, 一种是在客户端和递归解析服务器间拦截, 相应的拦截设备直接返回 DNS 应答; 另外一种是在递归解析服务器与上层服务器间, 拦截设备在上层服务器前向递归解析服务器返回应答. 但 DNS 拦截并不都是恶意的, 比如中国国家防火墙. 递归解析服务器有下面 4 种解析情况: ① 正常解析: 遵循正常的 DNS 解析过程, 中间没有被拦截; ② 请求重定向: 用户发往的 DNS 服务器的 DNS 请求被拦截, 同时由替代的解析器完成解析过程; ③ 请求复制: 用户发往的 DNS 服务器的 DNS 请求不变, 拦截设备复制一份请求, 发往替代的解析器, 最先到达用户的应答被用户接受; ④ 直接应答: 类似请求重定向, 不同的是替代的解析器不再向上联系权威服务器, 而是直接返回应答. 上面除了第 1 种情况外, 其他 3 种都是 DNS 拦截或劫持, DNS 拦截可能的设备有防火墙、Censor、恶意软件、防病毒软件、转发器等.

DNS 拦截在现实网络中非常普遍, 很多时候是对用户实施保护, 比如中国国家防火墙, 通过 DNS 拦截和对

DNS 响应的抢答来阻止一些敏感的域名, 文献 [18] 发现在我国有 39 个 ASN, 劫持 DNS 并进行部分域名的重定向. 文献 [19] 对全球的 DNS 部分数据分析发现, 在全球 3 047 个自治域中, 259 个 (8.5%) 自治域内发现了 DNS 解析路径劫持现象, 其中包括一些大的运营商, 如中国移动等, 此外, 他们发现部分进行拦截的 DNS 服务器用一些过时的含有漏洞的软件, 无法处理一些 DNS 的安全功能, 比如无法处理 DNSSEC 的请求等.

(2) DNS 重绑定 (rebinding) 攻击

通过重绑定攻击[20,21], 攻击者可以利用 DNS 协议渗透到防火墙内部的服务器进行窃取信息等攻击, 如图 5[20,21]所示, 攻击者本来和目标服务器通过防火墙隔离, 攻击者难以直接通过防火墙对目标服务器进行控制, 但可以通过 DNS 重绑定完成对目标服务器的控制, 为了实施 DNS 重绑定攻击, 攻击者只需要注册一个域名, 比如注册 attacker.com, 通过发布广告等方式吸引用户流量, 用户一旦点击, 就会发送 DNS 请求, 攻击者在构造 DNS 响应中, 将它注册域名的服务器 IP 连同一个很小的 TTL 值返回, 攻击者服务器收到对用户浏览器的 HTTP 请求报文后, 返回一个恶意的 HTML 文档 (含有可以执行攻击者恶意意图的代码), 这个恶意的 HTML 文档无法直接完成对目标服务器的攻击, 由于浏览器本身具有同源的安全策略.

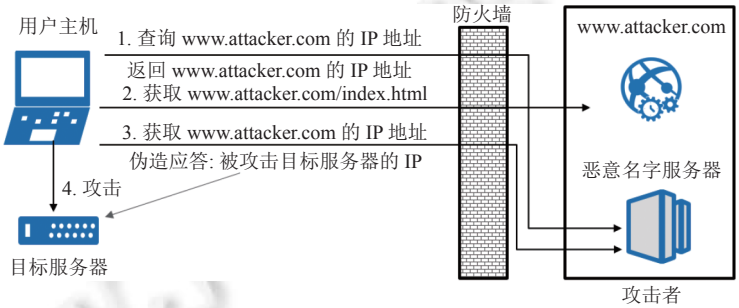

图 5 DNS 重绑定攻击[20,21]

但是, 这个恶意的 HTML 文档首先再次向 attacker.com 发送一个 HTTP 请求, 这时用户的 DNS 缓存关于 attacker.com 的记录已经过期, 这次, 攻击者服务器向用户返回的 DNS 响应报文为目标服务器的 IP 地址, 这时客户端浏览器就会允许攻击者的脚本读取目标服务器的 HTTP 响应, 因为两次链接实现了共享一个主机 (目标服务器), 符合客户端浏览器的同源策略, 这样攻击者就可以实现读取目标服务器的秘密文档, 窃取目标服务器的信息.

4.2 DNS 协议攻击检测技术

对中间人攻击和 DNS 欺骗的检测方法主要通过一种主动测量技术[22,23], 仍以图 4 为例, 首先由用户向递归解析服务器发送用户控制的域名服务器的域名, 为了判断是否出现拦截, 采用公开的递归解析服务器比如谷歌的 8.8.8.8 等为递归解析服务器 A, 这样相应的域名解析一定会到达控制的权威服务器 ns.att.bin.com, 而且是由递归解析服务器 B 发给它的, 这时, 如果 A=B, 说明中间没有出现拦截, 否则, 一定出现了拦截. 再具体分析拦截的情况, A 和 B 不同时, 当用户收到了 DNS 应答, 如果用户控制的权威域名服务器: ① 没有收到请求, 为直接应答; ② 只收到 B 的请求而没有收到 A 的请求, 则为请求重定向; ③ 收到 A 和 B 的请求, 则为请求复制.

实际上, DNS 拦截设备, 特别是善意的拦截设备会对一些域名进行直接应答, 而对另一部分域名进行重定向. 现实网络中存在大量的比如 ISP 或其他一些组织出于安全考虑对 DNS 流量的拦截, 对那些恶意的用户终端的 DNS 配置改变, 可以用一些蜜网节点运行使用户域名配置改变的恶意软件, 如 QHOST, DNSchanger, zecodec 等, 来发现和定位那些恶意的 DNS 服务器.

对于 DNS 重绑定攻击, 文献 [24] 提出一种增强浏览器插件来阻断恶意 Web 站点来获取任意 IP 地址的 socket 级别的访问, 他们对 Flash Player 和 Java 的 socket 访问策略进行了修改, 并得到相应公司的采用, 通过补丁增强了浏览器插件来阻止攻击. 同时他们开发了一个开源的递归解析服务器 dnswall[25]来阻断外面主机向内部用户进行解析. 另外, 服务器也可以通过验证 HTTP 头部并阻止那些未知的主机头部值进行自我防护.

4.3 DNS 服务器攻击

(1) DDoS/DoS

针对 DNS 服务器发动的 DDoS/DoS 攻击, 攻击者针对的目标是服务可用性发动攻击, 通过操纵傀儡机, 利用目标网络系统的服务功能缺陷进行攻击, 或者发送大量的无用报文, 消耗其系统资源, 使得该目标系统无法提供正常的服务. 根据文献 [26], 70% 针对 DNS 的攻击事件来自 DDoS/DoS, 并且在 2016 年爆发了迄今为止全球最大的 DDoS 攻击事件[27], 即表 3 中针对 DNS 运营商 Dyn 的 DDoS 攻击事件, 造成欧洲和北美大量的互联网服务不可用, 针对 DNS 服务的 DDoS 攻击会导致 DNS 服务不可用. 针对 DNS 服务器常见的 DDoS 攻击有以下几种.

● 随机子域名攻击, 攻击者向目标域名发送大量不存在的随机子域名解析请求, 使得相应的权威域名服务器不停地进行 NXDOMAIN 响应, 逐渐达到资源饱和, 无法处理其他正常请求.

● NXDOMAIN 攻击, 攻击者发送大量的不存在的域名, 使得本地递归解析服务器花费大量的时间和资源处理这些不存在域名的解析过程, 而难以响应正常的 DNS 请求.

● Phantom 域名攻击, 和 NXDOMAIN 攻击不同的是, 攻击者发送自己控制的域名服务器的解析域名的大量子域名, 而域名服务器从来不对递归解析服务器传来的解析请求进行应答或者延迟应答来消耗递归解析服务器资源.

● DNS 洪泛攻击, 攻击者对 DNS 服务器或解析器发送大量的 NXDOMAIN 域名请求, 导致服务器缓存逐渐被不存在的域名记录占满, 从而对正常的 DNS 请求记录响应缓慢.

● DRDoS, 操控肉鸡使用大量的解析器向 DNS 服务器进行反射增强攻击, 导致 DNS 服务器服务阻塞.

(2) 服务器劫持

DNS 劫持[28]是指用户在进行网页浏览时, 被强制访问某些网页, 或者在 App 使用中出现弹窗等现象. DNS 劫持主要分为数据劫持和域名劫持. 数据劫持主要对劫持的内容进行篡改, 比如插入广告或进行网络诱骗等, 域名劫持是将用户访问的目标域名劫持到攻击者设定的域名上, 使用户不能完成正常的访问, 更为严重的可以对用户造成网络钓鱼, 窃取用户私密信息或造成财产损失. 同时 DNS 劫持难以监管, 容易导致色情, 暴力等各种不健康信息传播.

(3) 缓存投毒

DNS 缓存投毒[29]是利用虚假 Internet 地址替换掉域名系统表中的地址, 进而制造破坏. 攻击者先是将假的地址植入到 DNS, 然后让服务器对假地址进行缓存记录, 最终在用户键入网址时把流量牵引到攻击者服务器, 完成缓存投毒的整个攻击流程. 攻击者可以利用 DNS 缓存投毒实现劫持攻击, 从而进一步进行网络诱骗, 植入恶意病毒, 窃取用户信息, 造成用户一系列不可估量的损失.

4.4 DNS 服务器攻击检测

(1) DDoS/DoS 检测

文献 [30] 提出了一种针对递归解析服务器进行 DNS 洪泛攻击的方法, 他们基于对递归解析服务器的 DNS 查询, 从 IP 到域名建立一个特征组, 根据不同时间窗口特征组的变化情况检测 DDoS 攻击, 文献 [31] 提出一种分层图结构从主机、子网、自治域 3 层通过机器学习的方法检测 DDoS, 他们的方法可以检测到低速的 DDoS 攻击. 文献 [32,33] 提出用名字过滤的方法将可疑的域名请求直接丢弃来检测并降低 DDoS 攻击的危害. 文献 [34,35] 提出当检测到大量的可疑的 DNS 请求时, 对部分可疑请求采取直接丢弃, 即用降低 DNS 响应速率的方式应对 DDoS 攻击. 文献 [36] 提出一种基于 FQDN 的白名单过滤机制检测 DDoS, 当检测到大量的域名请求不在白名单中时, 判断出现 DDoS 攻击, 通过直接将不在白名单中的域名丢弃的方式减低对 DNS 服务器的 DDoS 攻击的损害.

(2) 服务器劫持检测

文献 [37] 提出一种 REMeDy 方法检测对服务器的劫持, 具体流程如图 6[37]所示, 首先抽取 DNS 流量的特征, 在一个时间段内对每个 DNS 应答, 提取递归解析服务器的 IP 地址 R 和请求的域名 D, 及它们对应的 4 个特征: N_A, 应答的 A 和 AAAA 的地址数; N_{CNAME}, 应答中 CNAME 的数量; TTL, A 或 AAAA 记录的 TTL 值; ASN, IP 地址列表对应的自治系统号列表, 可以从 WHois 中得到; 这样得到{R, D, N_A, N_{CNAME}, TTL, ASN}特征序列, 然后按

域名 D 进行分组归并, 对于 D 的任意特征 f, 计算其频率分布: Freq(D, f). 对于 N_A, N_{CNAME}, ASN 这 3 个特征, 分别计算它们的频率分布后, 使用 BOX-PLOT 规则, 利用分布的低分位 Q1 和高分位 Q3 计算 IQR=Q3–Q1, 对任意 Freq(D, f), 把高于 Q3+1.5×IQR 的部分, 认为是异常点, 通过这样初步定为异常的 (R, D), 对于 TTL, 定义 (R, D) 为对应域名和递归解析服务器最大的 TTL 值, Freq(D, f=TTLMAX) 为所有递归解析服务器的 TTLMAX 频率分布, 仍用 BOX PLOT 规则来判断异常的 (R, D).

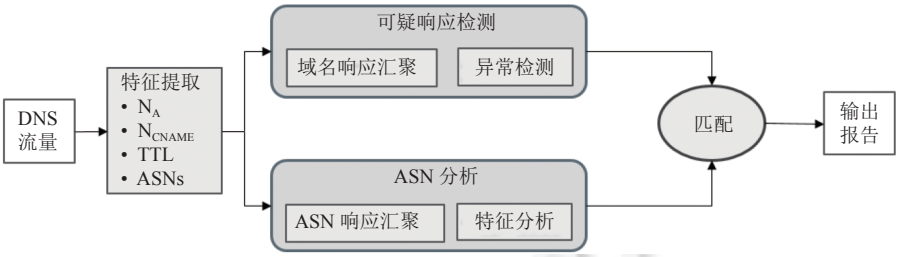

图 6 DNS 劫持检测流程[37]

在得到可疑的列表 (R, D) 后, 为了进一步降低错误率, 在每个 (R, ASN) 级别, 进一步计算每个特征的频率分布 Freq(R, ASN, f), 根据异常通常使得同一 AS 内相关特征变动较小的特性, 进一步从 (R, D) 可疑列表中把使 Freq(R, ASN, f) 变化较大的 (R, D) 过滤掉, 具体是过滤掉{D, N_A, N_{CNAME}, TTL}这 4 个特征中两个或以上变化比较大的, Freq(R, ASN, f) 变化较大定量为超过前面时间段正常 (R, D) 相应特征的平均值. 这样, 最后剩下的 (R, D) 序列即为异常的 DNS 域名和相应的递归解析服务器.

(3) 缓存投毒检测

文献 [38] 研究了一种新型的缓存投毒攻击, 通过将服务器缓存插入域名虚假的 CNAME 或 DNAME 记录, 这两种记录不会直接对相应域名的解析过程产生影响, 只有当这个域名在缓存中的真实记录过期时, 才会用插入的虚假记录进行解析, 从而进行缓存投毒攻击. 文献 [39] 通过测量 DNS 请求到 DNS 响应的时间分布情况, 发现呈一组泊松分布, 在相邻的两个泊松分布间有一个没有任何值的时间的空白, 而缓存投毒攻击可能使这个空白的时间段内产生, 基于这样的观察, 通过机器学习的方法建立模型进行检测.

4.5 域名滥用

域名滥用攻击通常指攻击者注册一些用户容易拼写错误或容易混淆的域名进行投放广告, 商标滥用, 钓鱼等攻击, 称为域名 squatting, 如表 5[40]所示, 通常有 5 种域名 squatting 形式: 1) typo-squatting[41], 域名的拼写错误; 2) bit-squatting[42], 域名存储或传输过程中导致的位反转; 3) homograph-squatting[43], 由于特定编码或者其他因素导致域名视觉上难以区分; 4) sound-squatting[44], 音频输入设备等方法导致的域名拼读上难以区分; 5) combo-squatting[45], 在原始域名中增加一个新的单词. 从表 5 中可以看到不同 squatting 的表现形式, 而通过这些域名滥用, 攻击者可以使用户重定向到他们注册的网站中对用户进行发布广告[46,47]、商标或机构的滥用[48]、钓鱼[49,50]等攻击.

表 5 域名滥用[40]

域名	滥用类型	利用滥用实施攻击类型
wwwyoutube.com	Typo (丢失点号)	钓鱼
outube.com	Typo (丢失字母)	投放广告
yuotube.com	Typo (字母顺序颠倒)	机构误用
uoutube.com	Typo (字母替换)	投放广告
yioutube.com	Typo (字母插入)	投放广告
qoutube.com	Bit	商标误用
yovtvbe.com	Homograph	投放广告
utube.com	Sound	商标误用
worldyoutuber.com	Combo	投放广告

4.6 域名滥用检测

这种由拼写或容易混淆的域名引起的域名 squatting, 一般的检测方法难以检测出来, 通常需要分析域名自身, 文献 [51] 对于特定域名的可能的拼写错误形式设计了 5 个模型, 用于检测 typo-squatting 的滥用. 文献 [42] 对一个特定域名的每个字母进行位反转的各种组合进行比对来检测 bit-squatting 的滥用. 文献 [52] 将现有的容易混淆的字母表进行扩展提出一个新的表称为 SimChar 对 homograph-squatting 进行检测. 文献 [44] 将特定域名内的单词进行分割并用一个同音异形的词库中的对应单词进行替代来检测 sound-squatting. 文献 [53] 通过一些预定义的商标列表联合域名的 whois 和 AS 信息对 combo-squatting 进行检测.

4.7 DNS 增强攻击

(1) DRDoS

如图 7 所示, DNS 反射放大攻击[54]主要是利用 DNS 回复包比请求包大的特点, 放大流量, 伪造请求包的源 IP 地址为受害者 IP, 将应答包的流量引入受害的服务器. 攻击者通过不断向 DNS 服务器发送伪造了源 IP 地址的解析请求, 形成 DRDoS 攻击, 发送的 DNS 查询请求数据包大小一般为 60 B 左右, 而基于 DNSSEC 对 ANY 记录查询返回结果的数据包大小通常为 3 000 B 以上, 放大因子为应答包的大小除以请求包的大小, DRDoS 通常可以造成相比于请求流量放大几十倍的攻击, 使目标主机大量消耗计算资源, 严重可以致瘫.

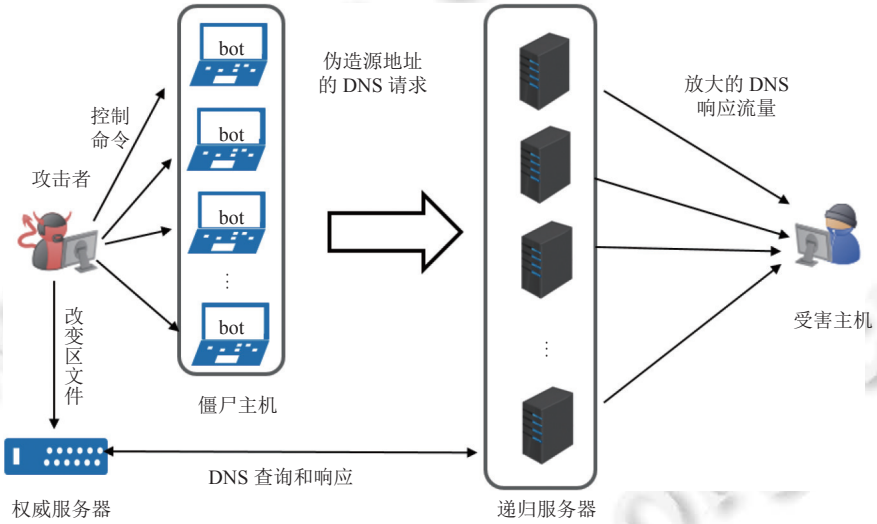

图 7 DRDoS 攻击过程

(2) Worm

蠕虫是一段可以自我复制的恶意代码, 它通过探测目标主机的漏洞在网络上传播, 在 IPv4 空间, 比较容易随机生成另外一个 IP 地址并用这个地址作为下一个传播的目标, 然而在 IPv6 这种方法不再实用, 因为 IPv6 空间太大, 为了在 IPv6 空间中进行有效传播, 蠕虫可以用类似 DGA 的方法产生主机名, 如果能通过 DNS 解析成相应的 IP 地址, 就把这个地址作为下个目标进行传播[55], 因此, 蠕虫在 IPv6 空间的传播过程中 DNS 取代了地址空间的扫描功能, 如果蠕虫在传播过程中具有准确的域名生成器, 传播速度会大大增强.

(3) Spam

发送策略框架 (sender policy framework, SPF) 和域名关键字认证邮件 (domain keys identified mail, DKIM) 是两种常用的保障邮件安全的 DNS 机制, SPF 主要通过 TXT 记录标记邮件服务器的允许列表, DKIM 允许接受端通过 TXT 记录里面的公钥来验证签名, 这两种措施在很大程度上解决了邮件的伪造问题, 但邮件炸弹可以自身利用这两种防御措施[56], 使这些邮件的来源域名显得合法, 这样就增加了恶意邮件的传播速率.

4.8 DNS 增强攻击检测

(1) DRDoS 攻击检测

文献 [57] 通过 sFlow 分析 DNS 请求 ID 可以实时检测和缓解 DRDoS. 文献 [58] 利用递归解析服务器缓存的过期 DNS 记录缓解 DRDoS 攻击. 文献 [59] 通过强制参与攻击的递归解析服务器的 DNS 查询速率共享来检测和阻断 DRDoS 攻击. 文献 [60] 设计了一种新型蜜罐 AmpPot 来捕获放大 DDoS 攻击, 他们发现良性域名和恶意域名都可以参与攻击. 文献 [61] 提出一种有效“Challenge-response”机制来验证 DNS 请求者的身份, 以检测和缓解 DRDoS 攻击. 文献 [62] 通过对 DNS 和简单服务发现协议 SSDP 联合测量检测 3 个国家的网络的 DRDoS 攻击, 并对增强因子进行估算. 文献 [63] 提出一种新的不需要利用 Botnet 的 DRDoS 攻击, 攻击因子可以达到 44, 相对于传统的 DRDoS 攻击, 这种攻击减少了可以被用于检测的整体流量, 因此, 更加难以检测.

(2) Worm 攻击检测

蠕虫在传播过程中请求生成的域名大概率是未经注册的域名, 会产生相应的 NXDOMAIN 应答报文, 文献 [64] 通过分析蠕虫经由 DNS 查询的传播经常会导致 NXDOMAIN 应答, 把 DNS 响应报文为 NXDOMAIN 的作为分析和探测蠕虫和僵尸网络的有效手段. 此外, 蠕虫在 IPv4 空间的传播过程中通过直接扫描 IP 地址进行传播, 而不通过 DNS 解析, 文献 [65] 提出一种分析蠕虫传播宿主主机向外发送流量中 DNS 查询流量情况的方法检测蠕虫, 正常情况下, 一次 DNS 请求后面紧跟一个 IP 访问, 如果出现大量的没有 DNS 查询的 IP 访问, 说明出现了蠕虫传播.

(3) Spam 攻击检测

邮件炸弹如果依赖于 DNS 进行传播, 同样也会在传播过程中产生相应的模式, 利用 SPF 进行传播的邮件炸弹会强制注册一个域名并且为每个邮件发送主机生成一个记录, 这样邮件炸弹会为成百上千的传播主机配置相应记录, 而这样的配置是异常的, 文献 [66] 通过主动测量的方式和机器学习的方法来识别这样异常域名配置, 从而检测邮件炸弹.

4.9 恶意域名/IP

(1) Botnet

僵尸网络 Botnet 是指采用一种或多种传播手段, 将大量主机感染 bot 程序病毒, 从而在控制者和被感染主机之间所形成的一个可一对多控制的网络, 控制者通过控制 (command and control, C&C) 服务器控制僵尸主机, 发布命令完成攻击, 攻击者可以通过 Botnet 进行垃圾邮件, 信息窃取, 网络诈骗, DDoS 等攻击. Botnet 根据 C&C 服务器对僵尸主机的控制方式可以分成不同类型, 本文主要指利用 DNS 服务进行 Botnet 攻击, 因此, 限于 C&C 服务器通过 DNS 消息和僵尸主机进行通信[67,68], 而不是广义的 Botnet. 受感染的主机通过 DNS 消息和 C&C 服务器连接, 然后通过 DNS 的 TXT 记录进行通信, 发布恶意攻击命令等.

(2) Fast-Flux

域名通常使用快速变换 (Fast-Flux) 它的 IP 地址的技术来逃脱检测[69,70], 比如广义的 Botnet 的 C&C 服务器的域名所有所有 DNS 记录的 TTL 值设置很小, 通常小于 5 min, 当 C&C 服务器的 IP 地址被检测到后所有的僵尸主机可以在很短的时间内学习到 C&C 服务器的变换的新的 IP 地址, 这种技术也通常用于 CDN 中进行流通均衡, 不同之处在于 CDN 服务通常具有自己的地址空间, 而恶意的 Fast-Flux 经常指向被控制的主机, 而不是同类地址空间或者同一个公司的地址空间.

(3) DGA (domain generation algorithm)

攻击者通常用 DGA 算法生成大量的域名实施恶意攻击[71], 比如广义的 Botnet 通常使用 DGA 算法使得他们的攻击活动难以检测, 攻击者往往通过在不同的 TLD 注册一些域名得到一个全球的恶意域名分布, 增加检测难度, 比如, Conficker 可以生成 5 万个随机的域名遍布 13 个 TLD[71,72], 使得检测这样的恶意域名非常困难.

4.10 恶意域名/IP 攻击检测

对这类攻击的检测, 无论是 Botnet、Fast-Flux、还是 DGA, 目标都是检测出恶意的域名和 IP, 从第 4.9 节的分

析也可以看到 Fast-Flux 和 DGA 同样也是 Botnet 逃脱检测的两种技术, 因此, 一些基于 DNS 检测 Botnet 的综述文献 [73,74] 将 Fast-Flux 和 DGA 也作为检测 Botnet 的两类方法, 但实际上, Fast-Flux 和 DGA 也不限仅用于 Botnet 上, 而且, 学术界的很多标志性的工作定位于检测恶意的域名和 IP, 并不局限于三者中的一种, 关于这类攻击的检测是目前学术界研究的热点, 同样也是本文综述的重点, 本文对学术界比较认可从 2009 年到现在的研究工作做一个简要介绍和对比分析.

(1) FFM (Fast-Flux monitor)[75]

基于实时分析 DNS 流量中与 Fast-Flux 行为相关的一些特征, 比如 TTL, IP 变化率、AS 等, 通过一些网站中对已知恶意域名的标注, 用贝叶斯分类器训练这些特征生成域名的分类模型, 检测 DNS 流量中新的恶意域名, 并每周把新检测的域名包含进来进行训练形成下一周的检测模型.

(2) Notos[76]

Notos 是 DNS 的第 1 个动态名声系统, 它用一些特征计算一个给定域名的评分值, 恶意域名相对于正常域名的评分值较小, 用于计算评分值的特征有: 1) 基于网络的特征, 比如和域名相关联 IP 的总数、这些 IP 的地理分布、和不同的 AS 号等; 2) 基于区文件的特征, 指一个 IP 地址相应的域名的特征, 比如域名的字符串特征、域名的平均长度、不同字符的频率等; 3) 基于证据的特征, 比如域名和已知恶意域名的通信数量、IP 地址和已知恶意域名的通信数量等计算. 用离线的方式基于这些特征训练评分系统, 然后用在线的方式对 DNS 流量中的域名进行评分.

(3) Kopis[77]

Kopis 通过监测 TLD 和权威服务器的 DNS 流量特征来检测恶意域名, 监测的主要特征有: 1) requester diversity (RD), 用于表征请求特定域名的 IP 地址分布特征, 其值依赖于请求的 IP 地址、BGP 前缀、ASN、国家代码; 2) requester profile (RP), 用于表征特定时间段请求域名数量的大小; 3) resolved-IPs reputation (IPR), 用于表征特征域名解析的 IP 地址空间曾用于恶意活动的情况. Kopis 基于一些已知的正常和恶意的域名的特征用 random forest (RF) 训练模型, 然后对 DNS 流量中新的域名进行分类.

(4) Exposure[78,79]

Exposure 通过提取 DNS 流量中已知正常域名和恶意域名的特征用决策树 (decision tree, DT) 进行训练模型, 然后在线的方式使用模型对恶意域名进行实时检测. 提取的主要特征有: 1) 基于时间的特征, 主要有特定域名被请求的时间段和频率; 2) 基于 DNS 响应的特征, 由解析的不同 IP 和域名、IP 地址所在的国家、IP 地址和正确域名的匹配情况计算; 3) 基于 TTL 的特征, 由特定时间段 TTL 的统计特征计算; 4) 基于域名的特征, 由域名中数字的频度、域名中有含义的最长子串的值计算.

(5) FluxBuster[80,81]

FluxBuster 主要为检测 Fast-Flux 设计, 首先它通过将 DNS 流量中确认为正常的域名 (TTL 值比较大, 解析的 IP 地址数量较少, IP 分布集中) 过滤掉, 剩下的域名用层次聚簇算法进行聚簇, 每个簇作为一个可疑的 Flux 网络, 为一些关联域名和他们解析的 IP 地址的集合, 每个簇再计算下列特征: 1) 解析的不同 IP 的数量; 2) 簇中不同域名的数量; 3) 簇中域名的 TTL 均值; 4) 前面时间段中解析到目前簇中的 IP 的不同域名的数量; 5) IP 地址的分散值; 6) 当前时间段中簇中每个 DNS 查询关联域名平均新发现的 IP 数量均值; 7) 当前时间段相比于上一时间段簇中每个 DNS 查询关联域名平均新发现的 IP 数量均值. 最后, 用 DT 自动划分哪个簇是 Flux 网络.

(6) ExecScent[82]

ExecScent 是一个通过分析 C&C 服务器经已知恶意软件进行 HTTP 通信的情况检测恶意的 C&C 域名, 它先基于已知的恶意软件 C&C 请求将 HTTP 请求通用化, 然后基于共享的目的 IP 地址对这些请求聚簇, 对于每个簇内的 C&C 请求, 用层次聚簇算法根据它们的相似性 (URL, 用户代理字符串, HTTP 头部域和值的数量) 进一步分组, 最终得到的簇中如果含有至少一个对已知 C&C 域名的请求, 则该簇生成控制协议模板, 对于这些带标签的正常和恶意域名及生成的控制协议模板, 用支持向量机进行训练分类器, 最后, 新的 HTTP 请求依据训练的分类器进行检测.

(7) multi-dimensional aggregation monitoring (MAM)[83,84]

MAM 通过测量域名和 IP 随时间的稳定性来判断异常, 它依据正常的域名不同时间解析的 IP 相对类似, 形成一个比较稳定域名和解析 IP 的对应模式, 而异常不具备这样的稳定性, 它将 DNS 流量构造一个树状结构, 节点代表了域名到 IP 的解析, 依据计算节点的稳定性来检测异常, 通过判断当前节点和上一个时间段中其他节点的相似性计算当前节点的稳定性, 如果两个节点的域名的 IP 属于一个子网并且域名的子域也属于同一个子网, 则这两个节点是相似的. 基于这样稳定性的变化情况检测节点的异常行为.

(8) Segugio[85,86]

Segugio 通过生成一个主机到域名的二分树 (代表新的域名到已知恶意域名的关系) 来检测恶意域名, 主要依据下面 3 个假设检测异常: 1) 如果一个主机感染, 它容易查询新的恶意域名; 2) 相同恶意软件家族感染的主机容易查询相同恶意软件家族控制的域名; 3) 正常主机不太可能去查询恶意域名. 因此主要依据 3 个特征: 1) 主机行为, 对一个域名 d 的查询的主机的总数中如果已知恶意的查询数越大, d 为恶意域名可能性越大; 2) 域名活动, 域名 d 在给定时间段内被查询的天数; 3) IP 滥用, 域名 d 解析的 IP 地址中在前面时间和已知恶意域名的关联情况. Segugio 首先创建二分树, 基于现有知识将树中节点划分成恶意域名, 正常域名, 未知域名, 同时裁剪掉树中不活跃的节点, 最后, 用以上特征训练得到的分类器对那些未知的域名进行正常或恶意域名的划分.

(9) DNS 失败图[87]

DNS 失败图只建立主机和 DNS 响应记录不为“NOERROR”的域名关系图, 通过图分解算法将 DNS 失败图分为 3 类: 1) Host-star, 少量的主机进行的大量 DNS 查询; 2) DNS-star, 大量的主机查询少量的域名; 3) Bi-mesh, 一部分主机和一部分域名存在紧密的查询关系. 应用已知的域名黑名单, 他们发现 Host-star 主要由特洛伊恶意软件访问域名引起, DNS-star 主要由垃圾邮件引起, Bi-mesh 主要是僵尸网络的活动.

(10) DNSMap[88]

DNSMap 实时读取 DNS 流量, 并将域名和 IP 的映射关系进行高效存储, 持续监控映射关系的变化情况, 进行图分析模块处理, 图分析模块采用二分图的方法将用户和域名的交互关系进行投射, 构建的二分图只代表在检测的当前时间段发生变化的用户和域名的映射关系, 而不是整个存在的映射关系, 即当前时间段相比前面时间段用户和域名的映射关系发生了变化, 才构建相应的节点和边, 没有变化的不构建, 这样的二分图有很多不同的连接单元, 把那些只包含一个 FQDN 和 IP 的单元去掉, 最后, 建立异常敏感特征向量, 根据不同图单元的特征向量值, 可以有效判断异常导致的图特征向量的变化, 从而定位恶意的域名和用户.

(11) BotGAD[89,90]

BotGAD 将 DNS 流量用域名黑白名单进行过滤, 生成一个域名和请求它的所有 IP 地址的映射, 通过这种映射关系计算域名间的相似性, 并进行组行为特征分析, 利用下面一些 DNS 特征进行聚簇: 1) 域名标签的数量; 2) 域名标签的平均长度; 3) 请求域名的次数; 4) 请求域名的不同 IP 地址数量; 5) 域名解析成不同 IP 地址的数量. 将和已知恶意域名同簇的域名检测为新的恶意域名.

(12) BotDAD[91]

BotDAD 首先由 host profile 模块处理 DNS 流量, 为网络中的所有主机建立它们的 profle, 然后由指纹生成模块预定义一个异常比较敏感的特征向量: 无应答请求记录 (发出的请求没收到相应应答, 可以作为判断 bot 的重要依据), 无请求应答记录 (只收到应答而没有对应的请求记录, 可以作为被攻击主机的重要依据), 特殊应答类型 (NXDOMAIN 记录, EDNS 记录, DNSKEY 记录, ANY 记录, DSEC, TXT 记录). 根据每个 host 的 profile 计算每个 host 的特征向量值, 异常检测模块根据实验情况预定义特征向量的阈值来检测每个 host 的 profile 的异常情况, 然后将结果作为训练数据用随机森林训练待检测模型, 由检测模型来检测异常情况.

(13) PsyBoG[92]

PsyBoG 不同于其他检测方法, 为了高效的应用 DNS 流量, 它只分析 DNS 请求的流量, 从请求流量中提取主机 IP, 请求的域名, 时间戳 3 个特征值生成每个主机的时间序列特征, 然后利用功率谱密度 (power spectral density,

PSD) 将时序信号转换成频率信号, 将高频部分的信号检测为 Botnet 流量.

(14) GMAD[93]

GMAD 用一个代表域名查询序列的域名旅游图 (domain name travel graph, DNTF) 检测恶意域名和受感染的主机, 它也只利用 DNS 查询流量, 首先提取 DNS 查询流量的 4 个特征 (查询的域名作为节点, 域名间的时序关系作为边, 节点的信息有请求域名的源 IP 列表和查询的总数) 建立 DNTF, 然后根据域名被相同的过程或服务查询的情况进行聚簇, 最后用域名黑名单检测新的恶意域名.

(15) 半人工标注[94]

这个方法的第 1 阶段先用 DNSMap[88]将易变的域名和 IP 的映射关系图单元构建出来; 第 2 阶段将那些至少 40 个域名或者至少 20 个 IP (这些 IP 至少跨越 2 个 AS) 图单元过滤出来, 作为下阶段分析的可疑单元; 第 3 阶段的自动分析阶段提取图单元的一些特征进行分析, 相应的特征分析有图特征分析、域名特征分析、IP 特征分析、域名白名单分析、域名黑名单分析、IP 黑名单分析; 第 4 阶段用 K-means 依据这些特征对图单元进行聚簇, 形成两个簇, 其中一个是代表异常的簇, 一个是正常的簇; 最后一个阶段就是人工验证阶段, 通过人工查看分析两个簇中图单元的各个特征, 进一步减少聚簇产生的错误, 得到最终的检测结果.

(16) Pleiades[95]

Pleiades 主要基于的思想是 DGA 算法生成的域名大部分会导致 NXDOMAIN 的响应报文, 同一僵尸网络的 Bot 用相同的 DGA 算法会引起相类似的 NXDOMAIN 报文. 在 DGA 发现阶段, 所有的返回那些 NXDOMAIN 的域名依据统计特征 (长度、频率、熵等) 的相似性进行聚簇, 主要为了发现那些相同 DGA 算法生成域名, 在 DGA 分类和检测阶段, 多类分类器用于标注各簇中那些已知的具体的 DGA 软件类型, 对于那些不能对应到已知 DGA 软件的簇, 用另外一个隐式的马尔科夫模型检测是否是一个新的 DGA 家族或现有 DGA 家族的变体.

(17) DBod[96]

DBod 主要基于相同 DGA 恶意软件感染的僵尸主机将查询相同集合的域名且大部分请求会失败, DBod 分析失败的 DNS 请求, 基于请求时间、请求次数分布等特征将网络中感染的主机和正常的主机聚簇, 基于黑名单机制, 用一个评分函数给每个簇打分, 来检测是恶意的还是正常的簇.

(18) LSTM[97]

文献 [97] 提出一种长短期记忆网络 (long short-term memory, LSTM) 结合注意力机制的方法对 DGA 产生的恶意域名进行分类, 通过对域名的提取、填充和词向量的转换, 转换成一个 54×128 的矩阵, 然后用 LSTM 和注意力机制相结合的深度学习模型进行分类.

(19) BDS[98]

BDS 提取 DNS 流量的特征信息, 对于 DNS 请求, 提取域名、源 IP、目的 IP、时间戳, 对于 DNS 响应, 提取解析的 IP、域名别名, 存入库中, 对于库中新出现的域名, 通过工具进一步查询一些辅助信息: Whios 查询信息 (DNS 记录年龄、DNS 邮件记录、权威服务器), 对 MX 服务器的查询信息, Web 服务是否存在的信息, 域名黑名单更新, 得到所有这些特征后, 由评估模块通过对这些特征值的联合计算来判断异常情况.

我们简要介绍了 19 种恶意域名和 IP 的检测方法, 我们通过表 6 对这些方法进行总结和分析, 总体来讲, 这些系统都是提取流量的一些关键特征, 然后通过一些方法对特征进行分析, 检测恶意的域名或进一步感染的 IP 地址, 其中 FFM[75]、FluxBuster[80,81]主要用于检测快速变换 IP 的域名, ExecScent[82]、Segugio[85,86]主要用于检测恶意软件相关的域名, Pleiades[95]、DBod[96]、LSTM[97]主要用于检测基于 DGA 的域名, 其他方法相对比较综合, 都基于检测域名和 IP 的异常情况, 检测 Botnet、DGA、Fast-Flux.

对于这些检测方法的优劣评价, 由于每种方法使用的流量、实验环境都不同, 所以, 无法对这些系统进行绝对的性能比较, 我们只能从这些系统的文献中大致说明它们的优势和不足之处. 表 7 从训练流量的时间 (用了多少天的流量进行训练), 重新训练周期、早期检测 (在异常出现在公共的黑名单之前检测出来)、检测新的异常、在线部署 (在线检测的能力)、检测效果这几个方面简要对比这些系统的检测能力, 其中 NA 代表无法从文献得知.

表 6 恶意域名/IP 检测方法对比一

系统	目标	数据	方法	特征	文献/时间
FFM	检测Fast-Flux域名	被动和主动探针测量的数据	贝叶斯网络	TTL, IP变动率, AS, 恶意域名和IP的关联	[75] 2009
Notos	检测恶意域名	两个ISP网络和SIE[98]共68天数据	DT	基于网络的和区文件的特征数据	[76] 2010
Kopis	检测恶意软件相关的域名	4个ISP和SIE[99]大约8个月从两个权威和.ca TLD采集的数据	RF	请求者IP的分布、请求者IP的数量	[77] 2011
Exposure	检测恶意域名	一个ISP和SIE[99]大约10周的DNS查询数据, 共约1亿个查询记录	DT	时间、DNS响应的IP和域名, TTL, 域名的字符串特征	[78] 2011, [79] 2014
FluxBuster	检测Fast-Flux域名	SIE[99]大约10月的DNS流量	DT	解析的IP数, 簇内域名数, TTL均值, IP分布及应用属性	[80] 2012, [81] 2009
ExecScent	检测恶意软件相关的C&C域名和感染的主机	两个学术和一个财政机构共4周的网络流量, 平均每天约3 500万HTTP查询	SVM	URL相似性属性 (中间路径、正则表达式、恶意域名家族), 域名欢迎度	[82] 2013
MAM	检测网络异常域名和IP	两个ISP约14个月的DNS数据, 近34万个IP和83万多个域名	对域名和IP映射的基于树的分析	稳定性测量的属性 (不同时间段域名到IP映射模式的变动情况)	[83] 2013, [84] 2012
Segugio	检测恶意软件相关的域名	两个ISP共8天随机采样的流量	RF	查询域名的状态, 域名被恶意软件查询的情况, 查询域名的恶意主机IP	[85] 2015, [86] 2016
DNS 失败图	检测网络异常域名和IP	大学网络3个月的流量, 每天约200万DNS请求	图分解算法	请求失败的域名和请求它的主机IP的关联关系	[87] 2010
DNSMap	检测网络异常的域名和IP	一个大型ISP大约3周的DNS流量, 约6.7 M不同的域名	图分析算法	发生变化的用户IP和请求的域名的映射关系	[88] 2016
BotGAD	检测网络异常的域名	大学校园网一天的DNS流量, 一个ISP一天的DNS流量和另外一天的DNS日志	域名的聚簇和相似性分析	域名标签的数量, 2域名标签的平均长度, 请求域名的次数, 请求域名的不同IP地址数量, 域名解析成不同IP地址的数量	[89] 2009, [90] 2012
BotDAD	检测网络异常的域名和IP	校园网2016年4–5月间随机10天的网络流量	RF	无应答请求记录, 无请求应答记录, 特殊应答类型 (NXDOMAIN记录, EDNS记录, DNSKEY记录, ANY记录, DSEC, TXT记录)等组成15个主机特征	[91] 2019
PsyBoG	检测网络异常的域名	两个校园网各一天的DNS流量, 两家大型公司的DNS流量, 两个不同.kr TLD的DNS流量	功率谱密度PSD	主机IP, 请求的域名, 时间戳	[92] 2016
GMAD	检测网络异常的域名和IP	美国两个ISP的2010年6月各两个小时的流量, 韩国两个ISP的2010年1月各两个小时流量	图分析及聚簇	查询的域名, 域名间的时序关系, 有请求域名的源IP列表和查询的总数	[93] 2014
半人工标注	检测网络异常的域名和IP	丹麦的一个边缘ISP的2013年9月中一周的DNS流量和2014年6到7月中4周的DNS流量	图分析、K-means、人工分析	图特征、域名特征、IP特征、域名白名单、域名黑名单、IP黑名单	[94] 2015
Pleiades	检测网络异常的域名和IP, 主要DGA-bots	北美一个ISP的15个月的DNS流量	K-means、DT、隐马尔科夫模型	NXDOMAIN、*n*-gram特征、熵特征、域名结构特征	[95] 2012
DBod	主要检测DGA-bots	一个教育网从2013年5月到2015年6月共大约26个月的流量	聚簇、评分函数	失败的DNS请求, 基于请求时间、请求次数分布	[96] 2017
LSTM	检测DGA域名并进行分类	Alexa最上边11万个域名和DGA域名[100]	LSTM、注意力机制	域名	[97] 2019
BDS	检测恶意域名	大学网络3天的流量, 大约45万个不同的域名, DNS请求在480多万, 响应在396多万	特征权重的联合计算	DNS请求 (域名、源IP、目的IP、时间戳), DNS响应 (解析的IP、域名别名), 辅助信息 (Whios、MX、Web、黑名单)	[98] 2011

表 7 恶意域名/IP 检测方法对比二

系统	训练数据 (天)	重新训练周期	早期检测	检测新的异常	在线部署	检测效果
FFM	NA	每周	×	×	NA	检测率比较依赖实验数据
Notos	15	×	√	×	√	真阳率96.8%, 检测的异常比较依赖历史数据
Kopis	30	×	√	√	√	已知恶意软件相关域名检测率98%, 未知恶意软件域名检测率74%, 难以检测短生命期的恶意软件相关的域名
Exposure	7	每天	√	√	√	可以检测各类Botnet恶意行为 (Flux、DGA、Malware), 检测率高98%, 误报率低
FluxBuster	NA	×	√	×	√	检测率高, 但需要32 h才能完成检测, 需要人工打标签影响系统扩展性
ExecScent	14	每天	×	×	√	检测率高, 误报率低, 不能检测到在训练阶段没有的加入的恶意软件家族
MAM	NA	NA	NA	×	NA	73%的真阳率和0.3%的假阳率, 难以检测量级较小和影响较小的异常
Segugio	1	×	√	√	√	94%的真阳率和0.1%的假阳率, 难以检测恶意软件控制的正常域名
DNS 失败图	1	每天	√	√	×	检测率比较高, 可以发现多种类型异常, 计算代价小, 受限于DNS响应失败的情况
DNSMap	>2	>2	√	√	√	训练时间短, 检测率高, 假阳率低, 无法检测映射数量较小的异常
BotGAD	×	×	√	√	√	效率高, 几分钟可以分析一个小时左右的流量, 检测率高95%以上, 假阳率低0.4%以下, 扩展性好, 无法检测不使用DNS进行C&C通信的僵尸网络
BotDAD	10	10	√	√	√	检测率较高99%以上, 训练和检测时间都比较短, 但目前方法主要检测1个小时的流量数据, 在这样短的时间段没有恶意活动的bot无法检测出来
PsyBoG	×	×	√	√	√	95%的检测率和0.1%的假阳率, 扩展性好, 易扩展应用到大型网络, 但不容易检测到随机和和慢速的异常
GMAD	×	×	√	√	√	检测率超过90%, 平均99%, 假阳率低于0.5%, 计算复杂度低O(n), n为DNS查询数, 易扩展, 易检测多域名的异常活动, 但对单个域名的异常活动检测能力有一定限制
半人工标注	>2	>2	√	√	×	检测率很高可达99%以上, 同时假阳率很低0.002以下, 但由于需要一部分人工判断, 扩展性差
Pleiades	1	每天	√	√	√	平均检测率达到99.7%, 平均假阳率为0.1%, 但不能识别使用相同DGA算法的不同僵尸网络, 检测结果依赖NXDOMAIN响应
DBod	NA	NA	√	√	√	检测率高的同时保持较低的误检率, 但无法检测期间停止活动的僵尸网络
LSTM	1	每天	×	×	×	检测分类精度在85%左右, 相对比较简单, 不需要提取具体的域名特征
BDS	3	NA	√	√	×	检测率达到68%的同时假阳率在3.8%

注: NA表示文献中无法得到

研究人员可以从这些方法的检测目标和检测能力中参考目前已有的方法开展研究, 也可以通过分析这些方法的研究手段和思路, 去改进现有的方法或开辟新的思路, 目前一部分方法是通过提取一些对于异常变化比较敏感的 DNS 流量的特征, 通过机器学习的方法进行检测, 另外一部分方法通过将域名和主机的映射关系建立图, 通过分析异常发生时图的一些变化情况进行检测, 还有一些方法基于统计特征直接进行分析, 还有通过深度学习的方法进行异常的分类. 而这些方法文献的检测率和误检率等检测指标相对依赖于论文中的流量数据, 且绝大部分工作没有提供源代码, 很难将这些方法在一个统一的环境中进行实验比较.

4.11 隐秘隧道

DNS 隧道是将其他协议的内容封装在 DNS 协议中, 然后以 DNS 请求和响应报文完成传输数据 (通信) 的技术. DNS 隧道是一种隐秘隧道, 由于防火墙和入侵检测设备难以过滤作为基础服务的 DNS 流量, 攻击者可以利用 DNS 实现和受防火墙等保护的内部主机的隐秘通信信道, 实现诸如远程控制, 文件传输等操作. 文献 [101] 提供了对 DNS 隧道及检测相对全面的调研, 我们这里相对简要介绍每类检测方法.

建立 DNS 隧道需要 3 个部分: 1) 攻击者控制的域名或子域名; 2) DNS 隧道工具; 3) 由 C&C 服务器伪装的

DNS 权威服务器. 依据受控主机和伪装的 DNS 服务器的通信方式, 在建立 DNS 隧道前, 受控主机和 C&C 服务器都需要安装并激活 DNS 隧道工具, 来进行 DNS 查询类型和数据编码方式的协商, DNS 隧道工具以 C/S 方式运行, 客户端安装在受控主机, 服务端安装在 C&C 服务器, DNS 隧道工具可分为 IP over DNS 和 TCP over DNS 隧道工具两种[102], IP over DNS 将 IP 报文封装到 DNS 隧道中, TCP over DNS 将 TCP 报文封装到 DNS 隧道中, 具体各个工具的能力如表 8[101]所示, 根据 RFC1034 [103], 域名最长为 253 B, 一些工具采用 DNS 扩展机制 EDNS(0) [104] 运行 DNS 报文超过 512 B. 想要了解每个具体的工具, 请参考表 8[101]中相应的文献, 这里不再详述.

表 8 DNS 隧道工具[101]

分类	工具	编码方式	查询类型	支持EDNS(0)
IP over DNS	NSTX[105]	Base64	TXT	×
	Dnscat2[106]	Hexadecimal	A, AAAA, CNAME, MX, TXT	×
	Iodine[107]	Base32, Base64, Base128	A, CNAME, NULL, MX, TXT, SRV	√
	TUNs[108]	Base32, Base64	CNAME	×
TCP over DNS	Dns2tcp[109]	Base64	TXT, KEY	×
	OzymanDNS[110]	查询: Base32 响应: Base64	A, TXT	×
	Heyoka[111]	Binary	TXT	√

我们以图 8[101]为例说明隐秘隧道的建立和数据传输过程. 1) 受控主机将想要传送的数据构造到 DNS 查询请求的子域名中, 如“updata.tunnel.com”, 向本地递归解析服务器发送这个域名的 DNS 请求; 2) 递归解析服务器缓存中存在.com 的 TLD 服务器地址, 会直接向服务器查询由 C&C 服务器伪装的 tunnel.com 的权威服务器地址; 3) .com 服务器向本地递归解析服务器返回 tunnel.com 的权威服务器地址 A.B.C.D; 4) 本地递归解析服务器继续向这个地址迭代查询 updata.tunnel.com 的地址, 这样由 C&C 伪装的 DNS 服务器就可以从这个查询报文中提取它需要的 updata; 5) C&C 伪装的 DNS 服务器在它向本地递归解析服务器的响应报文中可以传送一些类似 downdata 之类的控制命令; 6) 最后, 递归解析服务器将响应报文传回受控主机, 就完成了受控主机和 C&C 服务器间的通信.

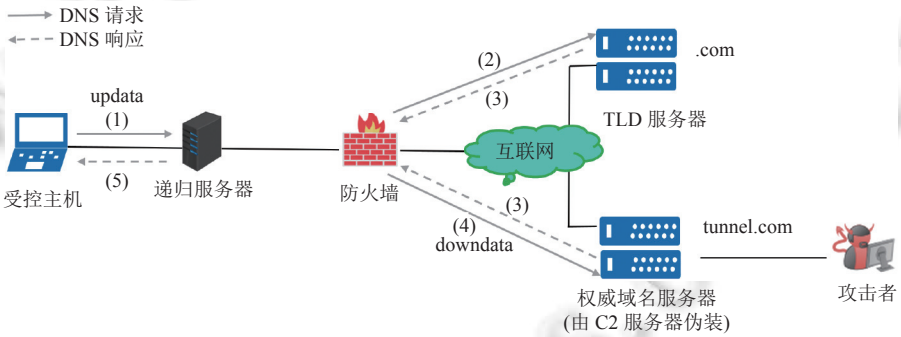

图 8 隐秘隧道攻击过程[101]

4.12 隐秘隧道检测

目前很多的企业网络为了尽可能地降低遭受网络攻击的风险, 一般将相关策略配置为仅允许与指定的可信任 DNS 服务器之间的流量通过, 这样就很好地避免了直连隧道引起的攻击, 但对于中继的 DNS 隧道, 这样的策略并不能阻止, 对于隐秘隧道的检测, 最直接的方法是特征提取和人工分析[112,113], 但是这种方法的检测率依赖于专业人士的分析能力, 且难以在目前的高速网络上应用.

文献 [101] 把隐秘隧道检测方法划分为基于规则的检测方法和基于模型的检测方法两大类, 基于规则的检测方法指手动设定某些特征的规则进行检测, 基于模型的检测方法是指用一些特征训练的模型自动产生检测的规则, 基于规则的检测方法又分为基于标志的和基于阈值的检测方法, 基于标志的检测方法主要通过字符串匹配的方式匹配字符串的特殊标记, 基于阈值的方法主要基于某个特征值超过设定阈值进行检测; 基于模型的方法分为

基于传统机器学习的方法和基于深度学习的方法; 后面我们就按基于标志、基于阈值、基于机器学习、基于深度学习这 4 类检测方法简要阐述隧道检测.

(1) 基于标志的方法

文献 [114,115] 利用开源的 Iodine 规则, 通过在 snort 上设定特定的规则检测 Iodine 隧道, 文献 [116] 利用 Dns2tcp 在 SSH 握手时的特别字符串, 在 Bro 上设定相应的规则检测 Dns2tcp 隧道. 这类方法的明显缺点是通用性差, 只能检测特定的隧道, 并且需要人工去分析特定规则的字符串, 容易出错.

(2) 基于阈值的方法

文献 [117] 把 DNS 通信中不同主机的数量设定一定阈值检测 DNS 隧道, 文献 [118] 把 DNS 流量的吞吐量作为判断的阈值, 文献 [119] 把域名查询次数的熵值作为判别的阈值, 文献 [120] 把每个内部主机通过外部 DNS 查询总的信息量的压缩值作为判断的阈值, 文献 [121] 基于 DNS 流量的一些统计特征, 用主元素分析和共有信息计算一个度量 mi 作为判断的阈值. 相比于基于标记的方法, 基于阈值的方法更加灵活, 实用性更广, 可以识别已知和未知的 DNS 隧道工具的攻击, 但是这种方法的测度在不同的网络环境阈值的大小不同, 需要人工分析调整.

(3) 基于机器学习的方法

基于机器学习的方法是检测 DNS 隧道比较典型的方法, 分为有监督和无监督的机器学习方法, 通常是提取流量特征用特定机器学习算法进行训练, 不同是无监督的方法只学习正常流量特征, 有监督的学习正常和异常两种特征, 由于这类方法比较多, 也比较典型, 我们将文献 [122] 中的典型方法示例说明, 其他文献的方法类似, 只是应用不同的流量特征和使用不同的机器学习算法, 具体比较如表 9[101]所示, 每个不同的方法详细请参考相应的文献, 这里不再详细阐述. 文献 [122] 方法工作流程如图 9[122]所示.

表 9 DNS 隐秘隧道机器学习检测方法对比[101]

文献	算法	恶意数据集工具	性能	缺点
[122]	Isolation foest	Iodine, Dns2tcp, FrameworkPOS, Backdoor.Win32.Denis	文中特定数据检测率达到100%同时保持较低的误报率	需更新域名白名单
[123]	Isolation forest	DNS exfiltration toolkit (DET)开源工具	检测率95%以上	假阳率高
[124]	RF	Dnscat2, DNSshell, cobaltstrike	检测率98.47%	需更新域名白名单
[125]	Logistic regression (LR), K-means	Dnscat, Bernhardpos	LR: 精度99.93%, 假阳率0.189%; K-means: 精度91.68%, 假阳率0.4%	检测TXT记录的隐秘通道性能差
[126]	Logistical regression, SVM, DT	Dns2tcp, Dnscat2, Iodine, OzymanDNS	SVM性能最高: 检测精度99.96%, 回召率99.93%	难以检测未知的DNS隧道
[127]	K-means, 单类支持向量机 (OCSVM)	SlowDNS	OSSVM性能最高: F1-Score 96%	缺少其他性能的实验, 且数据集小扩展性差
[128]	Logistic regression	Dns-grind, Dns2tcp, Dnscapy, Dnscat2, Iodine, OzymanDNS	检测率99.99%, 回召率99.99%	难以检测未知的DNS隧道

首先进行 DNS 数据收集, 这个过程每隔 M min 处理 DNS 日志, 生成如下元组$<Q, R, T>_j$, 其中 Q 代表请求的完整的域名, R 代表应答的值, T 代表请求的记录类型, j 代表日志文件的行索引值, 一个域名的主域名定义为 2-LD 和 1-LD 的连接, P_j=prim($<Q, R, T>_j$); 这样, t 个离散时间段中相同的主域名可以划分到一个组内: L_iP_i={$<Q, R, T>_j$|P_i=prim($<Q, R, T>_j$)}. 然后进行特征提取, 每个滑动窗口 WP_i, 进行一次特征提取, 得到相应的特征向量: fe(WP_i)=< P_i, 域名的字符数, 域名中大写字母数, 域名中数字数, 字符的熵值,… >. 从提取的特征向量中抽取这样的数据进行选取合适的无监督机器学习模型 Isolation forest 进行训练, 就可以得到相应用于检测的基线模型, 来检测这类由 DNS 秘密隧道引起的数据窃取, 恶意代码执行等事件.

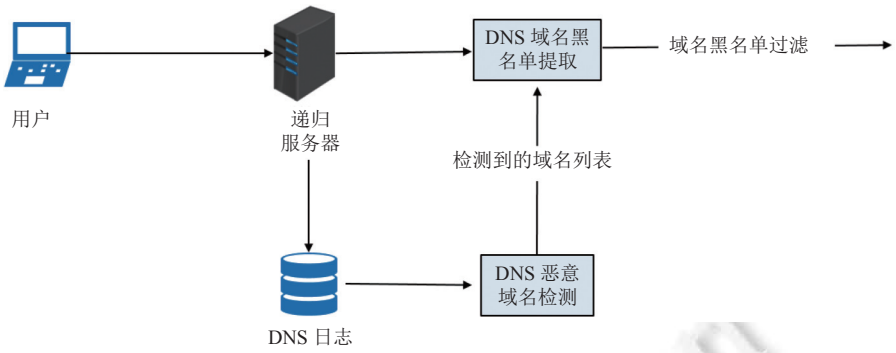

图 9 隐秘隧道检测过程[122]

(4) 基于深度学习的方法

深度学习不需要像机器学习那样人工提取 DNS 流量特征, 深度学习可以自动抽取原始数据的特征, 文献 [129] 利用前向神经网络作为训练模型, 把 DNS 报文的前面 512 个字节作为输入的数据, 最终的检测率可以达到 99.96%, . 文献 [130] 利用卷积神经网络, 他们将 DNS 报文转换成一个固定长度的字节向量, 通过嵌入层降维处理后进行训练, 最终可以达到 99.98% 的检测率, 优于传统的机器学习方法. 但是深度学习需要的数据量和硬件性能都比较高, 难以在线部署.

4.13 配置错误

一个 DNS 服务器含有不同的 zone 文件集合, DNS 解析主要来自它的 zone 文件的 RR, 因此 zone 文件是解析的关键, 这些 zone 文件通常有不同的组织管理, 操作错误和配置变化会导致 DNS 服务器的 zone 文件产生很多解析错误, 比如授权不一致、无效授权、丢失胶水记录、不存在的域名服务、应答不一致等[131−133], 具体如表 10[133]所示.

表 10 DNS 配置错误引起的 zone 文件问题[133]

错误类型	具体含义
委托不一致	对于委托父子节点的区文件中NS和A胶水记录不一致
错误委托	一个权威域名服务器不能提供正确的权威域名应答
丢失胶水记录	名字服务器的区文件的NS记录缺少对应的胶水记录
不存在的域名	对于一个可以解析的域名返回NXDOMAIN应答
区文件依赖循环	用于解析某个DNS请求的区文件Z1依赖Z2, Z2又依赖Z1
重写循环	DNS请求存在重写循环: q1→q2→q3→…→q1
请求超长	一个请求q1最终重写到qn, 而qn超过了域名的最大长度值
应答不一致	DNS应答在不同的执行中不一致
TTL值为0	存在一个请求导致应答的RR中的TTL值为0, 将无法缓存
重写黑洞	存在一个请求q1最终重写到qn, 而qn的应答为NXDOMAIN

4.14 配置错误检测

目前有些 DNS 诊断工具[134]可以帮助检测已知特定的 zone 文件的 DNS 配置错误, 但都比较局限, 文献 [131] 设计了一种利用多个监测点检测 DNS 配置错误的工具, 用于检测和正确配置相违反的情况, 而不是只针对特定的 DNS 配置错误, 主要可以检测到表 10 中前面 4 种配置错误.

文献 [133] 提出 GROOT 方案, 为了验证 DNS 的行为, 为 DNS 建立一个正式的数学语义模型, 把各种请求、查找和解析过程进行语义处理, 包括权威名字服务器的查询过程和递归解析服务器的解析过程, 然后把相同解析过程的 DNS 请求构建等价类, 具体为先进行 Label 图的构建, 如图 10[133], 是所有 DNS 服务器的所有 zone 文件中所有请求域名的集合, 红虚线表示 DNAME 记录, 这样从每个节点到根节点就会产生一个等价类 EC.

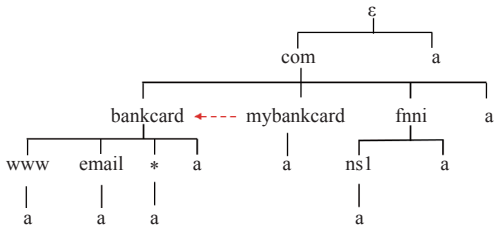

图 10 Label 图[133]

对每个等价类用建立的 DNS 数学语义模型进行符号执行, 这样符号执行算法就为每个等价类建立一个解释图, 代表了这个等价类所有 DNS 可能的执行情况, 由于解释图包含了每种执行情况所有的信息, 这样通过编写一个属性检测工具来检测每个解释图的执行中可能的错误情况. 这样, 就完成了对 zone 配置文件的验证, 可以用来检测由于配置变化或错误操作导致的 zone 解析错误的问题, 以帮助相应的管理人员及时修复.

4.15 DNS 服务软件漏洞

编写一个高效的, 多线程的 DNS 服务器实施系统, 同时在执行过程中完全符合 DNS RFC 的要求且不出现 BUG 在现实中是非常困难的, 常用的开源 DNS 服务软件如 BIND、POWERDNS、NSD、Knot 等在实现时本身系统也会有漏洞, 执行错误, 安全问题, 从而引起部署这些软件的 DNS 服务系统在实际运行中出现错误和安全问题[135−138].

4.16 DNS 服务软件漏洞检测

文献 [138] 提出一个 DNS RFC 协议检查验证工具 SCALE, 来查验 DNS 服务器在执行过程中和 RFC 协议要求不一致的情况, 这样要求工具自动生成测试用例可以覆盖 RFC 中要求的 DNS 的行为, 每个测试用例不仅要有 DNS 请求, 还要有相应的 zone 文件, 并且请求要和 zone 文件紧密关联, 以执行关键的请求解析逻辑, 用这些测试用例来验证不同 DNS 服务器部署系统的运行情况. 如图 11[138]所示.

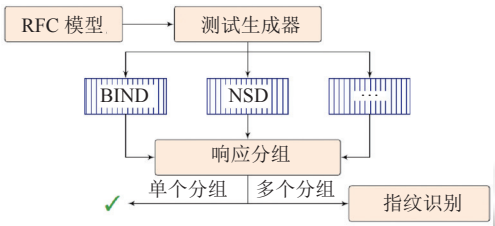

图 11 服务软件漏洞检测流程[138]

SCALE 的核心问题在于用 RFC 定义一个 DNS 解析过程的逻辑行为模型, 用这个模型来指导测试用例的生成. DNS 数学语义模型在 GROOT[133]中设计完成, 用于在一个 DNS 服务器上处理 DNS 请求, 整个执行过程见图 12[138], 可以编写成一个无状态的函数, 输入是一个查询请求和 zone 文件, 输出是 DNS 应答, 首先用 SELECTBESTRECORDS 函数选择 zone 文件中最近的匹配记录, 然后按图 12 中的决策逻辑执行, 其中精准类型表示查询匹配到的是 A 类型等直接可以返回给用户应答结果的类型, 每个叶子节点代表了 DNS 执行的一个情况.

下面, 通过符号执行 DNS 模型来生成 zone 文件, RFC 定义了许多限制来减少 zone 文件的错误, 如表 11[133]所示, 由于模型是按 RFC 协议规范建立的, 无论符号执行模型到哪条路径, 都可以保证生成的 zone 文件是符合规范的. 除了生成正确的 zone 文件, 还要生成一些违反表 11 规范的 zone 文件, 可以在生成正确 zone 文件的基础上, 系统生成一些违反表 11 任意一个条件的 zone 文件, 通常情况下, DNS 服务器的 DNS 系统软件如 BIND 等都会对 zone 文件进行语法检测, 或者拒绝或者把错误的 zone 文件转换成正确的, 如果拒绝就不存在问题, 如果接受(不是所有的系统软件都能修复文件, 也没有任何一个系统软件能修复所有错误), 就可能出现解析错误.

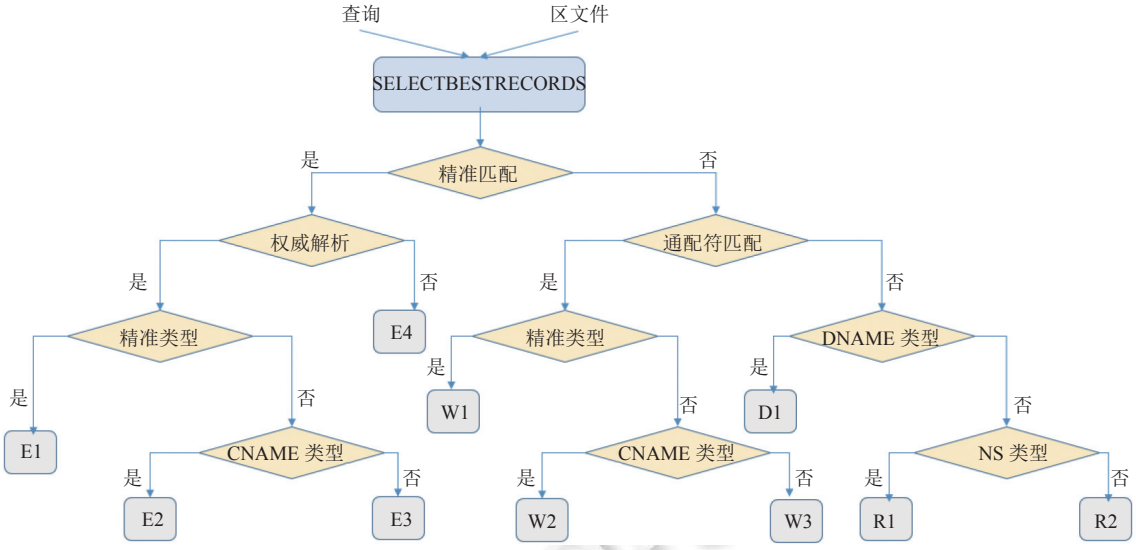

图 12 DNA 解析逻辑行为模型图[138]

表 11 RFC 定义的 RR 要求[133]

验证条件	RFC文档
1. 区文件中所有记录应该是唯一的, 不能有重复	2181
2. 一个区文件应该只有一条SOA记录	1035
3. 区的域应该是所有RR域名的前缀	1034
4. 一个域名只能有一个CNAME, 且如果有CNAME类型就不能有其他类型	1034
5. 一个域名只能有一个DNAME记录	6672
6. 一个域名不能同时有DNAME和NS记录, 除非同时有SOA记录	6672
7. DNAME记录的域名不能是另外一个记录域名的前缀	6672
8. NS记录不能有一个非SOA域名是另外一个NS记录的前缀	1034
9. 区文件中所有的NS记录必须有对应的胶水记录	1035

最后, 通过将生成的 zone 文件生成查询等价类, 用等价类中一个代表性的查询请求作为这个 zone 文件的 DNS 请求, 这样所有的测试例子 (zone 文件和对应的请求) 就都生成完毕, 就可以对不同的 DNS 服务器实施系统进行测试, 来检测这些系统自身部署的异常情况, 以帮助开发者及时修补漏洞, 解决 DNS 的系统风险和操作风险, 实现 DNS 系统的验证机制.

5 DNS 增强保护技术

除了对恶意域名和 IP 的检测技术外, 还有很多工作用于增强 DNS 自身的安全性和健壮性, 如在基础协议上部署增强的安全协议, 对 DNS 架构的改进, 使用 Anycast 技术抵御 DDoS 攻击, 对恶意域名进行阻断, 而对重要域名进行保全, 对攻击源进行追踪等, 我们按照不同的增强技术大致分为下面几类分别简要阐述.

5.1 安全协议

(1) DNS 安全扩展 (domain name system security extensions, DNSSEC)

DNSSEC 是由 IETF 提供的一系列 DNS 安全认证的机制 (RFC 4033–RFC 4035), 利用数字签名和公钥来实现 DNS 数据的完整性与可靠性, 抵御 DNS 劫持, DNSSEC 并不对数据进行加密, 只是验证访问的域名是否有效, 需要从根区到最终 FQDN 的查找过程中的每一步部署 DNSSEC, DNSSEC 利用上层的根和权威域名服务器的私钥

签署 RR, 解析服务器或下一级的权威服务器用上一级的根或权威服务器的公钥进行认证, 确定是是否是真正的上级根或权威服务器的数据, 如果验证成功, 则认为 DNS 数据信息来自可以信任的上级根或权威域名服务器, 认证成功后再进行解析服务器接收数据, DNSSEC 可以有效提高对 DNS 数据来源的可靠性, 防止 DNS 劫持, 但不能保证 DNS 信息的保密性. 文献 [139] 对全球 DNSSEC 的部署情况进行了测量, 发现前 3 个 TLD (.com, .net, .org) 中只有 0.1% 发布 DNSKEY, 而且 DNSSEC 不正确或者不完全的部署还可以导致域名劫持[140].

(2) DNSCurve

DNSCurve[141]使用 Curve25519 椭圆曲线加密算法创建密钥, 用来加密和验证解析器与域名服务器之间的 DNS 报文, 因此相比于 DNSSEC, 它通过验证 DNS 响应报文提供和 DNSSEC 类似的完整性支持, 不同的是, 它通过加密报文保证 DNS 报文的机密性, 此外, 它能更迅速地辨别伪造的数据包, 从而使攻击者更难达到阻止 DNS 数据通过的目的, 而在一定程度上保证 DNS 服务的可用性, 相对于 DNSSEC, 它的数据计算量和额外流量更小. 缺点是 DNSCurve 需要修改 DNS 协议, 增量部署有很大限制, 目前除了 OpenDNS[142]宣布支持 DNSCurve 外, 尚无其他 DNS 供应商明确部署 DNSCurve.

(3) 用户和解析器间的加密安全协议

除了 DNS 服务器之间的安全协议外, 从用户端到递归解析服务器之间的通信也至关重要, 如果没有加密措施, 很容易受到信息窃取或其他攻击, 目前共有 5 个用户和解析器间的加密协议, 包括 2016 年 RCF7858 提出的 DNS-over-TLS (DoT), 2018 年 RFC8484 提出的 DNS-over-HTTPS (DoH), 2017 年 RFC8094 提出的 DNS-over-DTLS, 2017 年 DNS-over-QUIC 草案, 2011 年提出的 DNSCrypt.

DoT 使用 TLS 协议加密传输用户和递归解析服务器之间的 DNS 消息, 起到防止中间用户窃听和域名查询隐私泄漏的作用.DoH 运行与 HTTPS 之上 (本身由 TLS 保护), 在用户通过浏览器进行 HTTPS 访问时将 DNS 查询消息直接嵌入到 HTTPS 报文中, 不需要专门的 TLS 握手建立连接, 因此, DoT 和 DoH 通信端口不同, DoT 使用专门的端口 853, DoH 使用 HTTPS 的端口 443. 目前, DoT 已取得一些 DNS 软件比如 Unbound[143]和 Stubby[144]等支持, DoH 在一些浏览器比如 Firefox[145]和 360 浏览器等提供支持, 另外, 一些 DNS 解析服务商比如 Cloudflare[146], Google[147]和 Quad9[148]中支持都支持 DoH 和 DoT.

DNS-over-DTLS 是 DoT 的一个变体, 为获取更好性能运行在 UDP 之上, 目前尚无系统部署. DNS-over-QUIC 也是为了提高 DoT 性能做的改进, 目前尚无部署. DNSCrypt 也是通过 HTTPS 报文进行传输, 作为最早提出的加密协议, 目前也有一些公共解析提供商 OpenDNS[142], Yandex[149], OpenNIC[150]提供支持. 文献 [151] 对全球的 DNS 解析器的加密部署情况作了第 1 次大的测量发现, 相比传统的 DNS 明文消息, 加密的消息难以被攻击者窃取, 然而现实部署中存在很多问题, 比如约有 25% 的 DoT 服务商用不正确的 SSL 认证, 现实网络中部署加密协议的用户很少, 但目前有上升的趋势. 同时文献 [151] 对这些加密协议进行了对比, 具体区别如表 12[151]所示.

表 12 DNS 加密安全协议对比[151]

范畴	准则	DoT	DoH	DNS-over-DTLS	DNS-over-QUIC	DNSCrypt
协议设计	使用应用层协议	○	●	○	○	●
	提供回调机制	●	○	●	●	○
安全	使用标准的TLS	●	●	●	●	○
	抵抗DNS流量分析	◎	●	◎	◎	●
可用性	对客户端用户改的较小	◎	●	○	○	◎
	相对于UDP上的DNS延迟较小	◎	◎	●	●	◎
可部署性	运行在标准协议上	●	●	●	○	○
	获得主流的DNS软件支持	●	◎	○	○	◎
成熟度	已被IETF标准化	●	●	●	○	○
	获得解析器的广泛支持	●	●	○	○	◎

注: ●表示全支持, ○表示不支持, ◎表示半支持

5.2 服务器增强技术

(1) Anycast

部署 Anycast[152]是利用网络路由的技术方案来增强 DNS 的安全性能. 通过对 Anycast 的部署, 可以实现 DNS 中提供相同服务的服务器组公用统一的 IP. 客户端向 DNS 发送的数据连接请求可以利用 Anycast 接入最近的一台服务器主机上, Anycast 通过将 DNS 的访问分散到不同的服务器上, 可以在一定程度上降低 DDoS 等攻击对 DNS 用户的影响, 目前根及顶级域名服务器采用的 Anycast 技术, 及权威服务器基于 CDN 的流量均衡技术, 有效的防御了近些年对 DNS 基础设施的 DDoS/DoS 攻击[153,154].

(2) 单机性能

RFC 2870[155]从 2000 年就推荐 DNS 根服务器对 DNS 查询的处理能力要是峰值负载的 3 倍, Kont DNS 的开发人员测试了他们所有部署 Knot 的权威域名服务器[156], 发现每个服务器至少需要处理 50 万 QPS 的查询才不至于丢弃 DNS 查询报文, Bind 的开放人员测试发现部署 Bind 的解析服务器需要至少处理 40 万 QPS 到 50 万 QPS 的 DNS 查询报文[157], 但是相同的测试标准在目前较新的硬件上机器可以处理的查询次数可达 500 QPS, 相应的测试工具参见文献 [158]. 另外, 在解析服务器上使用 CACHE 机制, 并且尽可能增加 CACHE 的空间, 可以使用户大部分的查询在解析器的 CACHE 中返回, 目前 90% 以上 CACHE 命中率是比较常见的, 良好的 CACHE 机制大大减轻了网络负载和加快了查询速度.

(3) 硬化技术

对服务器可以采取一些硬化技术来抵御攻击, 比如 DNS 服务器常被用于增强反射攻击, 为了应对这种误用, 一种常用的机制是降低对 DNS 查询请求的响应速率 (response rate limiting, RRL)[159], 常用方式是名字服务器记录每个解析器重复的查询次数, 如果一个解析器的查询重复次数超过设定的阈值, 则相应的 DNS 响应就会进行丢弃或减少. 另外一种类似的方式是减少响应报文的大小, 特别是对 ANY 类型的查询[160]. 此外, 在解析器可以使用 QNAME 最小化技术大大减少迭代查询数量[161], 比如对域名 www.na.icar.cnr.it 的查询, 如果解析器知道 ns1.cnr.it 的权威服务器地址, 直接向该地址进行发送 QTYPE=NS, QNAME=icar.cnr.it 查询. 最后, 对安全要求比较严格的场景可以对解析器进行访问控制限制, 只对特定用户开放访问权限.

5.3 DNS 系统增强

DNS 系统增强技术需要改进实现 DNS 协议的系统, 从协议改进层面进行增强, 但是这类增强方式通过对现有解析系统改造虽然能取得一定安全优势, 但也可能带来其他问题, 因此难以大范围部署, 比如 T-DNS[162]使用 TCP 和 TLS 协议替代 UDP 进行安全传输, CoDNS[163]利用 DNS 服务器协同机制预防单点失效, DR-DNS[164]使用多 DNS 服务器软件副本, 通过投票机制确定解析结果, 防止某个 DNS 软件设计的漏洞, 这 3 种机制难免会造成解析效率的降低.

PageDNS[165]通过对 TLD 和解析器的改造防止根和 TLD 服务器层面的信息泄露, 用户和其他域名服务器仍然用传统的 DNS 协议, 它主要通过将属于 TLD 区文件的响应记录进行分页, 这样解析器得到的解析结果是一页而不是单个记录, 达到防治信息泄露的目的. Oblivious DNS (ODNS)[166]通过改造用对称加密递归解析服务器的交互双方防止递归解析服务器相关信息的泄露.

5.4 体系结构增强

体系结构增强技术旨在突破目前 DNS 解析体系结构的限制, 用目前一些新的技术解决 DNS 系统的安全性、可靠性、健壮性等问题, 有下面几类.

(1) 基于 P2P 网络的分布式结构

利用 P2P 网络的优势对目前 DNS 系统去中心化, 基于 Chord[167]的 DDNS[168]利用分布式哈希表完成 DNS 记录的存储和检索, P-DONAS[169]和 CoDoNS[170]通过 P2P 网络建立现有域名系统的缓存系统, 用户先在基于 P2P 节点的缓存系统中查找, 如果没找到, 由 P2P 系统节点向传统 DNS 服务器交互并更新缓存, HDNS[171]将 TLD 和二级域名部分用 P2P 网络组织, 下层结构仍沿用传统 DNS 的树形结构, 这种混合结构一定程度上结合了现有域名系统

和 P2P 系统的优势, 但总体来讲, 基于 P2P 结构的域名系统存在查询延迟大, 节点信息不一致、P2P 网络本身易被攻击的缺陷.

(2) 基于区块链的 DNS 系统

区块链网络本身基于 P2P 网络实现的一个分布式账本系统, 具有去中心化和安全性好的网络特征, 基于区块链的 Bitcoin 系统, 文献 [172] 提出 Namecoin 方案, 将区块链交易信息替换为名称-数值映射, Blockstack[173]通过将域名数据和控制进行分层进一步提高 Dot-Bit 方案的扩展性, 区块链只实现控制, 而实际的域名在外部数据库中进行存储, 文献 [174] 提出 B-DNS 以改进目前区块链 DNS 系统的计算量大和查询效率低的问题, 另外, 还有其他一些区块链系统的域名服务, 如基于 Ethereum 的 ENS[175]、基于 Emercoin 的 EMCDNS[176]等.

虽然基于区块链的域名系统利用区块链自身的特性增强了 DNS 的安全可用性, 防止了劫持攻击, 但同时也存在区块链系统自身固有的缺陷, 比如区块链节点的加入缺乏认证机制, 容易遭恶意节点破坏, 需要存储过多的历史信息, 与传统 DNS 系统不兼容, 文献 [177] 对 Namecoin[172]和 Emercoin[176]的恶意攻击进行了研究, 发现区块链域名系统被恶意攻击的主要原因是被这些域名注册的 50.7% 的 IP 地址本身就曾被报告出现过恶意事件.

(3) 联盟根系统

还有些方案对 DNS 根服务器现有结构进行改造, 以解决过度中心化问题: 1) RFC8806 提出的本地根可以直接在递归服务器上进行根区文件镜像, 避免向根服务器的迭代查询; 2) 通过 Anycast 技术增加根服务器的全球影子节点; 3) 把递归服务器的根区查询设置到自主建设的一组根服务器上[178], 这组根服务器形成自主根联盟, 各个根服务器可以自主发布权威域名信息, 并互相扩散; 4) TD-Root[179]通过区块链对根服务器进行分布式管理, 每个根服务器通过共识算法维护一个一致的根区文件, 可以实现安全可信的根服务器管理, 并能对 1/3 的恶意根节点进行容错.

5.5 反向追踪技术

为了抵御攻击, 检测到攻击后能有效地对攻击进行溯源可以从源头定位攻击, 并制定反制措施, 精准的溯源技术可以对攻击者形成极大的网络震慑力, 由于目前源地址验证[180]还没有大量部署, 在巨大的网络中对攻击者进行溯源是一项非常困难的事情, 尽管目前一些溯源技术如 ICMP 溯源、报文标记、链路测试等可以用于追踪 DDoS 攻击, 但需要在网络中进行额外部署, 并且对其他攻击不一定适用, 文献 [181] 通过压缩报文标记算法和随机业务抽样进行溯源, 但需要较长的攻击时间和大量的攻击报文, 且目前的 DNS 攻击很多通过伪造攻击 IP 地址, 加大了溯源的难度.

5.6 攻击阻断 (过滤)

通过网络公开的域名黑名单[182]和检测到的恶意域名, 建立域名黑名单, 在 DNS 解析器中通过触发器设置阻断或过滤机制. DNS 触发器可根据用户查询请求触发阻断, 常见的有: 1) QNAME 触发器根据查询的 NAME 字段判断, 可以利用通配符阻止站点和所有子域, 如*.pupple.com 阻断前面所有子域; 2) IP 触发器与 DNS 响应中资源记录的 IP 地址匹配, 用于阻断已知的恶意 IP 地址; 3) 客户端 IP 触发器与发起查询的客户端的源 IP 地址匹配, 用于阻止已知恶意主机; 4) NSDNAME 触发器与 NS 记录的域名服务器的名称匹配, 用于阻断已知恶意的权威域名服务器; 5) NSIP 触发器与 NS 记录的域名服务器胶水记录的 IP 地址匹配, 用于阻止特定恶意的权威域名服务器.

常用的实现触发器的方法是通过设置响应策略区域 (response policy zone, RPZ)[183], RPZ 由 ISC 首度发明并在 BIND 中实现, 它是一个开放的厂商中立的标准, 用于交互 DNS 防火墙配置信息, 它是一种配置在特殊的 DNS 的 zone 文件中的 DNS 防火墙, 目前大部分 DNS 服务软件都具有 RPZ 功能. RPZ 为 DNS 服务器提供一种应答拦截机制, 匹配过滤特定的域名或 IP, 改变应答结果. RPZ 支持对查询域名、应答结果包含的 NS、A、AAAA 记录的 Name、RData 等数据进行匹配过滤, 干预动作包括返回域名不存在 (NXDOMAIN)、无主机记录 (NODATA)、要求客户端以 TCP 协议重新发起请求, 以及指向另外一个自主设定的域名并返回该域名的应答结果等.

6 未来工作展望

本文系统总结了 DNS 面临的安全问题, 包括对这些安全问题的检测方法和防御方法, 尽管 DNS 检测和防御技术在不断发展, 但根据 IDC 的年报[2], DNS 安全事件和它们对社会和经济的影响仍然有上升的趋势, 未来针对 DNS 的攻击, 会从大流量逐步转为精细化的精准攻击, 威胁会更严重, 更难被发现, 现有的方法仍不能很好满足 DNS 安全检测防护的重要需求, 通过对现有方法的总结分析, 未来的研究工作可以关注以下几个方面.

(1) DNS 安全协议的研究

目前的 DNS 安全协议还都不能保证 DNS 全链路的安全, DNSSEC 提供了从根到递归解析服务器端的安全性验证, 可以确保所接收到的 DNS 数据来源的合法性和数据的完整性, 但不对数据内容进行加密, 无法解决 DNS 劫持风险, DoT/DoH 实现了基于证书的 DNS 通信双方身份认证和数据加密, 从而保证了数据不被攻击者窃听或篡改, 但 DoT/DoH 并不是全链路安全方案. 它工作在客户端到递归解析服务器之间, 递归服务之后的路径, 则需要额外的安全保护, 同时这些安全协议带来了计算资源、业务时延、监管和滥用等新的挑战, 使得它们在实际部署中使用量很低, 而且带来新的安全风险.

因此, 一方面, 对这些安全协议进行进一步深入研究, 解决它们在部署中面临的实际问题, 不断推动在实际应用中的部署是一个很值得后续的研究方向, 另一方面, 需要研究新的可以对 DNS 实施全链路保护的安全协议, 实现多安全机制的结合、多协议的结合和安全标准化的推广, 将更有利于 DNS 安全的发展, 同时, 在安全协议的研究过程中, 需要考虑个人隐私保护和 DNS 流量监管的矛盾, 个人用户通常希望自己的行为是完全自由的, 但完全脱离监管的网络也不是安全的, 加密流量可以提供隐私保护但增加了计算资源, 同时也让用户行为难以监管, 如何解决监管难题, 平衡隐私与监管之间的关系, 也会是 DNS 安全积极探索的方向.

(2) DNS 架构的研究

目前的 DNS 系统是层次树状结构, 根服务器由同一组织集中管理, 尽管目前实施了 Anycast 技术, 仍然存在被攻击和误用的风险, 目前已有一些解决方案, 比如基于 P2P 网络的分布式结构、基于区块链的 DNS 结构、基于联盟根的 DNS 系统, 它们试图一方面尽量使得 DNS 的架构更加扁平化, 另外一方面提出以共识的方式进行根的管理, 使得发布中心和管理中心尽量是同一组织, 实现对根的多方共同管理, 但是由于这些方法带来的新的效率问题和运营问题, 目前这些方案还未能实现真正大规模的实施和部署.

因此, DNS 架构仍然是一个值得探索的研究方向, 比如对 P2P 网络中接入控制的研究, 查询效率低下的改进, 在根共识方式中的高效共识算法的探索, 共识过程中分叉问题的解决方案, 节点存储的强一致、高可用的服务研究, 以保证总是能从可用的 DNS 根服务器中获取到最新的 zone 文件. 此外, 还可以探索一些新型网络的 DNS 服务系统, 比如基于目前的以数据为中心的命名数据网络 NDN, 文献 [184] 提出 NDN DNS (NDNS), 利用 NDN 网络本身的安全特征对 DNS 进行安全验证.

(3) 递归解析服务器域名保全的研究

除了建立域名黑名单, 对黑名单进行过滤和阻断外, 对重要的域名还需要建立白名单机制, 对这些白名单的关键域名在递归解析服务器进行保全, 实现关键域名的保全, 首先要对要进行域名探测技术的研究, 通过不同地点对不同 DNS 服务器探测的保全域名数据, 进行比对分析, 再加上一些权威的数据库搜取的信息, 确保基础数据的正确性, 在基础数据建立好后, 需要建立定期更新机制, 确保保全域名的时效性.

在保全策略研究方面, 可以探索对域名进行不同粒度的保全, 对探测的 FQDN 进行细粒度的保全, 在递归解析服务器侧直接返回结果, 这样就避免了对上层 DNS 服务器的迭代查询, 一方面增加了查询效率, 另一方面避免了迭代查询引起的劫持风险, 对其他级别的域名查询, 可以探索粗粒度的域名保全策略, 尽量让递归解析服务器避免从根服务器进行迭代查询, 而直接从知道的最近的权威服务器进行迭代查询, 以增加查询效率和避免劫持风险.

(4) 云 DNS 安全技术研究

随着云计算的兴起, 云服务正逐步成为主要的互联网基础服务设施, 但云服务因其开放性、虚拟化等特征, 相对于传统网络将面临更多安全问题, 目前各大云服务商如华为云、腾讯云、谷歌云等的用户都使用云服务商提供

的 DNS 解析服务, 特别是在全球新冠肺炎大流行期间, 使用混合云环境的企业增多, 部署在混合云的关键业务也在增加, DNS 攻击者越来越多地将目标对准云计算, 遭受云服务停机影响的公司也在逐年大幅增长[2].

因此, 对于云 DNS 安全技术的研究对目前的云服务安全非常重要, 特别是目前针对云 DNS 服务的 DDoS 攻击, 从之前的大流量 DDoS 演变成低信号攻击, 在规模和频次上逐渐上升, 带宽越来越便宜, 可利用设备越来越多, 研究云上 DNS 抵抗日益增长的高流量 DDoS 攻击, 是云服务稳定性的关键. 同时, 云服务提供的 DNS 解析会造成客户端到解析器间的较大的延迟, 特别在客户访问分布较广的 CDN 时, 针对这类问题进行探索, 设计高效、低延时的云 DNS 服务, 增强云服务商对安全方案的支持程度, 保证客户的 DNS 安全, 也在很大程度上保证了客户整个网络架构的安全.

(5) 高效检测技术的研究

对 DNS 数据的高效检测一直是学术界研究的热点, 目前仍然存在着不足和可改进的空间, 具体有:

1) DNS 流量的轻量级探测技术

目前的检测技术大多依赖对检测点被动测量的 DNS 流量, 流量相对受限, 而且相应的检测方法普适性较差, 未来需要研究轻量级的被动测量和主动测量结合的 DNS 数据采集方法, 部署多台探测服务器分布式并行探测不同地点的 DNS 流量, 同时实时对错误数据进行清洗, 以降低实际数据的存储量.

2) DNS 流量数据高效聚合技术

目前的检测技术大多基于采集的原始 DNS 流量直接进行特征提取或流量分析, 这对于单点的被动采集的流量可以满足, 但多于多点分布式采集的流量难以处理, 需要研究多点采集的流量的高效聚合技术, 比如多点采集的流量可以聚合到大数据平台上进行处理, 以发挥大数据平台分布式计算分析优势, 从而提高数据处理的实时性能.

3) 流量特征体系

目前不同的检测方法为了检测不同类型的异常提取了 DNS 流量中不同的特征, 没有形成一个特征的基准, 需要进一步研究可以涵盖尽量多异常的流量特征体系, 形成一套检测异常的流量特征标准体系, 由用户根据实际的网络状况配置选择使用.

4) 高效的检测模型

目前大多检测方法需要大量的历史样本数据作为训练集, 训练模型的时间较长, 训练的用于检测异常的基线模型比较依赖历史数据, 训练数据中的域名黑名单也不可能覆盖所有的恶意域名, 历史数据对检测结果影响比较大, 因此, 研究轻量级的在线检测模型, 以消耗比较小的计算资源和存储空间, 同时获得高的检测率和低的误检率, 一直是学者追求的目标, 也可以研究不需要训练历史数据而直接用特征流量统计进行检测的方法, 以避免数据污染的影响, 加快检测效率.

另外, 由于目前 DNS 加密部署的逐步增加, 如何检测加密流量的异常也是一个重要的研究方向. 除了对攻击行为引起的异常进行检测外, 还有一种是系统软件设计、编码和系统配置过程中的脆弱性所导致的异常, 对这类异常的检测和验证技术的研究对增强 DNS 系统的健壮性意义重大.

(6) 动态阻断防御技术

目前对恶意域名的阻断技术主要是通过建立域名黑名单, 然后对黑名单里面的域名进行阻断, 这通常需要动态更新维护域名黑名单, 但对于目前由 DGA 算法或是其他随机生成的恶意域名, 由于随机性强, 不可解析的域名量比较大, 而且很多就出现一次, 这样的域名放入域名黑名单中系统消耗和开销特别大, 而且也不及时, 因此, 需要研究相应的动态阻断防御技术, 比如可以根据攻击特点, 通常生成类似 23adasdaa2332scd.baidu.com 这样左侧随机的大量域名进行攻击, 攻击刚开始时, 这样的恶意域名相对较少, 随着恶意主机增加, 类似的恶意域名会大量增加, 根据这样的特点, 为了减少误判, 可以设置这样的动态阻断机制: 1) 收到这个二级域名的一个 FQDN 的应答为 NXDOMAIN; 2) 同一个二级域名的随机的 FQDN 超过一个阈值.

7 总 结

本文简要介绍了 DNS 工作原理和过程, 从递归侧角度全面梳理和总结了 DNS 工作过程中面临的安全问题,

包括由攻击或系统漏洞等引起各类安全事件, 各类安全事件的具体检测方法, 各类防御保护技术, 在对各类安全事件、检测和防御保护技术总结的过程中, 对相应方法的特点进行了深入分析, 并对一些典型的方法进行了对比, 并由此对未来此领域的研究方向做了一个展望, 希望抛砖引玉, 为从事 DNS 安全领域的研究的学者提供参考.

References:

[1] Wang Y, Hu MZ, Li B, Yan BR. Survey on domain name system security. Journal on Communications, 2007, 28(9): 91–103 (in Chinese with English abstract). [doi: 10.3321/j.issn:1000-436x.2007.09.015]

[2] IDC 2021 global DNS threat report. 2021. https://www.efficientip.com/resources/idc-dns-threat-report-2021/

[3] Liu WF, Zhang Y, Zhang HL, Fang BX. Survey on domain name system measurement research. Ruan Jian Xue Bao/Journal of Software, 2022, 33(1): 211–232 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6218.htm [doi: 10.13328/j.cnki.jos.006218]

[4] Wang WT, Hu N, Liu B, Liu X, Li SD. Survey on technology of security enhancement for DNS. Ruan Jian Xue Bao/Journal of Software, 2020, 31(7): 2205–2220 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6046.htm [doi: 10.13328/j.cnki.jos.006046]

[5] Hu N, Deng WP, Yao S. Issues and challenges of internet DNS security. Chinese Journal of Network and Information Security, 2017, 3(3): 13–21 (in Chinese with English abstract). [doi: 10.11959/j.issn.2096-109x.2017.00154]

[6] Kim TH, Reeves D. A survey of domain name system vulnerabilities and attacks. Journal of Surveillance, Security and Safety, 2020, 1(1): 34–60. [doi: 10.20517/jsss.2020.14]

[7] Khormali A, Park J, Alasmary H, Anwar A, Saad M, Mohaisen D. Domain name system security and privacy: A contemporary survey. Computer Networks, 2021, 185: 107699. [doi: 10.1016/j.comnet.2020.107699]

[8] Chandramouli R, Rose S. Challenges in securing the domain name system. IEEE Security & Privacy, 2006, 4(1): 84–87. [doi: 10.1109/MSP.2006.8]

[9] Zou FT, Zhang SY, Pei B, Pan L, Li LS, Li JH. Survey on domain name system security. In: Proc. of the 1st IEEE Int'l Conf. on Data Science in Cyberspace (DSC). Changsha: IEEE, 2016. 602–607. [doi: 10.1109/DSC.2016.96]

[10] Ramdas A, Muthukrishnan R. A survey on DNS security issues and mitigation techniques. In: Proc. of the 2019 Int'l Conf. on Intelligent Computing and Control Systems. Madurai: IEEE, 2019. 781–784. [doi: 10.1109/ICCS45141.2019.9065354]

[11] Usman Aijaz N, Misbahuddin M, Raziuddin S. Survey on DNS-specific security issues and solution approaches. In: Proc. of the Data Science and Security. Singapore: Springer, 2021. 79–89. [doi: 10.1007/978-981-15-5309-7_9]

[12] Van Der Toorn O, Müller M, Dickinson S, Hesselman C, Sperotto A, Van Rijswijk-Deij R. Addressing the challenges of modern DNS a comprehensive tutorial. Computer Science Review, 2022, 45: 100469. [doi: 10.1016/j.cosrev.2022.100469]

[13] The NSA and GCHQ's QUANTUMTHEORY hacking tactics. 2014. https://theintercept.com/document/2014/03/12/nsa-gchqs-quantumtheory-hacking-tactics/

[14] It should be easy to identify governments on the Internet. 2023. https://home.dotgov.gov/

[15] Reassessing RuNet Russian Internet isolation and implications for Russian cyber behavior. 2021. https://www.atlanticcouncil.org/wp-content/uploads/2021/07/RuNet-Issue-Brief-2021.pdf

[16] EU wants to build its own DNS infrastructure with built-in filtering capabilities. 2022. https://therecord.media/eu-wants-to-build-its-own-dns-infrastructure-with-built-in-filtering-capabilities/

[17] CIRA Canadian Shield. 2022. https://www.cira.ca/cybersecurity-services/canadian-shield

[18] Anonymous. The collateral damage of Internet censorship by DNS injection. ACM SIGCOMM Computer Communication Review, 2012, 42(3): 21–27. [doi: 10.1145/2317307.2317311]

[19] Liu BJ, Lu CY, Duan HX, Liu Y, Li Z, Hao S, Yang M. Who is answering my queries: Understanding and characterizing interception of the DNS resolution path. In: Proc. of the 27th USENIX Conf. on Security Symp. Baltimore: USENIX Association, 2018. 1113–1128.

[20] Dean D, Felten EW, Wallach DS. Java security: From HotJava to Netscape and beyond. In: Proc. of the 1996 IEEE Symp. on Security and Privacy. Oakland: IEEE, 1996. 190–200. [doi: 10.1109/SECPRI.1996.502681]

[21] Roskind J. Attacks against the Netscape browser. In: Proc. of the 2001 RSA Conf. Invited Talk. 2001.

[22] Dagon D, Provos N, Lee CP, Lee W. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In: Proc. of the 2008 Network and Distributed System Security Symp. San Diego, 2008.

[23] Cheng YN, Liu YL, Li C, Zhang ZX, Li N, Du YJ. In-depth evaluation of the impact of national-level DNS filtering on DNS resolvers over space and time. Electronics, 2022, 11(8): 1276. [doi: 10.3390/electronics11081276]

[24] Jackson C, Barth A, Bortz A, Shao WD, Boneh D. Protecting browsers from DNS rebinding attacks. ACM Trans. on the Web, 2009,

3(1): 2. [doi: 10.1145/1462148.1462150]
[25] Bortz A, Barth A, Jackson C. Google dnswall. 2007. http://code.google.com/p/google-dnswall/
[26] Top ten DNS attacks. 2020. https://info.infoblox.com/resources-ebooks-top-ten-dns-attacks
[27] IDC FutureScape: Worldwide it industry 2018 predictions. 2020. https://www.idc.com/getdoc.jsp?containerId=US43171317
[28] Vissers T, Barron T, Van Goethem T, Joosen W, Nikiforakis N. The wolf of name street: Hijacking domains through their nameservers. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 957–970. [doi: 10.1145/3133956.3133988]
[29] Hao S, Wang HN. Exploring domain name based features on the effectiveness of DNS caching. ACM SIGCOMM Computer Communication Review, 2017, 47(1): 36–42. [doi: 10.1145/3041027.3041032]
[30] Alonso R, Monroy R, Trejo LA. Mining IP to domain name interactions to detect DNS flood attacks on recursive DNS servers. Sensors, 2016, 16(8): 1311. [doi: 10.3390/s16081311]
[31] Lyu MZ, Gharakheili HH, Russell C, Sivaraman V. Hierarchical anomaly-based detection of distributed DNS attacks on enterprise networks. IEEE Trans. on Network and Service Management, 2021, 18(1): 1031–1048. [doi: 10.1109/TNSM.2021.3050091]
[32] Yoshida T, Kawakami K, Kobayashi R, Kato M, Okada M, Kishimoto H. Detection and filtering system for DNS water torture attacks relying only on domain name information. Journal of Information Processing, 2017, 25: 854–865. [doi: 10.2197/ipsjjip.25.854]
[33] Chen LG, Zhang YD, Zhao Q, Geng GG, Yan ZW. Detection of DNS DDoS attacks with random forest algorithm on spark. Procedia Computer Science, 2018, 134: 310–315. [doi: 10.1016/j.procs.2018.07.177]
[34] Recursive client rate limiting in bind 9.9.8, 9.10.3 and 9.11.0. 2021. https://kb.isc.org/docs/aa-01304
[35] Vixie P, Schryver V. DNS response rate limiting (DNS RRL). 2014. http://ss.vix.su/~vixie/isc-tn-2012-1.txt
[36] Hasegawa K, Kondo D, Tode H. FQDN-based whitelist filter on a DNS cache server against the DNS water torture attack. In: Proc. of the 2021 IFIP/IEEE Int'l Symp. on Integrated Network Management (IM). Bordeaux: IEEE, 2021. 628–632.
[37] Trevisan M, Drago I, Mellia M, Munafò MM. Automatic detection of DNS manIPulations. In: Proc. of the 2017 IEEE Int'l Conf. on Big Data. Boston: IEEE, 2017. 4010–4015. [doi: 10.1109/BigData.2017.8258415]
[38] Klein A, Shulman H, Waidner M. Internet-wide study of DNS cache injections. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Communications. Atlanta: IEEE, 2017. 1–9. [doi: 10.1109/INFOCOM.2017.8057202]
[39] Berger H, Dvir AZ, Geva M. A wrinkle in time: A case study in DNS poisoning. Int'l Journal of Information Security, 2021, 20(3): 313–329. [doi: 10.1007/s10207-020-00502-x]
[40] Zeng YW, Zang TN, Zhang YZ, Chen XX, Wang YP. A comprehensive measurement study of domain-squatting abuse. In: Proc. of the 2019 IEEE Int'l Conf. on Communications (ICC). Shanghai: IEEE, 2019. 1–6. [doi: 10.1109/ICC.2019.8761388]
[41] Agten P, Joosen W, Piessens F, Nikiforakis N. Seven months' worth of mistakes: A longitudinal study of typosquatting abuse. In: Proc. of the 2015 Network and Distributed System Security Symp. San Diego, 2015. [doi: 10.14722/ndss.2015.23058]
[42] Nikiforakis N, van Acker S, Meert W, Desmet L, Piessens F, Joosen W. Bitsquatting: Exploiting bit-flips for fun, or profit? In: Proc. of the 22nd Int'l Conf. on World Wide Web. Rio de Janeiro: ACM, 2013. 989–998. [doi: 10.1145/2488388.2488474]
[43] Holgers T, Watson DE, Gribble SD. Cutting through the confusion: A measurement study of homograph attacks. In: Proc. of the USENIX Annual Technical Conf. Boston: USENIX Association, 2006.
[44] Nikiforakis N, Balduzzi M, Desmet L, Piessens F, Joosen W. Soundsquatting: Uncovering the use of homophones in domain squatting. In: Proc. of the 17th Int'l Conf. on Information Security. Hong Kong: Springer, 2014. 291–308. [doi: 10.1007/978-3-319-13257-0_17]
[45] Kintis P, Miramirkhani N, Lever C, Chen YZ, Romero-Gómez R, Pitropakis N, Nikiforakis N, Antonakakis M. Hiding in plain sight: A longitudinal study of combosquatting abuse. In: Proc. of the 2017 ACM SIGSAC Conf. on Computer and Communications Security. Dallas: ACM, 2017. 569–586. [doi: 10.1145/3133956.3134002]
[46] Alrwais S, Yuan K, Alowaisheq E, Li Z, Wang XF. Understanding the dark side of domain parking. In: Proc. of the 23rd USENIX Conf. on Security Symp. San Diego: USENIX Association, 2014. 207–222.
[47] Vissers T, Joosen W, Nikiforakis N. Parking sensors: Analyzing and detecting parked domains. In: Proc. of the 2015 Network and Distributed System Security Symp. San Diego, 2015. [doi: 10.14722/ndss.2015.23053]
[48] Mekky H, Torres R, Zhang ZL, Saha S, Nucci A. Detecting malicious HTTP redirections using trees of user browsing activity. In: Proc. of the 2014 IEEE Conf. on Computer Communications (INFOCOM 2014). Toronto: IEEE, 2014. 1159–1167. [doi: 10.1109/INFOCOM.2014.6848047]
[49] Banerjee A, Barman D, Faloutsos M, Bhuyan LN. Cyber-fraud is one typo away. In: Proc. of the 27th Conf. on Computer Communications (INFOCOM 2008). Phoenix: IEEE, 2008. 1939–1947. [doi: 10.1109/INFOCOM.2008.258]
[50] Li Z, Zhang KH, Xie YL, Yu F, Wang XF. Knowing your enemy: Understanding and detecting malicious Web advertising. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. Raleigh: ACM, 2012. 674–686. [doi: 10.1145/2382196.2382267]

[51] Wang YM, Beck D, Wang J, Verbowski C, Daniels B. Strider typo-patrol: Discovery and analysis of systematic typo-squatting. In: Proc. of the 22nd Conf. on Steps To Reducing Unwanted Traffic on the Internet. San Jose: USENIX Association, 2006.

[52] Suzuki H, Chiba D, Yoneya Y, Mori T, Goto S. ShamFinder: An automated framework for detecting IDN homographs. In: Proc. of the 2019 Internet Measurement Conf. Amsterdam: ACM, 2019. 449–462. [doi: 10.1145/3355369.3355587]

[53] Maroofi S, Korczyński M, Duda A. From defensive registration to subdomain protection: Evaluation of email anti-spoofing schemes for high-profile domains. In: Proc. of the 2020 Network Traffic Measurement and Analysis Conf. 2020.

[54] Rijswijk-Deij R, Sperotto A, Pras A. DNSSEC and its potential for DDoS attacks. In: Proc. of the 2014 Conf. on Internet Measurement Conf. Vancouver: ACM, 2014. 449–460. [doi: 10.1145/2663716.2663731]

[55] Kamra A, Feng H, Misra V, Keromytis AD. The effect of DNS delays on worm propagation in an IPv6 Internet. In: Proc. of the 24th IEEE Annual Joint Conf. of the IEEE Computer and Communications Societies. Miami: IEEE, 2005. 2405–2414. [doi: 10.1109/INFCOM.2005.1498526]

[56] Zink T. How spammers get around SPF. Technicl Report, 2007. http://www.circleid.com/posts/782012_spammer_get_around_spf/

[57] Aizuddin AA, Atan M, Norulazmi M, Noor MM, Akimi S, Abidin Z. DNS amplification attack detection and mitigation via sflow with security-centric SDN. In: Proc. of the 11th Int'l Conf. on Ubiquitous Information Management and Communication. Beppu: ACM, 2017. 3. [doi: 10.1145/3022227.3022230]

[58] Ballani H, Francis P. A simple approach to DNS DoS mitigation. In: Proc. of the 2006 Workshop on Hot Topics in Networks. 2006. 67–78.

[59] Verma S, Hamieh A, Huh JH, Holm H, Rajagopalan SR, Korczynski M, Fefferman N. Stopping amplified DNS DDoS attacks through distributed query rate sharing. In: Proc. of the 11th Int'l Conf. on Availability, Reliability and Security. Salzburg: IEEE, 2016. 69–78. [doi: 10.1109/ARES.2016.93]

[60] Krämer L, Krupp J, Makita D, Nishizoe T, Koide T, Yoshioka K, Rossow C. AmpPot: Monitoring and defending against amplification DDoS attacks. In: Proc. of the 18th Int'l Symp. on Research in Attacks, Intrusions, and Defenses. Kyoto: Springer, 2015. 615–636. [doi: 10.1007/978-3-319-26362-5_28]

[61] Al-Dalky R, Rabinovich M, Allman M. Practical challenge-response for DNS. ACM SIGCOMM Computer Communication Review, 2018, 48(3): 20–28. [doi: 10.1145/3276799.3276802]

[62] Anagnostopoulos M, Lagos S, Kambourakis G. Large-scale empirical evaluation of DNS and SSDP amplification attacks. Journal of Information Security and Applications, 2022, 66: 103168. [doi: 10.1016/j.jisa.2022.103168]

[63] Anagnostopoulos M, Kambourakis G, Kopanos P, Louloudakis G, Gritzalis S. DNS amplification attack revisited. Computers & Security, 2013, 39: 475–485. [doi: 10.1016/j.cose.2013.10.001]

[64] Kammas P, Komninos T, Stamatiou YC. Modeling the co-evololution DNS worms and anti-worms in IPv6 networks. In: Proc. of the 5th Int'l Conf. on Information Assurance and Security. Xi'an: IEEE, 2009, 171–174. [doi: 10.1109/IAS.2009.334]

[65] Whyte D, Kranakis E, van Oorschot P. DNS-based detection of scanning worms in an enterprise network. In: Proc. of the 2005 Network and Distributed System Security Symp. San Diego, 2005.

[66] Van der Toorn O, van Rijswijk-Deij R, Geesink B, Sperotto A. Melting the snow: Using active DNS measurements to detect snowshoe spam domains. In: Proc. of the 2018 IEEE/IFIP Network Operations and Management Symp. IEEE, 2018. 1–9. [doi: 10.1109/NOMS.2018.8406222]

[67] Cisco Talos Intelligence Group. Covert channels and poor decisions: The tale of DNSMessenger. 2017. https://blog.talosintelligence.com/2017/03/dnsmessenger.html

[68] Cisco Talos Intelligence Group. Spoofed SEC emails distribute evolved DNSMessenger. 2017. https://blog.talosintelligence.com/2017/10/dnsmessenger-sec-campaign.html

[69] Security Stability Advisory Committee, Los Angeles, CA, USA. SAC 025: SSAC advisory on Fast-Flux hosting and DNS. 2008. https://www.icann.org/en/system/files/files/sac-025-en.pdf

[70] Holz T, Gorecki C, Rieck K, Freiling FC. Measuring and detecting Fast-Flux service networks. In: Proc. of the 15th Network and Distributed System Security Symp. (NDSS). 2008.

[71] Plohmann D, Yakdan K, Klatt M, Bader J, Gerhards-Padilla E. A comprehensive measurement study of domain generating malware. In: Proc. of the 25th USENIX Conf. Security Symp. Austin: USENIX Association, 2016. 263–278.

[72] Nadji Y, Antonakakis M, Perdisci R, Lee W. Connected colors: Unveiling the structure of criminal networks. In: Proc. of the 16th Int'l Symp. on Research in Attacks, Intrusions, and Defenses. Rodney Bay: Springer, 2013. 390–410. [doi: 10.1007/978-3-642-41284-4_20]

[73] Alieyan K, Almomani A, Manasrah A, Kadhum MM. A survey of Botnet detection based on DNS. Neural Computing & Applications, 2017, 28(7): 1541–1558. [doi: 10.1007/s00521-015-2128-0]

[74] Singh M, Singh M, Kaur S. Issues and challenges in DNS based Botnet detection: A survey. Computers & Security, 2019, 86: 28–52.

[doi: 10.1016/j.cose.2019.05.019]

[75] Caglayan A, Toothaker M, Drapeau D, Burke D, Eaton G. Real-time detection of Fast-Flux service networks. In: Proc. of the 2009 Cybersecurity Applications & Technology Conf. for Homeland Security. Washington: IEEE, 2009. 285–292. [doi: 10.1109/CATCH.2009.44]

[76] Antonakakis M, Perdisci R, Dagon D, Lee W, Feamster N. Building a dynamic reputation system for DNS. In: Proc. of the 19th USENIX Conf. on Security. Washington: USENIX Association, 2010. 273–290.

[77] Antonakakis M, Perdisci R, Lee W, Vasiloglou N, Dagon D. Detecting malware domains at the upper DNS hierarchy. In: Proc. of the 20th USENIX Conf. on Security. San Francisco: USENIX Association, 2011. 1–16.

[78] Bilge L, Kirda E, Kruegel C, Balduzzi M. EXPOSURE: Finding malicious domains using passive DNS analysis. In: Proc. of the 18th Network and Distributed System Security Symp. San Diego, 2011.

[79] Bilge L, Sen S, Balzarotti D, Kirda E, Kruegel C. Exposure: A passive DNS analysis service to detect and report malicious domains. ACM Trans. on Information and System Security, 2014, 16(4): 14. [doi: 10.1145/2584679]

[80] Perdisci R, Corona I, Giacinto G. Early detection of malicious flux networks via large-scale passive DNS traffic analysis. IEEE Trans. on Dependable and Secure Computing, 2012, 9(5): 714–726. [doi: 10.1109/TDSC.2012.35]

[81] Perdisci R, Corona I, Dagon D, Lee W. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In: Proc. of the 2009 Annual Computer Security Applications Conf. Honolulu: IEEE, 2009. 311–320.

[82] Nelms T, Perdisci R, Ahamad M. ExecScent: Mining for new C&C domains in live networks with adaptive control protocol templates. In: Proc. of the 22nd USENIX Conf. on Security. Washington: USENIX Association, 2013. 589–604.

[83] Dolberg L, François J, Engel T. Multi-dimensional aggregation for DNS monitoring. In: Proc. of the 38th Annual IEEE Conf. on Local Computer Networks. Sydney: IEEE, 2013. 390–398. [doi: 10.1109/LCN.2013.6761271]

[84] Dolberg L, François J, Engel T. Efficient multidimensional aggregation for large scale monitoring. In: Proc. of the 26th Int'l Conf. on Large Installation System Administration: Strategies, Tools, and Techniques. San Diego: USENIX Association, 2012. 163–180.

[85] Rahbarinia B, Perdisci R, Antonakakis M. Segugio: Efficient behavior-based tracking of malware-control domains in large ISP networks. In: Proc. of the 45th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks. Rio de Janeiro: IEEE, 2015. 403–414. [doi: 10.1109/DSN.2015.35]

[86] Rahbarinia B, Perdisci R, Antonakakis M. Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks. ACM Trans. on Privacy and Security, 2016, 19(2): 4. [doi: 10.1145/2960409]

[87] Jiang N, Cao J, Jin Y, Li LE, Zhang ZL. Identifying suspicious activities through DNS failure graph analysis. In: Proc. of the 18th IEEE Int'l Conf. on Network Protocols. Kyoto: IEEE, 2010. 144–153. [doi: 10.1109/ICNP.2010.5762763]

[88] Berger A, D'Alconzo A, Gansterer WN, Pescapé A. Mining agile DNS traffic using graph analysis for cybercrime detection. Computer Networks, 2016, 100: 28–44. [doi: 10.1016/j.comnet.2016.02.009]

[89] Choi H, Lee H, Kim H. BotGAD: Detecting Botnets by capturing group activities in network traffic. In: Proc. of the 4th Int'l ICST Conf. on Communication System Software Middleware. Dublin: ACM, 2009. 1–8. [doi: 10.1145/1621890.1621893]

[90] Choi H, Lee H. Identifying Botnets by capturing group activities in DNS traffic. Computer Networks, 2012, 56(1): 20–33. [doi: 10.1016/j.comnet.2011.07.018]

[91] Singh M, Singh M, Kaur S. Detecting bot-infected machines using DNS fingerprinting. Digital Investigation, 2019, 28: 14–33. [doi: 10.1016/j.diin.2018.12.005]

[92] Kwon J, Lee J, Lee H, Perrig A. PsyBoG: A scalable Botnet detection method for large-scale DNS traffic. Computer Networks, 2016, 97: 48–73. [doi: 10.1016/j.comnet.2015.12.008]

[93] Lee J, Lee H. GMAD: Graph-based malware activity detection by DNS traffic analysis. Computer Communications, 2014, 49: 33–47. [doi: 10.1016/j.comcom.2014.04.013]

[94] Stevanovic M, Pedersen JM, D'Alconzo A, Ruehrup S, Berger A. On the ground truth problem of malicious DNS traffic analysis. Computers & Security, 2015, 55: 142–158. [doi: 10.1016/j.cose.2015.09.004]

[95] Antonakakis M, Perdisci R, Nadji Y, Vasiloglou N, Abu-Nimeh S, Lee W, Dagon D. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In: Proc. of the 21st USENIX Conf. on Security Symp. Bellevue: USENIX Association, 2012.

[96] Wang TS, Lin HT, Cheng WT, Chen CY. DBod: Clustering and detecting DGA-based Botnets using DNS traffic analysis. Computers & Security, 2017, 64: 1–15. [doi: 10.1016/j.cose.2016.10.001]

[97] Qiao YC, Zhang B, Zhang WZ, Sangaiah AK, Wu HL. DGA domain name classification method based on long short-term memory with attention mechanism. Applied Sciences, 2019, 9(20): 4205. [doi: 10.3390/app9204205]

[98] Prieto I, Magaña E, Morató D, Izal M. Botnet detection based on DNS records and active probing. In: Proc. of the 2011 Int'l Conf. on

Security and Cryptography. Seville: IEEE, 2011. 307–316.
[99] Security Information Exchange. 2022. https://www.farsightsecurity.com/solutions/security-information-exchange/
[100] Woodbridge J, Anderson HS, Ahuja A, Grant D. Predicting domain generation algorithms with long short-term memory networks. arXiv:1611.00791, 2016.
[101] Wang Y, Zhou AM, Liao S, Zheng RF, Hu R, Zhang L. A comprehensive survey on DNS tunnel detection. Computer Networks, 2021, 197: 108322. [doi: 10.1016/j.comnet.2021.108322]
[102] Merlo A, Papaleo G, Veneziano S, Aiello M. A comparative performance evaluation of DNS tunneling tools. In: Proc. of the 4th Int'l Conf. on Computational Intelligence in Security for Information Systems. Málaga: Springer, 2011. 84–91. [doi: 10.1007/978-3-642-21323-6_11]
[103] Mockapetris P. RFC 1034. Domain names—Concepts and facilities. 1987. https://tools.ietf.org/html/rfc1034
[104] Vixie P. RFC 6891. Extension mechanisms for DNS (EDNS(0)). 1999. https://tools.ietf.org/html/rfc6891
[105] NSTX. 2016. https://sourceforge.net/projects/nstx/
[106] Dnscat2. 2022. https://github.com/iagox86/dnscat2
[107] Iodine. 2023. https://code.kryo.sse/iodine/
[108] Nussbaum L, Neyron P, Richard O. On robust covert channels inside DNS. In: Proc. of the 24th IFIP Int'l Information Security Conf. on Emerging Challenges for Security, Privacy and Trust. Pafos: Springer, 2009. 51–62. [doi: 10.1007/978-3-642-01244-0_5]
[109] Dns2tcp. 2022. http://www.hsc.fr/ressources/outils/dns2tcp/index.html.en
[110] OzymanDNS. 2009. https://room362.com/post/2009/2009310ozymandns-tunneling-ssh-over-dns-html/
[111] Heyoka. 2022. http://heyoka.sourceforge.net/
[112] Tatang D, Quinkert F, Dolecki N, Holz T. A study of newly observed hostnames and DNS tunneling in the wild. arXiv:1902.08454, 2019.
[113] Tatang D, Quinkert F, Holz T. Below the radar: Spotting DNS tunnels in newly observed hostnames in the wild. In: Proc. of the 2019 APWG Symp. on Electronic Crime Research. Pittsburgh: IEEE, 2019. 1–15. [doi: 10.1109/eCrime47957.2019.9037595]
[114] Sheridan S, Keane A. Detection of DNS based covert channels. In: Proc. of the 14th European Conf. on Information Warfare and Security. 2015. 267–275.
[115] Al-Kasassbeh M, Khairallah T. Winning tactics with DNS tunnelling. Network Security, 2019, 2019(12): 12–19. [doi: 10.1016/S1353-4858(19)30144-8]
[116] Ghosh T, El-Sheikh E, Jammal W. A multi-stage detection technique for DNS-tunneled Botnets. In: Proc. of the 34th Int'l Conf. on Computers and their Applications, 2019, 58: 137–143.
[117] Sani AF, Setiawan MA. DNS tunneling detection using elasticsearch. IOP Conf. Series: Materials Science and Engineering, 2020, 722: 012064. [doi: 10.1088/1757-899X/722/1/012064]
[118] Ellens W, Żuraniewski P, Sperotto A, Schotanus H, Mandjes M, Meeuwissen E. Flow-based detection of DNS tunnels. In: Proc. of the 7th IFIP Int'l Conf. on Autonomous Infrastructure, Management, and Security. Barcelona: Springer, 2013. 124–135. [doi: 10.1007/978-3-642-38998-6_16]
[119] Himbeault M. A novel approach to detecting covert DNS tunnels using throughput estimation. 2014. http://hdl.handle.net/1993/23550
[120] Paxson V, Christodorescu M, Javed M, Rao J, Sailer R, Schales D, Stoecklin MP, Thomas K, Venema W, Weaver N. Practical comprehensive bounds on surreptitious communication over DNS. In: Proc. of the 22nd USENIX Conf. on Security. Washington: USENIX Association, 2013. 17–32.
[121] Cambiaso E, Aiello M, Mongelli M, Papaleo G. Feature transformation and mutual information for DNS tunneling analysis. In: Proc. of the 8th Int'l Conf. on Ubiquitous Future Networks. Vienna: IEEE, 2016. 957–959. [doi: 10.1109/ICUFN.2016.7536939]
[122] Nadler A, Aminov A, Shabtai A. Detection of malicious and low throughput data exfiltration over the DNS protocol. Computers & Security, 2019, 80: 36–53. [doi: 10.1016/j.cose.2018.09.006]
[123] Ahmed J, Gharakheili HH, Raza Q, Russell C, Sivaraman V. Real-time detection of DNS exfiltration and tunneling from enterprise networks. In: Proc. of the 2019 IFIP/IEEE Symp. on Integrated Network and Service Management. Arlington: IEEE, 2019. 649–653.
[124] Luo YQ, Liu SL, Meng Y, Wu DY. DNS tunnel Trojan detection method based on communication behavior analysis. Journal of Zhejiang University (Engineering Science), 2017, 51(9): 1780–1787 (in Chinese with English abstract). [doi: 10.3785/j.issn.1008-973X.2017.09.012]
[125] Das A, Shen MY, Shashanka M, Wang JS. Detection of exfiltration and tunneling over DNS. In: Proc. of the 16th IEEE Int'l Conf. on Machine Learning and Applications. Cancun: IEEE, 2017. 737–742. [doi: 10.1109/ICMLA.2017.00-71]
[126] Liu JK, Li SH, Zhang YZ, , Xiao J, Chang P, Peng CW. Detecting DNS tunnel through binary-classification based on behaviour feature.

In: Proc. of the 2017 IEEE Trustcom/BigDataSE/ICESS. Sydney: IEEE, 2017. 339–346. [doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.256]

[127] Do VT, Engelstad P, Feng BN, Van Do T. Detection of DNS tunneling in mobile networks using machine learning. In: Proc. of the 2017 Information Science and Applications. Singapore: Springer, 2017. 221–230. [doi: 10.1007/978-981-10-4154-9_26]

[128] Wu KM, Zhang YZ, Yin T. CLR: A classification of DNS tunnel based on logistic regression. In: Proc. of the 38th IEEE Int'l Performance Computing and Communications Conf. London: IEEE, 2019. [doi: 10.1109/IPCCC47392.2019.8958731]

[129] Lai CM, Huang BC, Huang SY, Mao CH, Lee HM. Detection of DNS tunneling by feature-free mechanism. In: Proc. of the 2018 IEEE Conf. on Dependable and Secure Computing. Kaohsiung: IEEE, 2018. 1–2. [doi: 10.1109/DESEC.2018.8625166]

[130] Liu C, Dai L, Cui WJ, Lin T. A byte-level CNN method to detect DNS tunnels. In: Proc. of the 38th IEEE IEEE Int'l Performance Computing and Communications Conf. London: IEEE, 2019. 1–8. [doi: 10.1109/IPCCC47392.2019.8958714]

[131] Pappas V, Fältström P, Massey D, Zhang LX. Distributed DNS troubleshooting. In: Proc. of the 2004 ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality. Portland: ACM, 2004. 265–270. [doi: 10.1145/1016687.1016694]

[132] Pappas V, Wessels D, Massey D, Lu SW, Terzis A, Zhang LX. Impact of configuration errors on DNS robustness. IEEE Journal on Selected Areas in Communications, 2009, 27(3): 275–290. [doi: 10.1109/JSAC.2009.090404]

[133] Kakarla SKR, Beckett R, Arzani B, Millstein T, Varghese G. GRooT: Proactive verification of dns configurations. In: Proc. of the 2020 Annual Conf. of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. Virtual Event: Association for Computing Machinery, 2020. 310–328. [doi: 10.1145/3387514.3405871]

[134] DNSChecker. 2022. http://www.squish.net/dnscheck/

[135] Eduard K. DNS servers crash due to bind security flaw. 2018. https://www.securityweek.com/dns-servers-crash-due-bind-security-flaw

[136] Rashid FY. ISC updates critical DoS bug in bind DNS software. 2016. https://www.infoworld.com/article/3126472/isc-updates-critical-DoS-bug-in-bind-dns-software.html

[137] Tung L. Azure global outage: Our DNS update mangled domain records, says Microsoft 2019. 2019. https://www.zdnet.com/article/azure-global-outage-our-dns-update-mangled-domain-records-says-microsoft/

[138] Siva KRK, Ryan B, Todd M, George V. SCALE: Automatically finding RFC compliance bugs in DNS nameservers. In: Proc. of the 19th USENIX Symp. on Networked Systems Design and Implementation. Renton: USENIX Association, 2022. 307–323.

[139] Chung T, Van Rijswijk-Deij R, Chandrasekaran B, Choffnes D, Levin D, Maggs BM, Mislove A, Wilson C. A longitudinal, end-to-end view of the DNSSEC ecosystem. In: Proc. of the 26th USENIX Conf. on Security Symp. Vancouver: USENIX Association, 2017. 1307–1322.

[140] Herzberg A, Shulman H. Fragmentation considered poisonous, or: One-domain-to-rule-them-all.org. In: Proc. of the 2013 IEEE Conf. on Communications and Network Security. National Harbor: IEEE, 2013. 224–232. [doi: 10.1109/CNS.2013.6682711]

[141] Dempsky M. DNSCurve: Link-level security for the domain name system. RFC draft-dempsky-dnscurve-01, 2010. https://datatracker.ietf.org/doc/id/draft-dempsky-dnscurve-01.html

[142] CISCO. OpenDNS. 2022. http://www.opendns.com/

[143] NLnet Labs. Unbound. 2019. https://www.nlnetlabs.nl/projects/unbound/about/

[144] DNS privacy daemon—Stubby. 2022. https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+Daemon+-+Stubby

[145] Patrick M. Improving DNS privacy in Firefox—Firefox nightly news. 2018. https://blog.nightly.mozilla.org/2018/06/01/improving-dns-privacy-in-firefox/

[146] Cloudflare resolver. 2022. https://cloudflare-dns.com/

[147] Google public DNS. 2022. https://developers.google.com/speed/public-dns/

[148] Quad9 DNS: Internet security & privacy in a few easy steps. 2022. https://www.quad9.net/

[149] Yandex. DNS. 2022. https://dns.yandex.com/

[150] OpenNIC project. 2022. https://www.opennic.org/

[151] Lu CY, Liu BJ, Li Z, Hao S, Duan HX, Zhang MM, Leng CY, Liu Y, Zhang ZF, Wu JP. An end-to-end, large-scale measurement of DNS-over-encryption: How far have we come? In: Proc. of the 2019 Internet Measurement Conf. Amsterdam: ACM, 2019. 22–35. [doi: 10.1145/3355369.3355580]

[152] Abley J, Lindqvist K. Operation of Anycast services. RFC 4786. 2006. https://tools.ietf.org/html/rfc4786

[153] Moura GCM, de O. Schmidt R, Heidemann J, de Vries WB, Müller M, Wei L, Hesselman C. Anycast vs. DDoS: Evaluating the November 2015 root DNS event. In: Proc. of the 2016 Internet Measurement Conf. Santa Monica: ACM, 2016. 255–270. [doi: 10.1145/

2987443.2987446]
[154] Yu YD, Wessels D, Larson M, Zhang LX. Authority server selection in DNS caching resolvers. ACM SIGCOMM Computer Communication Review, 2012, 42(2): 80–86. [doi: 10.1145/2185376.2185387]
[155] Bush R, Karrenberg D, Kosters M, Plzak R. Root name server operational requirements. RFC 2870. 2000. https://tools.ietf.org/html/rfc2870
[156] cz.nic. Knot DNS: Benchmark. 2019. https://www.knot-dns.cz/benchmark-old/
[157] ISC. BIND 9 performance history. 2017. https://www.isc.org/blogs/bind9-performance-history/
[158] cz.nic. DNS shotgun. 2019. https://gitlab.labs.nic.cz/knot/shotgun/
[159] ISC. Using the response rate limiting feature. 2018. https://kb.isc.org/docs/aa-00994
[160] Abley J, Gudmundsson O, Majkowski M, Hunt E. Providing minimal-sized responses to DNS queries that have QTYPE=ANY. RFC 8482, 2019. https://tools.ietf.org/html/rfc8482
[161] Bortzmeyer S. DNS query name minimisation to improve privacy. RFC 7816. 2016. www.rfc-editor.org/rfc/rfc7816.txt
[162] Zhu L, Hu Z, Heidemann J, Wessels D, Mankin A, Somaiya N. T-DNS: Connection-oriented DNS to improve privacy and security (poster abstract). In: Proc. of the 2014 ACM Conf. on SIGCOMM. Chicago: ACM, 2014. 379–380. [doi: 10.1145/2619239.2631442]
[163] Park K, Pai VS, Peterson L, Wang Z. CoDNS: Improving DNS performance and reliability via cooperative lookups. In: Proc. of the 6th Conf. on Symp. on Operating Systems Design and Implementation. San Francisco: USENIX Association, 2004.
[164] Khurshid A, Kiyak F, Caesar M. Improving robustness of DNS to software vulnerabilities. In: Proc. of the 27th Annual Computer Security Applications Conf. Orlando: ACM, 2011. 177–186. [doi: 10.1145/2076732.2076758]
[165] Asoni DE, Hitz S, Perrig A. A paged domain name system for query privacy. In: Proc. of the 16th Int'l Conf. on Cryptology and Network Security, Hong Kong, China: Springer, 2017. 250–273. [doi: 10.1007/978-3-030-02641-7_12]
[166] Schmitt P, Edmundson A, Mankin A, Feamster N. Oblivious DNS: Practical privacy for DNS queries. Proc. on Privacy Enhancing Technologies, 2019, 2019(2): 228–244. [doi: 10.2478/popets-2019-0028]
[167] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego: ACM, 2001. 149–160. [doi: 10.1145/383059.383071]
[168] Cox R, Muthitacharoen A, Morris RT. Serving DNS using a peer-to-peer lookup service. In: Proc. of the 1st Int'l Workshop on Peer-to-peer Systems. Cambridge: Springer, 2002. 155–165. [doi: 10.1007/3-540-45748-8_15]
[169] Danielis P, Altmann V, Skodzik J, Wegner T, Koerner A, Timmermann D. P-DONAS: A P2P-based domain name system in access networks. ACM Trans. on Internet Technology, 2015, 15(3): 11. [doi: 10.1145/2808229]
[170] Ramasubramanian V, Sirer EG. The design and implementation of a next generation name service for the internet. In: Proc. of the 2004 ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. Portland: ACM, 2004. 331–342. [doi: 10.1145/1015467.1015504]
[171] Song YT, Koyanagi K. Study on a hybrid P2P based DNS. In: Proc. of the 2011 IEEE Int'l Conf. on Computer Science and Automation Engineering. Shanghai: IEEE, 2011. 152–155. [doi: 10.1109/CSAE.2011.5952823]
[172] Namecoin. 2022. https://Namecoin.info
[173] Ali M, Nelson J, Shea R, Freedman MJ. Blockstack: A Global naming and storage system secured by blockchains. In: Proc. of the 2016 USENIX Conf. on Usenix Annual Technical Conf. . Denver: USENIX Association, 2016. 181–194.
[174] Li ZC, Gao S, Peng Z, Guo ST, Yang YY, Xiao B. B-DNS: A secure and efficient DNS based on the blockchain technology. IEEE Trans. on Network Science and Engineering, 2021, 8(2): 1674–1686. [doi: 10.1109/TNSE.2021.3068788]
[175] ENS. 2022. https://ens.domains/
[176] EMCDNS. 2022. https://emercoin.com/
[177] Casino F, Lykousas N, Katos V, Patsakis C. Unearthing malicious campaigns and actors from the blockchain DNS ecosystem. Computer Communications, 2021, 179: 217–230. [doi: 10.1016/j.comcom.2021.08.023]
[178] Zhang Y, Xia ZD, Fang BX, Zhang HL. An autonomous open root resolution architecture for domain name system in the Internet. Journal of Cyber Security, 2017, 2(4): 57–69 (in Chinese with English abstract). [doi: 10.19363/j.cnki.cn10-1380/tn.2017.10.005]
[179] He GB, Su W, Gao S, Yue JR. TD-Root: A trustworthy decentralized DNS root management architecture based on permissioned blockchain. Future Generation Computer Systems, 2020, 102: 912–924. [doi: 10.1016/j.future.2019.09.037]
[180] Zhen LF, Wu ZQ, Ma K. Domestic research progress of Internet real source address verification. Computer Systems and Applications, 2022, 31(4): 14–32 (in Chinese with English abstract). [doi: 10.15888/j.cnki.csa.008419]
[181] Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback. In: Proc. of the 2000 Conf. on Applications,

Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). Stockholm: ACM, 2000. 295–306. [doi: 10.1145/347059.347560]

[182] DNSBL. Spam Database Lookup. 2010. http://www.dnsbl.info/

[183] DNS Response Policy Zones. 2022. https://dnsrpz.info/

[184] Zhang ZY, Yu YD, Zhang HT, Newberry E, Mastorakis S, Li YB, Afanasyev A, Zhang LX. An overview of security support in named data networking. IEEE Communications Magazine, 2018, 56(11): 62–68. [doi: 10.1109/MCOM.2018.1701147]

附中文参考文献:

[1] 王垚, 胡铭曾, 李斌, 闫伯儒. 域名系统安全研究综述. 通信学报, 2007, 28(9): 91–103. [doi: 10.3321/j.issn:1000-436x.2007.09.015]

[3] 刘文峰, 张宇, 张宏莉, 方滨兴. 域名系统测量研究综述. 软件学报, 2022, 33(1): 211–232. http://www.jos.org.cn/1000-9825/6218.htm [doi: 10.13328/j.cnki.jos.006218]

[4] 王文通, 胡宁, 刘波, 刘欣, 李树栋. DNS安全防护技术研究综述. 软件学报, 2020, 31(7): 2205–2220. http://www.jos.org.cn/1000-9825/6046.htm [doi: 10.13328/j.cnki.jos.006046]

[5] 胡宁, 邓文平, 姚苏. 互联网DNS安全研究现状与挑战. 网络与信息安全学报, 2017, 3(3): 13–21. [doi: 10.11959/j.issn.2096-109x.2017.00154]

[124] 罗友强, 刘胜利, 颜猛, 武东英. 基于通信行为分析的DNS隧道木马检测方法. 浙江大学学报(工学版), 2017, 51(9): 1780–1787. [doi: 10.3785/j.issn.1008-973X.2017.09.012]

[178] 张宇, 夏重达, 方滨兴, 张宏莉. 一个自主开放的互联网根域名解析体系. 信息安全学报, 2017, 2(4): 57–69. [doi: 10.19363/j.cnki.cn10-1380/tn.2017.10.005]

[180] 甄龙飞, 吴振强, 马克. 国内互联网真实源地址验证研究进展. 计算机系统应用, 2022, 31(4): 14–32. [doi: 10.15888/j.cnki.csa.008419]

张宾(1976－), 男, 博士, 高级工程师, 博士生导师, 主要研究领域为网络安全, 网络管理, 数据分析.

张伟哲(1976－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为信息安全, 系统结构.

张宇(1978－), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为互联网基础设施安全, 网络拓扑测量, 未来网络体系.