

递归侧DNSSEC错误配置检测能力大规模探测分析

刘林晖¹, 涂菲帆¹, 陈勇², 左鹏², 刘东杰¹, 张银炎¹, 耿光刚¹

(1. 暨南大学网络空间安全学院, 广东 广州 510632; 2. 中国互联网络信息中心, 北京 100079)

摘要: DNSSEC (domain name system security extensions) 是一种域名系统 (DNS, domain name system) 的安全扩展协议, 通过为DNS记录添加签名来增加DNS的安全性。域名递归服务器能否有效验证DNSSEC配置的正确性, 并且在配置错误时返回相应的错误类型, 对保障整个DNS的安全至关重要。为此, 基于RFC 8914标准, 选择了8种在权威侧可配置的DNSSEC错误类型, 并在8个不同的子域分别配置了相应的DNSSEC错误。接下来, 面向全球范围内的公共DNS服务器, 筛选出其中支持DNSSEC的递归服务器作为探测对象, 针对上述8个子域发起解析请求, 对探测结果进行了收集、分析及可视化。探测结果表明, 对于部分错误, 多数支持DNSSEC的递归服务器可以正确地检测出域名的DNSSEC错误配置, 并且返回相应的错误类型, 如signature_expired、signature_not_valid、RRSIG_missing、DNSKEY_missing等错误。对当前全球范围内重要递归服务器检测DNSSEC错误配置的能力开展了大规模测量分析, 可有效指导未来DNSSEC广泛部署中递归侧的能力建设。

关键词: 域名系统; 域名系统安全扩展; 错误配置检测; 递归服务器

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2024068

Large-scale measurement and analysis on misconfigurations of DNSSEC from recursive side

LIU Linhui¹, TU Feifan¹, CHEN Yong², ZUO Peng², LIU Dongjie¹,
ZHANG Yinyan¹, GENG Guanggang¹

1. Institute of Cyberspace Security, Jinan University, Guangzhou 510632, China

2. China Internet Network Information Center, Beijing 100079, China

Abstract: Domain name system security extensions (DNSSEC) was a secure extension protocol for the domain name system (DNS), which enhanced DNS security by adding signatures to DNS records. It was very important to ensure the security of the entire DNS that the domain name recursive server could effectively verify the correctness of the DNSSEC configuration and return the corresponding error type when the configuration was wrong. For this

收稿日期: 2024-03-28; 修回日期: 2024-08-30

通信作者: 耿光刚, gggeng@jnu.edu.cn

基金项目: 国家重点研究计划 (2022YFB3103000)

Foundation Item: The National Key R&D Project of China (2022YFB3103000)

引用格式: 刘林晖, 涂菲帆, 陈勇, 等. 递归侧DNSSEC错误配置检测能力大规模探测分析[J]. 网络与信息安全学报, 2024, 10(5): 81-94.

Citation Format: LIU L H, TU F F, CHEN Y, et al. Large-scale measurement and analysis on misconfigurations of DNSSEC from recursive side[J]. Chinese Journal of Network and Information Security, 2024, 10(5): 81-94.

purpose, building upon the RFC 8914 standard, eight configurable error types were selected and corresponding DNSSEC errors were configured in eight different subdomains. Next, the recursive server supporting DNSSEC was selected as the probe object for the global public DNS server, and the resolution requests were launched for the aforementioned eight subdomains, and the probe results were collected, analyzed, and visualized. Experiments showed that most recursive servers that supported DNSSEC could correctly detect the DNSSEC misconfiguration of domain names and return the corresponding error type for some errors such as signature_expired, signature_not_valid, RRSIG_missing, DNSKEY_missing, and so on. This large-scale detection and analysis provided valuable insights into the capabilities of important recursive servers worldwide in validating DNSSEC configurations, guiding future efforts in enhancing DNSSEC deployment on the recursive side.

Keywords: DNS, DNSSEC, misconfiguration detection, recursive server

0 引言

域名系统 (DNS, domain name system) 是互联网的基本组成部分之一, 将用户友好的域名映射到数字 IP 地址。然而, DNS 长期以来一直存在安全问题, 如 DNS 欺骗和缓存中毒^[1]。为了应对 DNS 固有的安全缺陷, 研究人员开发了一种创新的安全增强方案, 即域名系统安全扩展 (DNSSEC, domain name system security extensions), 通过提供源认证、完整性认证和不存在认证, 以防止存储在 DNS 中的数据被非法篡改。此外, DNSSEC 还部署了加密密钥, 以保证用户收到真实的回复^[2]。

近年来, DNSSEC 的覆盖率正在逐步提高。到 2017 年年初, 已有超过 90% 的顶级域名和超过 47% 的国家/地区顶级域名支持 DNSSEC^[3], 主要的权威 DNS 服务器软件和库都已支持 DNSSEC。一些 Web 调试工具, 如 DNSSEC Debugger 和 DNSViz 都可以帮助管理者验证 DNSSEC 的部署情况。

然而, 由于 DNSSEC 的复杂性, 只有从根到叶子的每个签署人以及验证签名的解析器都履行自己的职责时, DNSSEC 才能正常工作, 其中任一环节的错误配置都可能对 DNSSEC 的安全性及性能产生影响。例如, 在服务器端, 签名过期或签名链的损坏都可能对域的完整性造成破坏; 在客户端, 可能出现有缺陷的 DNS 解析器无法捕获无效或缺失的签名的情况^[4]。DNSSEC 错误配置导致服务不可达的事情经常发生, 这不仅会影响用户的使用体验, 还会给运营商带来严重的经济损失。因此, 当发生 DNSSEC 配置错误时, 递归服务器能否对其进行检测并正确判断出是哪一种配置错误类型变得十分重要。

为了解决上述问题, 本文提出一种递归侧 DNSSEC 错误配置检测能力大规模探测分析方法。

1 背景

1.1 DNS

DNS 服务是互联网的一项核心服务, 负责将字母域名映射到数字 IP^[5]。DNS 信息保存在分布式的 DNS 服务器中, 方便用户根据需要进行查询^[6]。为了支持 DNS 高效地请求从域名到 IP 地址的映射, DNS 采用了树状层次结构, 当客户端连接到本地 DNS 递归服务器后, 服务器将会遍历树以接收请求的信息。普遍而言, DNS 的遍历树有 3 层, 即根 (root) 层、顶级域层和权威层。

每台 DNS 递归服务器都有一个包括所有根服务器主机名和 IP 地址的列表, 这台根服务器本身不进行任何域名映射, 而是返回一个包含下一层域名服务器及其 IP 的 NS 记录, 指向负责请求的顶级域服务器。最常见的顶级域名是通用顶级域名, 如 .com、.net、.edu、.org 以及国家代码的顶级域名。这些顶级域服务器再次指向下一层次, 该层次通常是该域名的权威服务器, 并返回请求的映射。图 1 展示了域名在 DNS 分布树的位置关系。

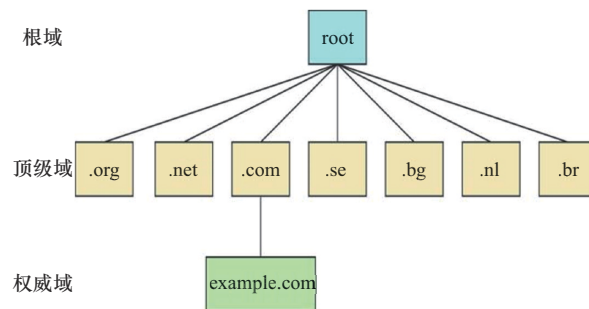


图 1 DNS 分布树

Figure 1 DNS distribution tree

将域名转换为对应 IP 地址的过程即 DNS 解析^[7], DNS 的解析过程如图 2 所示, 展现了客户端如何通过 DNS 解析获取 Web 服务器的地址。

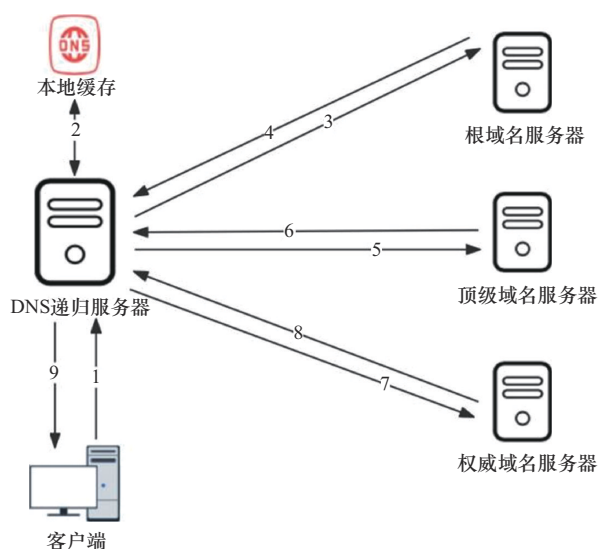


图2 DNS 的解析过程
Figure 2 DNS resolution process

然而, 最初 DNS 的设计采用了明文发送数据包的形式, 导致其容易受到被动攻击或主动攻击, 这种非加密的设计使用户面临隐私泄露的威胁^[8]。通过 DNS 流量, 可以对客户机进行指纹识别和分析用户行为^[9], 即使是简单地分析被动 DNS 数据, 也可以跟踪客户机在互联网上的活动^[10]。此外, 还可以伪造 DNS 数据, 将其转换为恶意的 IP 地址, 诱导用户连接不安全的站点。综上所述, 未受保护的 DNS 流量可能会给互联网用户带来重大的隐私风险。

1.2 DNSSEC

基于上述 DNS 可能遇到的安全隐患, 研究人员开发出新的安全协议以减轻 DNS 的隐私问题。在 DNS 协议中添加机密性和身份验证属性是解决 DNS 隐私威胁的有效方法, DNSSEC 正是一种可以为 DNS 提供身份验证的安全协议。

DNSSEC 采用公钥加密技术, 允许域名服务器对记录进行签名。在 DNSSEC 中, 权威域名服务器与公钥和私钥相关联。这些域名服务器使用私钥对 DNS 记录 (如 A 记录、MX 记录和 SOA 记录) 进行签名, 并将生成的签名存储在 RRSIG (resource record signature) 记录中。使用域名服

务器的公钥可以验证这些 RRSIG 记录的真实性, 该公钥存储在另一个称为 DNSKEY 的记录中。然而, DNSKEY 记录本身的可信性并不能得到保证。

为建立信任, DNSKEY 记录使用权威域名服务器持有的特殊密钥签名密钥进行签名, 这个密钥签名密钥的哈希值存储在域名服务器父域的 DS (delegation signer) 记录中。因此, 信任的负担转移到了父域中的域名服务器上 (例如, example.com 的父域是 .com)。父域使用其私有区域签名密钥对 DS 记录进行签名, 相应的公共区域签名密钥存储在 DNSKEY 记录中, 然后由密钥签名密钥对其进行签名。该密钥签名密钥的 DS 记录存储在 .com 的父域中, 对应于由点 (.) 表示的根域。根域的 DNSKEY 在 DNS 解析器工具中是硬编码的, 并且代表了信任链中的“信任锚点”^[11]。信任链以所需的记录 (在本例中为 example.com 的 A 记录) 结束。因此, DNSSEC 中的身份验证过程涉及对记录进行递归验证, 一直到根域^[12]。

为了实现上述提到的加密技术, DNSSEC 在 DNS 中增加了 4 种记录类型: RRSIG、DNSKEY、DS 和 NSEC (next secure)。这些扩展的记录, 使得 DNS 开销变大, 并且增加了处理时间和数据包大小^[6]。由于 DNSSEC 显著增加了现有 DNS 基础设施的复杂性, 因此错误配置的可能性会增加^[3], 而 DNSSEC 错误配置容易给用户和运营商带来如服务不可用、客户端验证失败、安全性损害、DNS 信息篡改等危害。如果 DNSSEC 验证过程中出现问题, 一些 DNS 查询可能会失败或被拒绝, 这会导致网站无法访问、电子邮件传递延迟或失败, 以及其他基于 DNS 的服务中断。此外, 攻击者会利用错误配置的 DNSSEC 来获取敏感信息, 如域名密钥或其他验证凭据^[13]。基于以上 DNSSEC 错误配置可能产生的风险, DNS 递归服务器具有及时检测 DNSSEC 是否存在错误配置以及判断属于哪种错误配置类型的能力就显得至关重要。

2 相关工作

由于 DNSSEC 的配置部署情况比较复杂, 目前已有许多学者对其相关问题进行研究。本

节将讨论关于 DNSSEC 错误配置的研究工作。在目前关于 DNSSEC 的论文中,大部分研究集中在 DNSSEC 的性能上,对于错误配置的关注则比较少^[14]。张曼等^[15]就 DNSSEC 验证服务的部署情况展开了调查,发现虽然约 91% 的顶级域已部署,但二级域部署率只占约 6%;此外,虽然推出了 NSEC3,但部分系统依旧部署先前的 NSEC,存在区文件被遍历的风险。Migault 等^[16]分析了 DNSSEC 解析和服务的可用性,但没有讨论 DNSSEC 配置的正确性。Yang 等^[17]调查了 DNS 解析侧的验证情况,发现大部分 DNS 解析器并没有对 DNSSEC 签名进行正确的验证。

Deccio 等^[18]对 2 000 个域进行了 5 个月的调查,发现 20% 的 RRSIG 至少过期一次的域,其 RRSIG 可能会过期更多次。Wang 等^[19]发现仅有大约 1% 的 .com、.net 和 .org 域名部署了 DNSSEC,在这些部署了 DNSSEC 的域名中,超过 30% 的域名因缺少 DS 记录而产生配置错误。许多部署了 DNSSEC 的客户端,仍会出现解析故障。Adrichem 等^[2]对一天的二级域名进行了分析,发现其中 4% 的域名出现了错误配置。Dai 等^[20]发现 Alexa 排名前 100 万的二级域名中 19.46% 具有来自根的无效信任链。Chuang 等^[4]则对全球近 6 万台 DNS 解析器进行了主动测量,以评估解析端验证情况,发现 DNSSEC 基础设施管理不善的情况普遍存在,其中 31% 的域名没有发布验证所需的所有记录,39% 的域名使用了强度较低的密钥签名密钥等。

以上的研究都只局限于对域名现有 DNSSEC 错误配置的检测,而并没有自己搭建权威服务器配置 DNSSEC 错误并检测错误,验证递归服务器的检错能力。为了弥补 DNSSEC 配置并检测研究的空缺,本文提出了一种对支持 DNSSEC 的递归解析服务器的错误配置检测能力的验证方法,分析递归解析服务器是否提供相应的错误提示,以判定递归服务器对 DNSSEC 配置错误是否有效验证。

3 实验方法

本文实验的整体思路为搭建支持 DNSSEC 的权威服务器并划分子域,在其子域分别设

置不同的错误配置,再使用支持 DNSSEC 的递归服务器来验证搭建的子域上的 DNSSEC 配置,从而探究递归服务器是否有发现 DNSSEC 错误配置的能力。首先,本文基于 RFC 8914 标准,全面梳理了 DNSSEC 错误配置的类型,再结合实际可部署情况最终选择了包括“un-supported_DNSKEY_algorithm”在内的 8 种错误类型。然后,本次实验在二级域 iwbtfy.top 搭建了 8 个三级子域,并在子域上配置不同的错误,同时设置了一个正确配置的子域作为对照。在数据集的选择上,本文收集了全球范围内的 DNS 服务器地址,筛选出了支持 DNSSEC 的递归服务器,并向筛选出来的 3 607 台支持 DNSSEC 的递归服务器发送对所搭建子域的请求,对返回的日志进行了全面的研究分析,以验证两个问题:①递归服务器是否能验证错误;②如果能验证错误,是否可以返回相应的错误。DNSSEC 错误检测整体框架如图 3 所示。

3.1 工具介绍

本文在编写筛选支持 DNSSEC 的递归服务器以及向这些服务器发送请求的自动化脚本时使用了 DIG (domain information groper) 工具,在搭建 DNS 服务器时使用了 BIND (berkeley internet name domain) 工具,下面对这两种工具进行简要介绍。

(1) DIG

DIG 属于 DNS 服务器软件 BIND 的一部分,是一个可以用来查询 DNS 服务器的命令行工具。DIG 根据查询回来的日志中的一些字段值判断服务器的相关属性。例如,日志中的 flag 字段,其部分字段值如表 1 所示。本文根据这个字段值对服务器进行判断筛选。

(2) BIND

BIND 是一款用于实现域名系统的开源软件,能够将域名解析为相应的 IP 地址以及执行其他 DNS 相关任务,广泛应用于互联网的 DNS 服务器。本文使用 BIND 9.11.4 搭建 DNS 解析服务器,该版本支持 eDNS,可以直接在返回的日志里显示 eCode,表示当前服务器检测出域名的 DNSSEC 的错误配置。

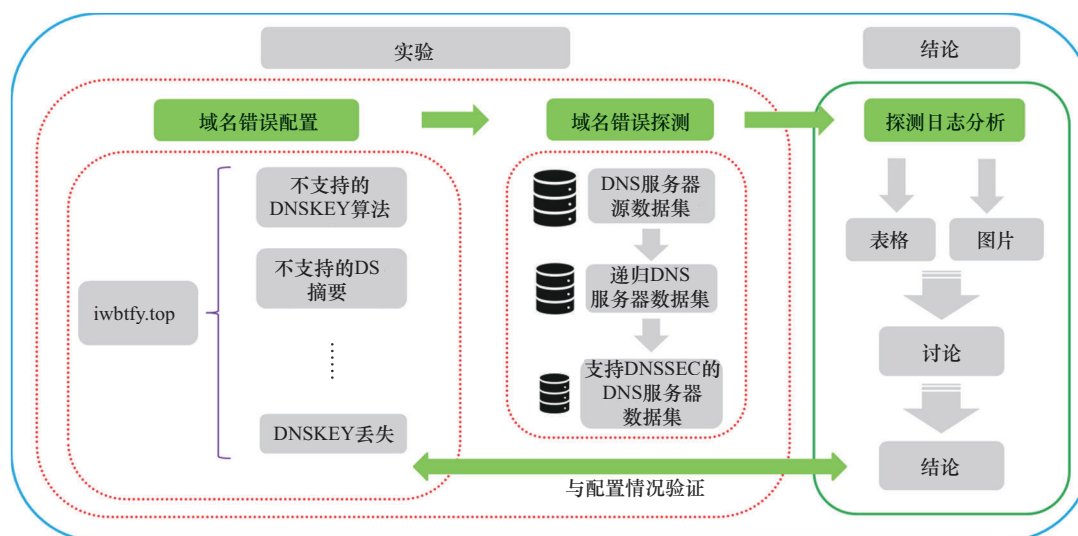


图3 DNSSEC错误检测整体框架

Figure 3 Overall framework of DNSSEC error detection

表1 flag的部分字段值

Table 1 Some field values of flag

flag 值	属性
QR=1	该消息是对查询的响应
RA=1	服务器为查询执行递归
AD=1	响应中的数据已使用DNSSEC进行身份验证

本文实验使用支持eDNS的DIG 9.16.13, 这个版本的DIG返回的查询日志可以直接显示eCode, eCode的部分字段值如表2所示。本文后续的验证基于日志中的eCode字段进行分析。

表2 eCode的部分字段值

Table 2 Some field values of eCode

错误类型	eCode
unsupported_DNSKEY_algorithm	1
unsupported_DS_digest_type	2
signature_expired	7
signature_not_yet_valid	8
DNSKEY_missing	9
RRSIG_missing	10
no_zone_key_bit_set	11
NSEC_missing	12

3.2 服务器数据集获取

本文从一个收集公共DNS服务器列表的网站下载了一个开源的DNS数据集, 这个数据集收集了全球的DNS服务器。然而, 这个数据

集不会实时更新, 可能会存在服务器已经不使用但是还在数据表中的情况。因此, 需要对这个数据集进行进一步筛选: ①该DNS服务器是否仍在提供DNS解析服务; ②该DNS服务器是否是递归服务器; ③该DNS服务器是否支持DNSSEC。

根据筛选条件, 本文利用DIG工具编写了Python脚本进行筛选, 从13 382台DNS服务器中筛选出4 869台递归服务器, 最后得到3 607台仍在提供服务的支持DNSSEC的递归服务器, 筛选结果如表3所示。

表3 服务器的筛选结果

Table 3 Filter results for servers

原始DNS服务器台数	递归服务器台数	支持DNSSEC的递归服务器台数
13 382	4 869	3 607

从筛选的结果可以看出, 目前的DNSSEC配置还未完全部署, 许多递归服务器还未支持DNSSEC验证。

3.3 域名环境搭建

在获得服务器数据后, 就可以着手搭建域名环境。首先, 注册域名iwbtty.top; 然后, 租用一台云服务器, 将解析该域名的DNS服务器地址指向这台云服务器; 最后, 在云服务器上使用BIND搭建DNS服务器, 配置域环境, 以及为域

配置 DNSSEC，提供 DNS 解析服务。为了方便配置，以及让后续的可视化等分析更加清晰，在此域名下搭建了 8 个三级子域，分别对应 8 种错误，同时设置了一个正常配置的子域作为对照，其错误配置类型为 Null，表示未配置错误。DNSSEC 错误配置及其对应子域分配如表 4 所示。

表 4 DNSSEC 错误配置及其对应子域分配

Table 4 DNSSEC misconfigurations and corresponding subdomain assignments

子域名	DNSSEC 错误配置类型
unsupportedDnskey.iwbtfy.top	unsupported_DNSKEY_algorithm
unsupportedDs.iwbtfy.top	unsupported_DS_digest_type
signatureExpired.iwbtfy.top	signature_expired
signatureNotValid.iwbtfy.top	signature_not_yet_valid
rrsigMissing.iwbtfy.top	RRSIG_missing
noZoneKey.iwbtfy.top	no_zone_key_bit_set
nsecMissing.iwbtfy.top	NSEC_missing
dnskeyMissing.iwbtfy.top	DNSKEY_missing
right.iwbtfy.top	Null

(1) 配置域及 DNSSEC

域名的 DNSSEC 配置流程如图 4 所示，主要的步骤如下。

- 1) 创建并且配置子域的域文件。
- 2) 生成 KSK 和 ZSK 文件。

3) 将生成的 KSK 和 ZSK 文件写入子域文件。

4) 使用 KSK 生成 DS 记录。

5) 将 DS 记录写入父域的域文件。

6) 重新签名父域的域文件。

7) 用 KSK 签名子域的域文件。

8) 在 named 的配置文件 named.conf 中添加这个子域。

由于每个子域的操作相似，因此本文实验编写了 shell 脚本进行批量化配置，运行脚本即可完成对该子域的相关配置。

(2) 配置相应的错误

1) 错误梳理：本文实验基于 RFC 8914 标准，选择了其中实际可配置的 8 种错误，如表 5 所示。

2) 错误配置：根据梳理的错误，在搭建的三级子域上针对性地配置错误，如表 6 所示。

3.4 检测

在筛选出全球范围内仍在提供服务的支持 DNSSEC 的递归服务器，并且搭建好存在 DNSSEC 配置错误的子域后，本文编写了批量化检测的 Python 脚本向这些递归服务器发送请求，获得返回日志，并且将探测的结果保存起来，使用 Python 脚本对其进行分析和可视化。

本文实验检测的 DNS 服务器是全球范围内的，网络环境非常复杂，在检测过程中，检测日志中出现了丢失数据包的情况。为了降低丢失数

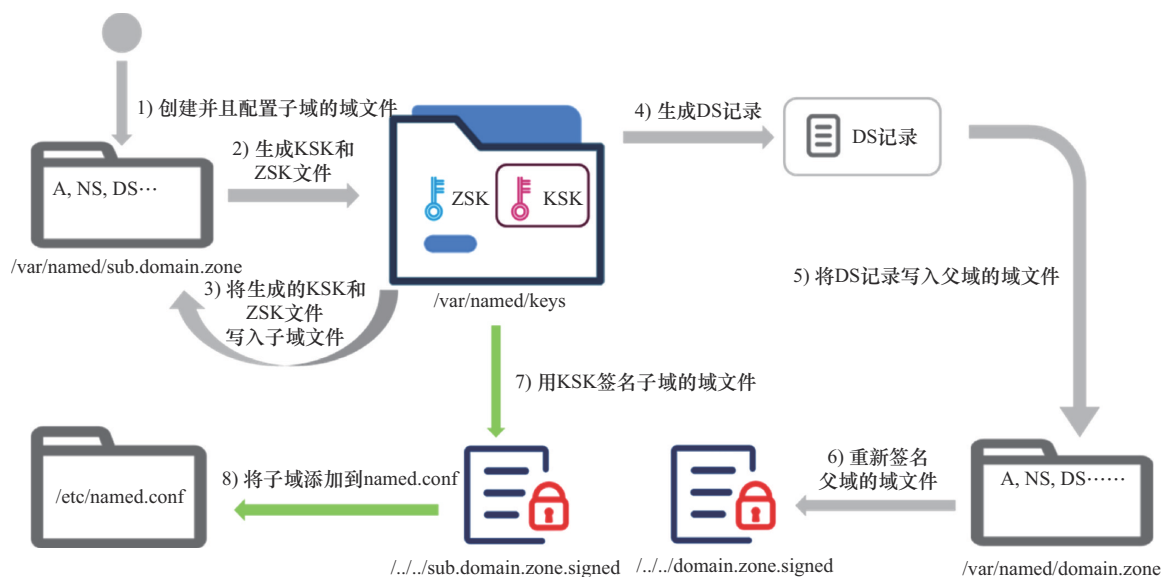


图 4 域名的 DNSSEC 配置流程

Figure 4 DNSSEC configuration process for domain names

表 5 DNSSEC 配置错误的梳理
Table 5 DNSSEC configuration error sorting

代号	DNSSEC 错误配置类型	说明
1	unsupported_DNSKEY_algorithm	DNSKEY RRset contained only unsupported DNSSEC algorithms
2	unsupported_DS_digest_type	a DS RRset contained only unsupported digest types
7	signature_expired	no signatures are presently valid and some (often all) are expired
8	signature_not_yet_valid	no signatures are presently valid and at least some are not yet valid
9	DNSKEY_missing	a DS record existed at a parent, but no supported matching DNSKEY record could be found for the child
10	RRSIG_missing	no RRSIGs could be found for at least one RRset where RRSIGs were expected
11	no_zone_key_bit_set	no zone key bit was set in a DNSKEY
12	NSEC_missing	the requested data was missing and a covering NSEC or NSEC3 was not provided

表 6 DNSSEC 的错误配置
Table 6 Misconfigurations of DNSSEC

代号	DNSSEC 错误配置类型	错误配置操作
1	unsupported_DNSKEY_algorithm	使用 SHA-1 算法对 DNSKEY 进行签名
2	unsupported_DS_digest_type	将 DS 记录的表示摘要类型的字段改成不支持的算法
7	signature_expired	将签名的结束时间改为现在时间之前
8	signature_not_yet_valid	将签名的起始时间改为现在时间之后
9	DNSKEY_missing	将 DNSKEY 注释掉
10	RRSIG_missing	将 RRSIG 签名注释掉
11	no_zone_key_bit_set	将 ZSK 注释掉
12	NSEC_missing	将 NSEC 注释掉

据包对实验造成的影响，提高实验的鲁棒性，本文采用了重复发送探测包的办法。如果出现丢失数据包的情况，就会向该服务器重新发送 DNS 请求。为了保证检测的效率，设定最大连接请求重发次数为 5。

4 实验结果

本文使用 3 607 台支持 DNSSEC 的递归服务器分别对 8 个存在错误配置的子域以及 1 个正常子域的 DNS 的 A 记录发送请求，将响应的日志记录到本地，并对获取到的结果进行统计分析。

4.1 服务器是否正确判错验证

本文实验首先针对递归服务器是否能验证错误进行了统计分析。服务器如果判断出域名 DNSSEC 配置存在错误，就不会解析域名，即不会返回 A 记录。与之相反，如果返回 A 记录，则代表服务器没有检测出域名存在 DNSSEC 错误配置情况。本文统

计了每个错误类型子域对应的日志中返回 A 记录的服务器数量。同时，为了说明不能解析确实是配置错误导致的，本文还统计了正常子域的 A 记录服务器返回情况作为对照，如表 7 所示。

表 7 正常解析的服务器数量
Table 7 Number of servers that can be resolved normally

DNSSEC 错误配置类型	服务器台数	占比
unsupported_DNSKEY_algorithm	3 279	91%
unsupported_DS_digest_type	81	2%
signature_expired	82	2%
signature_not_yet_valid	78	2%
DNSKEY_missing	87	2%
RRSIG_missing	74	2%
no_zone_key_bit_set	87	2%
NSEC_missing	74	2%
Null	3 292	91%

从表 7 中可以清晰地看到, 对于正常配置的子域, 91% 的服务器可以正常解析并返回 A 记录, 剩下 9% 的服务器是因网络问题而丢包的服务器, 但是对于 unsupported_DS_digest_type、signature_expired、signature_not_yet_valid、DNSKEY_missing、RRSIG_missing、no_zone_key_bit_set、NSEC_missing 这 7 种错误, 只有 2% 的服务器由于没有正确配置 DNSSEC 检测, 依然返回了 A 记录, 并且这 7 种错误的正常解析的数量相近, 都处于 74~87。根据日志信息进行验证, 证实了它们基本是同一批服务器, 它们对每一种错误都不具有检错能力。除去这些服务器, 剩下的 98% 的服务器都可以正确检测域名的 DNSSEC 配置错误, 并且不解析域名。有 2% 的递归服务器尽管在前期筛选中证明其支持 DNSSEC, 但在经过实验测试后, 发现这部分递归服务器并不具备验证 DNSSEC 错误配置的能力。而对于 unsupported_DNSKEY_algorithm 这一错误, 它属于一种特殊情况, 91% 的服务器没有检测出该错误, 在下文中将单独分析这一结果。根据以上实验结果可以推断, 大部分的递归服务器可以检测出域名的 DNSSEC 是否存在错误配置的情况。

4.2 验证服务器是否返回相应错误类型

本文针对递归服务器是否能验证错误并返回

相应错误类型的问题, 对日志进行了更加细致的统计分析。本文对 8 个错误配置类型以及 1 个正常域名的 DNS 服务器的检测情况进行了统计, 结果如表 8~表 16 所示。

返回结果字段解释及其原因解释。

1) no_error_code(false): 该服务器返回的结果能够判断错误, 但是没有返回具体的错误类型, 即没有 error_code 这个字段。这是因为 DIG 的版本不支持 eCode, 或者服务器本身不支持 eDNS。

2) no_error_code(true): 该服务器没有判断出错误, 所以没有返回 error_code。

对于上述两种情况如何进行区分, 本文读取了对应情况的全部日志。如果在该服务器返回的日志中存在 A 记录, 说明该服务器成功解析, 则属于服务器没有判断出错误, 那么这种情况就属于 no_error_code(true)。反之, 没有 A 记录则属于 no_error_code(false)。

3) DIG_command_failed: 对于该返回结果, 本文实验在代码中的判断条件为“是否收到响应包”。如果未收到响应包, 就会判断出现了丢包情况。而出现丢包情况有两种可能, 一种是网络环境不稳定, 在查询过程中数据包传输的时候出现丢包; 另一种是配置错误导致服务器内部发生丢包。对比分析表 8 与表 9~表 16, 发现在配置

表 8 正常子域检测结果
Table 8 Detection results of normal subdomain

配置错误	检测错误	服务器台数	占比
Null	no_error_code(true)	3 292	91%
	DIG_command_failed	307	8%
	DNSKEY_missing(9)	1	0%
	no_reachable_authority(22)	7	1%

表 9 unsupported_DNSKEY_algorithm 检测结果
Table 9 Detection results of unsupported_DNSKEY_algorithm

配置错误	检测错误	服务器台数	占比
unsupported_DNSKEY_algorithm(1)	no_error_code(true)	2 504	69%
	DIG_command_failed	278	8%
	NSEC_missing(12)	1	0%
	unsupported_DNSKEY(1)	820	23%
	no_reachable_authority(22)	4	0%

表 10 unsupported_DS_digest_type 检测结果

Table 10 Detection results of unsupported_DS_digest_type

配置错误	检测错误	服务器台数	占比
unsupported_DS_digest_type(2)	DNSSEC_bogus(6)	148	4%
	no_error_code(false)	1 971	54%
	cached_error(13)	61	2%
	DIG_command_failed	352	10%
	DNSKEY_missing(9)	852	24%
	unsupported_DNSKEY(1)	64	2%
	no_reachable_authority(22)	78	2%
	no_error_code(true)	81	2%

表 11 signature_expired 检测结果

Table 11 Detection results of signature_expired

配置错误	检测错误	服务器台数	占比
signature_expired(7)	DIG_command_failed	293	8%
	no_error_code(false)	1 202	33%
	signature_expired(7)	1 789	50%
	DNSSEC_bogus(6)	111	3%
	DNSKEY_missing(9)	65	2%
	cached_error(13)	65	2%
	no_error_code(true)	82	2%

表 12 signature_not_yet_valid 检测结果

Table 12 Detection results of signature_not_yet_valid

配置错误	检测错误	服务器台数	占比
signature_not_yet_valid(8)	DNSSEC_bogus(6)	67	2%
	no_reachable_authority(22)	1	0%
	DIG_command_failed	284	8%
	cached_error(13)	1	0%
	signature_not_valid(8)	2 005	56%
	no_error_code(false)	1 171	32%
	no_error_code(true)	78	2%

错误和未配置错误两种情况下，DIG_command_failed字段的服务器数量相近，从而可以判断配置错误并不会增加丢包情况，并进一步确定是网络环境复杂导致的丢包。

对比分析表 8 与表 9~16，根据正常子域除去丢包情况，基本是正常解析，说明 8 个配置错误的子域会返回错误，确实是 DNSSEC 配置错误导致的。

此外，通过分析结果日志统计表，发现 8 种

错误类型的探测结果存在共性与个性的地方。本文从共性和个性两个角度对这 8 种错误类型的探测结果进行了分析，并且将结果可视化。

(1) 共性方面

从实验结果来看，在对每一种错误的探测中，DIG_command_failed的服务器数量基本一致。如图 5 所示，在 8 次探测中返回 DIG_command_failed的服务器数量为 265~355。虽然本文实验在探测过程中使用了重复发包方式来减少

表 13 DNSKEY_missing 检测结果
Table 13 Detection results of DNSKEY_missing

配置错误	检测错误	服务器台数	占比
DNSKEY_missing(9)	DNSSEC_bogus(6)	106	3%
	DNSKEY_missing(9)	1 116	31%
	DIG_command_failed	312	9%
	no_reachable_authority(22)	11	0%
	no_error_code(false)	1 159	32%
	RRSIG_missing(10)	815	23%
	no_error_code(true)	87	2%

表 14 RRSIG_missing 检测结果
Table 14 Detection results of RRSIG_missing

配置错误	检测错误	服务器台数	占比
RRSIG_missing(10)	no_error_code(false)	1 189	33%
	not_authoritative(20)	13	0%
	DNSSEC_bogus(6)	65	2%
	invalid_data(24)	5	0%
	RRSIG_missing(10)	1,949	54%
	DIG_command_failed	296	8%
	cached_error(13)	4	0%
	not_ready(14)	5	0%
	no_error_code(true)	81	2%

表 15 no_zone_key_bit_set 检测结果
Table 15 Detection results of no_zone_key_bit_set

配置错误	检测错误	服务器台数	占比
no_zone_key_bit_set(11)	DNSKEY_missing(9)	10	0%
	DNSSEC_bogus(6)	988	27%
	no_reachable_authority(22)	46	1%
	no_error_code(false)	2 096	58%
	cached_error(13)	63	2%
	DIG_command_failed	317	9%
	no_error_code(true)	87	2%

连接失败的服务器数量，并为了保证探测效率而将最大重发次数设为 5，但是仍然存在一些服务器在 5 次重发后依旧出现丢包的情况。这一结果符合网络环境十分复杂的现实情况，也从侧面反映了探测实验的有效性和说服力。

(2) 个性方面

根据 8 种错误的探测日志结果，可以将 8 种错误大致分为 3 类。①大部分服务器可以正

确检测出域名的 DNSSEC 配置错误类型的 4 种错误：signature_expired、signature_not_valid、RRSIG_missing、DNSKEY_missing。②服务器可以检测出错误，但是不能检测出准确的 DNSSEC 配置错误类型的 3 种错误：NSEC_missing、no_zone_key、unsupported_DS。③还有一类特殊情况是 unsupported_DNSKEY，本文单独进行分析。

表 16 NSEC_missing检测结果
Table 16 NSEC_missing detection results

配置错误	检测错误	服务器台数	占比
NSEC_missing(12)	DNSKEY_missing(9)	3	0%
	DIG_command_failed	302	8%
	DNSSEC_bogus(6)	105	3%
	no_error_code(false)	1 183	33%
	cached_error(13)	1	0%
	no_reachable_authority(22)	2	0%
	RRSIG_missing(10)	1 937	54%
	no_error_code(true)	74	2%

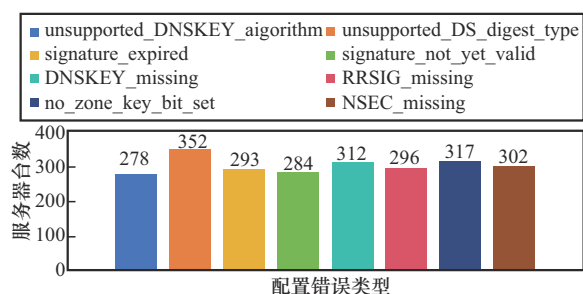


图5 DIG_command_failed分布(丢包)

Figure 5 DIG_command_failed distribution(packet loss)

对于第一类情况，将实验结果数据可视化得到如图6~图9所示的分布图。

如图6所示，1 789台（约占全部服务器的50%）服务器认为存在signature_expired错误，结果与预设的一致。其余服务器连接失败或返回其他错误。

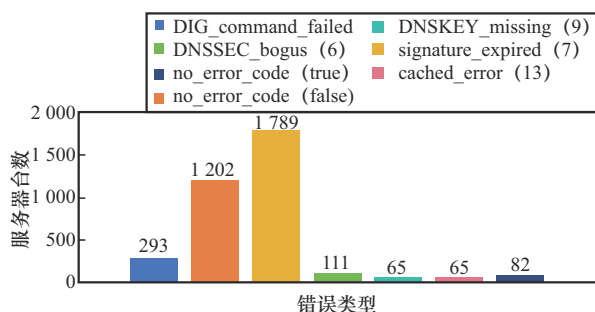


图6 signature_expired探测错误结果分布

Figure 6 Distribution of signature_expired detection error results

如图7所示，返回RRSIG_missing错误的服务器数量最多（1 949台，约占全部服务器的54%），结果与预设的一致。其余服务器连接失败或者返回其他错误。

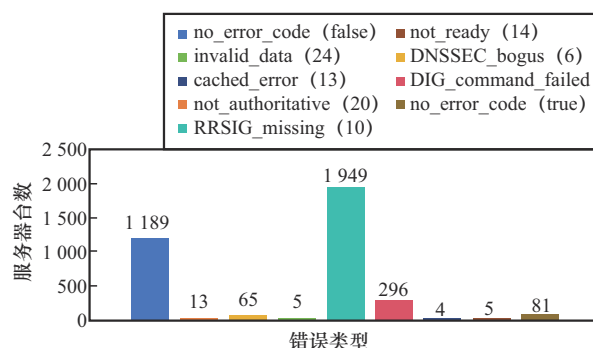


图7 RRSIG_missing探测错误结果分布

Figure 7 Distribution of RRSIG_missing detection error results

如图8所示，返回DNSKEY_missing错误的服务器最多（1 116台，约占全部服务器的31%），结果与预设的一致。815台服务器将错误判断为RRSIG_missing（约占全部服务器的23%）。

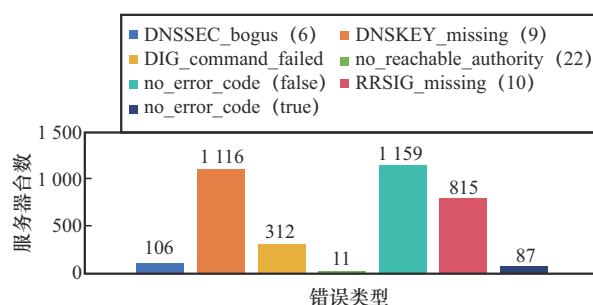


图8 DNSKEY_missing探测错误结果分布

Figure 8 Distribution of DNSKEY_missing detection error results

如图9所示，返回signature_not_valid错误的服务器最多（2 005台，约占全部服务器的56%），结果与预设的一致。其余40%左右的服务器连接失败或者不能返回eCode。

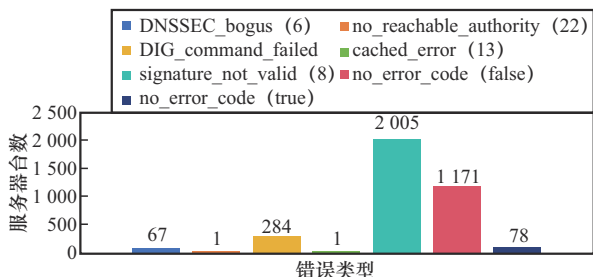


图9 signature_not_valid探测错误结果分布

Figure 9 Distribution of signature_not_valid detection error results

从实验结果来看,这4种错误在可以返回错误类型的服务器中,几乎全部的服务器可以正确返回预设的错误。而对于 NSEC_missing、no_zone_key、unsupported_DS 这3种错误,检测结果并没有正确地返回预设的错误,可视化结果如图10~图12所示。

如图10所示,判断出有错误的服务器最多,但是不能返回具体错误类型的服务器有2096台,约占全部服务器的58%,返回的其他错误类型也不是预设的错误,说明许多服务器现在可以判断DNSSEC配置出错,但是不能正确返回预设的错误类型。

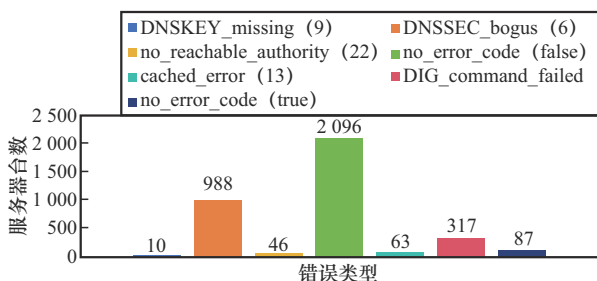


图10 no_zone_key探测错误结果分布

Figure 10 Distribution of no_zone_key detection error results

如图11所示,返回RRSIG_missing错误的服务器最多(1937台,约占全部服务器的54%),返回的其他错误类型不是预设的错误,其余约41%的服务器是没有返回错误代码或者连接失败的,说明几乎全部的服务器现在可以判断有错误,但是不能正确返回预设的错误。

如图12所示,返回有错误但是不能返回具体错误类型的服务器最多(1971台,约占全部服务器的54%),返回的其他错误类型不是预设的错误,说明几乎全部的服务器现在可以判断有错误,但是不能正确返回预设的错误。

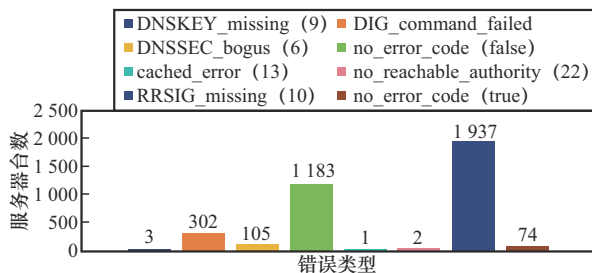


图11 NSEC_missing探测错误结果分布

Figure 11 Distribution of NSEC_missing detection error results

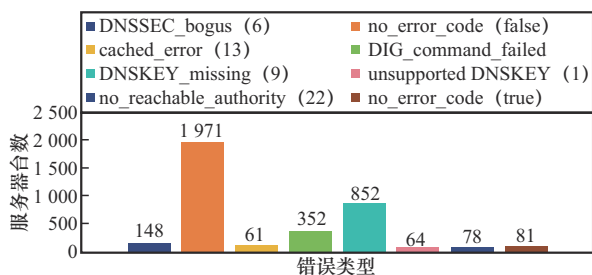


图12 unsupported_DS探测错误结果分布

Figure 12 Distribution of unsupported_DS detection error results

总体而言,目前的DNS递归服务器对NSEC_missing、no_zone_key、unsupported_DS这3种错误,并不能有效地返回预设的错误,说明目前DNSSEC的部署并不是特别规范和完善。

此外,unsupported_DNSKEY这一特殊情况,其探测错误结果分布的可视化结果如图13所示。

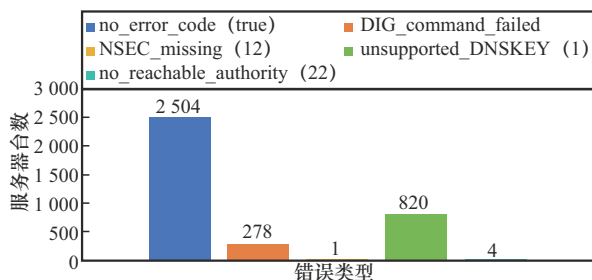


图13 unsupported_DNSKEY探测错误结果分布

Figure 13 Distribution of unsupported_DNSKEY detection error results

如果服务器返回error_code,判断出域名的DNSSEC配置错误,就不会解析域名。但是,本文实验在探测这个错误的时候,有820台服务器返回了error_code,即服务器判断存在配置错误时,依然返回了A记录,也就是成功解析域名。通过分析,这是由于不同服务器的判错标准不一

样,有的服务器较严格,而有的服务器较宽松。

对于 unsupported_DNSKEY 的配置错误,本文实验采用 SHA-1 算法获得 DNSKEY,导致密钥长度只有 512 位,比较不安全,于是部分服务器返回了 unsupported_DNSKEY_key_size,即不支持的密钥长度。当 SHA-1 被引入时,对抗攻击的安全强度估计为 80 位,然而,随着密码分析和计算能力的改进,这个安全强度已降至 60 位或更低,SHA-1 的安全强度降低也使得攻击者能够更快地生成哈希碰撞,从而冒充用户解密/加密数据。密钥过短导致算法不再安全,该算法被服务器认为是不支持的,因此报错。然而,由于其余的验证过程全部成功通过,因此也返回了 A 记录。这就出现了成功解析但返回错误的情况。而其余的服务器的内部校验机制不够严格,认为这个密钥长度属于可接受范围,导致其根本没有对这个密钥过短的问题返回错误。这也说明目前服务器的 DNSSEC 配置仍然不够规范、不够完善。

综上所述,可以得出以下结论。

1) 对于全部的 8 种错误,几乎所有支持 DNSSEC 验证的递归服务器可以正确验证出该域名的 DNSSEC 配置确实有错误。但是,因为每台服务器查询到的信息不一样,所以返回的错误不完全一致,而是存在多种错误。

2) 对于 signature_expired、signature_not_valid、RRSIG_missing、DNSKEY_missing 等错误,大部分支持 DNSSEC 的服务器可以正确返回对应的错误,而对于其余的几种错误,这些递归服务器则无法正确判断错误类型。

5 结束语

DNS 递归服务器能否检测出域名存在 DNSSEC 错误配置,对于保障 DNS 安全有重大的意义。本文通过给 8 个三级子域配置不同的 DNSSEC 错误,并使用 DNS 递归服务器向这 8 个子域发送请求,以此判断服务器是否有检测 DNSSEC 错误配置与错误类型的能力。实验结果表明,除了 DNS 服务器对 DIG 版本可能不支持(即出现 no_error_code)的情况外,几乎所有支持 DNSSEC 的递归服务器可以验证出该域名的 DNSSEC 错误配置。然而,在检测具体错误类型上,部分 DNS 服务器则无法检测出准确的错误配置类型。

本文首次引入了对递归侧 DNS 服务器进行验证的思路,有效弥补了 DNSSEC 配置安全分析在此领域的空缺。同时,实验结果表明,目前的 DNS 服务器还不能准确地检测出 DNSSEC 错误配置情况,需要后续服务器开发者增强相关安全意识,严格遵守 DNSSEC 相关标准要求。

在未来 DNSSEC 的全面部署中,域名注册管理机构和服务机构也应致力于提供用户友好的 DNSSEC 签名服务,以降低技术应用的入门障碍。此外,这些机构应积极参与国际合作,为 DNS 安全标准的制定贡献力量,并深入理解国际 DNSSEC 的进展与实施策略,从而在网络与信息安全领域中提升影响力。同时,应加强域名注册信息的审核与备案机制,提升域名服务机构的管理效能,确保域名解析过程的安全性。

参考文献:

- [1] ATKINS D, AUSTEIN R. Threat analysis of the domain name system (DNS)[J]. RFC, 2004, 3833: 1-16.
- [2] ADRICHEM N L M, BLENN N, LÚA A R, et al. A measurement study of DNSSEC misconfigurations[J]. Security Informatics, 2015, 4(1): 8.
- [3] HERZBERG A, SCHULMANN H. Towards adoption of DNSSEC: availability and security challenges[J]. IACR Cryptol EPrint Arch, 2013, 2013: 254.
- [4] CHUNG T, VAN RIJSWIJK-DEIJ R, CHANDRASEKARAN B, et al. A longitudinal, {end-to-end} view of the {DNSSEC} ecosystem[C]// Proceedings of the 26th USENIX Security Symposium (USENIX Security 17). BOSTON:USENIX, 2017: 1307-1322.
- [5] KUMARI W, HUNT E, ARENDS R, et al. Extended DNS errors[R]. 2020.
- [6] KIM T H, REEVES D. A survey of domain name system vulnerabilities and attacks[J]. Journal of Surveillance, Security and Safety, 2020, 1(1): 34-60.
- [7] MOCKAPETRIS P V. RFC0882: domain names: concepts and facilities[S]. 1983.
- [8] LU C Y, LIU B J, LI Z, et al. An end-to-end, large-scale measurement of DNS-over-encryption: how far have we come [C]// Proceedings of the Internet Measurement Conference. New York: ACM, 2019.
- [9] CHANG D, ZHANG Q, LI X. Study on os fingerprinting and nat/tethering based on DNS log analysis[C]// Proceedings of the IRTF & ISOC Workshop on Research and Applications of Internet Measurements (RAIM). 2015.
- [10] PARK J, JANG R, MOHAISEN M, et al. A large-scale behavioral analysis of the open DNS resolvers on the Internet[J]. IEEE/ACM Transactions on Networking, 2022, 30(1): 76-89.
- [11] WANG M, ZHANG Z, XU H. DNS configurations and its security analyzing via resource records of the top-level domains[C]//Pro-

ceedings of the 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID). IEEE, 2017: 21-25.

- [12] CHUNG T, VAN RIJSWIJK-DEIJ R, CHOFFNES D, et al. Understanding the role of registrars in DNSSEC deployment[C]//Proceedings of the 2017 Internet Measurement Conference. New York: ACM, 2017: 369-383.
- [13] 徐格, 凌思通, 李琦, 等. 基于区块链的网络安全体系结构与关键技术研究进展[J]. 计算机学报, 2021, 44(1): 55-83.
XU K, LING S T, LI Q, et al. Research progress of network security architecture and key technologies based on blockchain[J]. Chinese Journal of Computers, 2021, 44(1): 55-83.
- [14] LIAN W, RESCORLA E, SHACHAM H, et al. Measuring the practical impact of DNSSEC deployment[C]//Proceedings of the 22nd USENIX Conference on Security. BOSTON: USENIX, 2013: 573-588.
- [15] 张曼, 姚健康, 李洪涛, 等. DNS 信道传输加密技术: 现状、趋势和挑战[J]. 软件学报, 2023, 35(1): 309-332.
ZHANG M, YAO J K, LI H T, et al. Encryption technologies for DNS channel transmission: status, trends and challenges[J]. Journal of Software, 2023, 35(1): 309-332.
- [16] MIGAUT D, GIRARD C, LAURENT M. A performance view on DNSSEC migration[C]//Proceedings of the 2010 International Conference on Network and Service Management. Piscataway: IEEE Press, 2010: 469-474.
- [17] YANG Z W, GAO S, XU J L, et al. Authentication of range query results in mapreduce environments[C]//Proceedings of the Third International Workshop on Cloud Data Management. New York: ACM, 2011: 25-32.
- [18] DECCIO C T. A case for comprehensive DNSSEC monitoring and analysis tools[R]. 2011.
- [19] 王文通, 胡宁, 刘波, 等. DNS 安全防护技术研究综述[J]. 软件学报, 2020, 31(7): 2205-2220.
WANG W T, HU N, LIU B, et al. Survey on technology of security enhancement for DNS[J]. Journal of Software, 2020, 31(7): 2205-2220.
- [20] DAI T, SHULMAN H, WAIDNER M. DNSSEC misconfigurations in popular domains[C]//Proceedings of the Cryptology and Network Security: 15th International Conference. Berlin: Springer, 2016: 651-660.

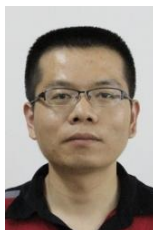
[作者简介]



刘林晖(2001-), 男, 江西赣州人, 暨南大学硕士生, 主要研究方向为 DNS 安全、邮件安全。



涂菲帆(2000-), 女, 广东广州人, 暨南大学硕士生, 主要研究方向为时间序列异常检测。



陈勇(1977-), 男, 湖南涟源人, 中国互联网络信息中心高级工程师, 主要研究方向为 DNS 安全。



左鹏(1985-), 男, 安徽, 中国互联网络信息中心高级工程师, 主要研究方向为网络安全、互联网基础资源关键技术。



刘东杰(1989-), 女, 河南信阳人, 暨南大学讲师, 主要研究方向为网络安全、机器学习。



张银炎(1994-), 男, 广东汕尾人, 暨南大学研究员, 主要研究方向为计算智能。



耿光刚(1980-), 男, 山东泰安人, 博士, 暨南大学教授, 主要研究方向为机器学习、大数据分析和互联网基础资源安全。