DNS安全

域名系统（DNS）位于协议栈应用层，为互联网提供核心服务，包括web页面访问，收发邮件等互联网应用通过DNS查询IP地址后获取资源，已成为互联网关键基础设施

作为一个分布式数据库，域名系统使得任意联网计算机能够通过域名访问互联网；灵活的扩展性和优异的解析性能，能够持续高效地支持数亿规模的域名解析。互联网渐进式演进发展模式决定了域名系统中大量安全威胁将长期存在，不可能通过重新设计的途径解决。

DNS 概述：P9

DNS 演进 P10

网络空间两套命名体系：用于路由寻址的IP地址和便于人类记忆的域名（Domain Name）

域名系统（DNS）功能：实现域名与IP间转换

DNS域名结构：P12

域名系统采用层次化树形结构：树形最顶层为根（Root），进一步划分顶级域（TLD），顶级域管理机构授权给二级域名（SLD），层次化授权行为最多可迭代127 次

DNS区域组织形式：P13

一个域名没有被分割成子域或包含了子域全部数据，则域名和区域相同

一个域名被分割成子域，每个子域有自己的区域时，域名和区域具有不同的意义

权威域名服务器 P14

每个DNS区域的权威域名服务器，发布关于该区域的信息，并响应DNS查询请求

权威域名服务器可以配置主从服务器，主服务器存储所有区域记录的记录，而从服务器使用自动更新机制维护主记录的副本

DNS使用及解析过程 P17

1. P19~21 DNS使用 & 请求过程：并不是每一次域名解析都完成整个查询流程：按照域名空间层次化结构自上而下应答，本地DNS - Root - 顶级域 - 二级域名
2. DNS迭代查询过程**交互信息格式**：P22~26
3. DNS 反向查询 P27

DNS攻击 P29

攻击原因：1. 客观上协议设计的不完美；2. 主观上基于利益驱动，攻击者不断挖掘漏洞

共同特征：1. 针对明文传输和无身份认证的实体进行欺骗性攻击；2. 寻找并突破域名间复杂依赖关系，实现对域名服务器攻击；3. 针对防护措施不足的服务器发起拒绝服务攻击。

DNS 各个层面的攻击 P30

用户主机；本地 DNS 服务器；互联网 DNS 服务器。

操控用户机 P31

1. 修改/etc/resolv.conf：将主机DNS服务器修改成攻击者控制的服务器，攻击者回复恶意内容
2. 修改/etc/hosts：直接修改地址与域名映射关系，修改为攻击者控制的IP地址
3. 嗅探回复：监听主机发出的DNS查询请求，注入恶意回复

本地缓存中毒攻击 P32~35

伪造DNS回复，攻击者需要知道请求中的一些参数，如UDP源端口号、请求交易ID、请求问题等；由于UDP包没有加密，攻击者可以在局域网直接捕获并解析请求。

1. 只针对回复部分伪造，影响面：一个主机名；在伪造回复中，把主机名映射到攻击地址，告诉本地DNS服务器域名对应 地址是攻击者的计算机。
2. 针对授权部分攻击，影响面：整个域；将ns.attack.com放在授权部分，查询目标域内任何一个主机名时，本地 DNS服务器把请求发给ns.attack.com。

远程缓存中毒攻击 P36

攻击难点：由于不能嗅探DNS请求，很难获取两个数据：UDP16bit的头部端口号，DNS头部的16bit交易ID。远程攻击者猜测准确的概率为1/2^32，成功概率极低。

实现攻击的前提：本地DNS服务器发起域名查询请求。

如何伪造一个被受害者接收的应答包 P38

接受欺骗应答面临的约束：

1. 缓存时限约束：域名不能在缓存中
2. 应答包匹配约束：端口号、交易ID匹配
3. 时间约束：伪造包比正常DNS应答快

P39 构建回复包头；P40 包头参数猜测；P41 构建回复包负载；P42 侧信道攻击；P43 基于IPID的攻击；P44 基于交易ID的攻击；幽灵域名 P46~47 恶意域名被删除后，利用DNS漏洞继续存活。

P47~49 来自恶意DNS服务器的污染

在附加部分伪造数据，在授权部分伪造数据，反向查找回复部分伪造。

拒绝服务攻击 DDoS：P50

如果攻击者可以成功攻破root区域的服务器，则整个互联网将会崩溃。但是由于root域名服务器基础设施采用分布式部署方式，很难被全部攻破。

DNS攻击预防策略

通过非对称加密验证身份 - DNSSEC：P55~60

DNSSEC引入公钥加密/认证体系，通过签名提供端到端数据真实性和完整性保护。部署DNSSEC的权威域名服务器对其区域文件中资源记录用私钥进行数字签名。接收方用公钥验证签名，判定域名解析结果是否在传输过程被篡改。

DMSSEC 的不足 P60~62

1. 经济因素制约其部署，大量签名和验证带来严重的负载，影响效率，验证能力取决于递归服务器以及客户端是否对数字签名记录做校验。
2. 离线存储与频繁使用密钥，私钥的安全无法保证；没有提供机密性和一致性检验，无法抵御重放攻击；只提供单向认证，没有对客户的认证，不能解决缓存中毒； 没有提供DoS防护。
3. 引入了新的实现和配置错误。
4. 用 DoT / DoH 改进 P61~62

基于系统管理：P63~64

1. 规范并梳理DNS配置过程中出现的漏洞
2. 限制用户在短时间内发起大量DNS查询
3. 增加端口猜测难度，如用于查询的UDP端口不再是默认的53，而在UDP端口范围内随机选择(排除预留端口)
4. 分布式部署、恶意流量过滤等方案
5. 严格检查：附加区记录和问题区问题不在同一域管辖，则不会采信，防范恶意权威DNS虚假记录污染缓存

新型架构设计 Blockstack：P65~66

Blockstack 旨在建立一个去中心化的域名系统及公钥基础设施

典型案例分析

Kaminsky攻击：P68~72

恶意人员发送随机查询请求到DNS服务器，抢在权威应答前伪造应答包发送给服务器，修改授权资源记录。关键：**在权威区和附加区实施欺骗**。

根本原因

- 缺乏端节点身份和发布内容验证
- 数据未采用加密传输
- 协议自身检查采信机制不足

防御策略

- 针对协议内容和通信实体设计可靠的验证方式
- 通过加密等策略确保对用户身份和发布内容进行鉴别，实现有效防御

恶意服务器回复伪造攻击：P73~76

攻击者可依赖目标权威服务器或 DNS 软件漏洞，控制特定权威域名服务器，篡改区域文件中的授权数据，形成恶意服务器回复伪造攻击，成功攻破权威域名服务器的难度较大，实际中可行性并不高

利用域名系统冗余的架构设计使很多域名之间存在错综复杂的解析依赖，通过控制其中一环，逐步实现劫持特定域名权威服务器

根本原因

- 域名系统存在复杂的解析依赖
- DNS管理方面缺乏验证配置内容能力，存在错误输入的记录

防御策略

- 规范域名应用管理，减少使用环节引入的安全威胁
- 对配置内容建立审核机制，确保每一条域名记录的正确性，从根上防止接受恶意信息

拒绝服务攻击：P77

向被攻击的服务器发送大量域名解析请求，给服务器带来了很大负载，超过一定数量造成 DNS服务器反应缓慢甚至停止服务

攻击者可以通过拒绝服务攻击使得整个国家的因特网受到严重威胁

攻击者通过控制大量用户发起DDoS攻击，受害域名服务器可以通过分布式部署的方式，提升自己的处理能力

根本原因

- 大量恶意请求到达被攻击服务器并被攻击服务器接受

防御策略

- 安全的网络体系架构，如采用真实网络地址保障每一台接入网络计算机 的安全性，建立快速DDoS溯源机制
- 专用DNS请求过滤系统，针对DNS数据包进行恶意流量进行过滤；分 布式部署降低攻击对性能的影响