DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels

(CCS ‘20)

主要贡献

DNS承载了许多其他的安全关键应用程序，如反垃圾邮件防御，路由安全等。DNS在TLS信任方面也起着至关重要的作用，破坏DNS记录完整性将会带来灾难性的安全隐患，攻击者发布的虚假证书，将会对公钥加密技术造成影响

•揭示了应用程序级操作，和操作系统级行为之间的相互作用；发现了**通用的对抗UDP源端口随机化的策略**——ICMP消息的全局速率限制引入的侧信道漏洞

•研究了源端口去随机化这种攻击方式对现有攻击模型的可行性，为了有足够时间进行去随机化攻击，找到了新的方法**扩大攻击窗口**

•针对各种DNS软件、配置和网络状况进行广泛评估——大多数情况下攻击只需要几分钟就能成功，最后给出了有效的缓解措施

背景：DNS缓存毒化攻击

•攻击的目标：DNS解析器

•攻击手段

- 攻击者请求DNS解析器解析一个域名，DNS解析器没有该域名的缓存，便向更权威的DNS服务器请求解析
- 攻击者试着在权威DNS服务器返回解析结果之前，向DNS解析器注入一个伪造的查询响应
- 只要攻击者伪造的响应中，源端口和目的端口，源IP和目的IP，以及一个16位的事务ID与请求包中的标识相同，攻击者就能成功向DNS解析器的缓存中注入一个假IP-域名解析记录
- 在2008年，只需要暴力破解16位的事务ID就能成功（65536），更不用说使用生日攻击能够更快速地猜出事务ID

背景：DNS防御措施

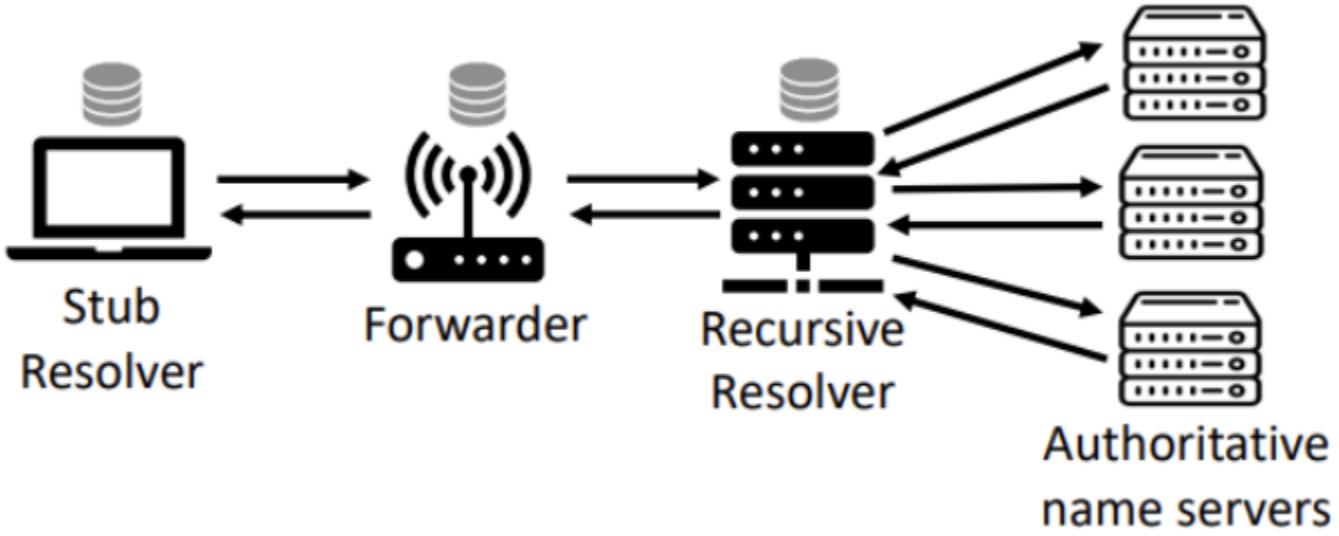
•源端口随机化：将随机性从2^16扩大到2^32

•域名中的字母随机化大写（0x20编码）：提供的随机性取决于字母的数量（请求和响应中的域名中的字母大小写要一致）

•随机选择DNS服务器地址：取决于域名服务器的多少；此外，已经证明，攻击者可以导致针对特定名称服务器的查询失败，从而有效地将解析程序“固定”到剩余的一个名称服务器上

•DNSSEC：它提供了来源鉴定和数据完整性的扩展；取决于解析器和权威域名服务器的支持，但布署率过于低了

背景：DNS缓存的攻击面

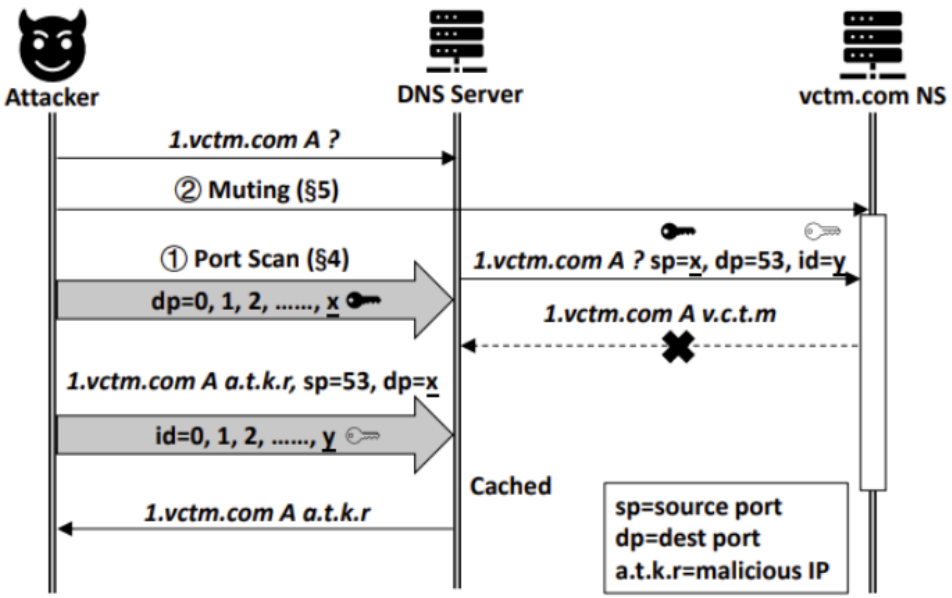


现代DNS基础设施拥有多层缓存
从技术上讲，所有缓存层都会受到DNS缓存中毒攻击

攻击概述-威胁模型

- 聚焦DNS转发器，DNS解析器
- 采用非路径（off-path）攻击（无法窃听转发器和解析器的通信）和IP欺骗
- 通用的攻击方法，适用于所有现代DNS软件栈，影响所有层的DNS缓存
- 破解了当前最常用、最有效的源端口随机化防御措施

攻击概述-攻击流程



1.推断DNS请求源端口

为了克服源端口的随机性，提出一种在网络栈中通用的侧信道来扫描并发现源主机使用哪个端口发送DNS解析请求，速度大约为1000猜/秒

2.扩展攻击窗口

正常情况下，一个DNS解析请求会在数十或者上百毫秒内收到上游主机的回复。作者发现了一种有效的策略将这个时间延长到数秒甚至10多秒

一旦得知源端口，攻击者就可以使用大量的虚假DNS响应暴力穷举TxID

推测DNS请求源端口

UDP源端口可扫描性分析 RFC-8085

- UDP端口不绑定到一对特定的端口和IP地址时**（公共端口）**
- 向该主机已经开放的端口发送UDP数据包时不会触发任何事件，操作系统接收数据包，到达应用时被丢弃　向该主机未开放的UDP端口发送数据包时，ICMP协议返回端口不可达消息

- 使用 connect() API将端口与远端主机的IP和端口绑定**（私有端口）**
- 向该主机已经开放的端口发送UDP数据包时，数据包会在系统层面被过滤并返回端口不可达的ICMP消息

推测DNS请求源端口

侧信道：ICMP速率限制

有效扫描UDP端口的限制来自主机对ICMP报文发送速率的限制：

- 对于发送给单个IP的ICMP报文的速率限制
- 全局ICMP报文发送速率限制

默认情况下，Linux对于ICMP报文发送速率规定如下：

- 单个IP速率限制1个/秒，最大累计突发数6个
- 全局速率限制50个/秒，最大累积突发数50个

推测公共端口号

探测过程

•攻击者在目标主机具有最大ICMP报文突发流量时，发送50个伪造的UDP探测包，每个包具有不同的源IP

•如果受害服务器在这50个端口中没有打开任何源端口，则会触发50个ICMP端口不可达消息（攻击者无法直接观测到这些消息）

•此时攻击者再发送一个包，如果收到ICMP报文说明里面有端口打开了，没有收到就说明这50个端口一个都没开

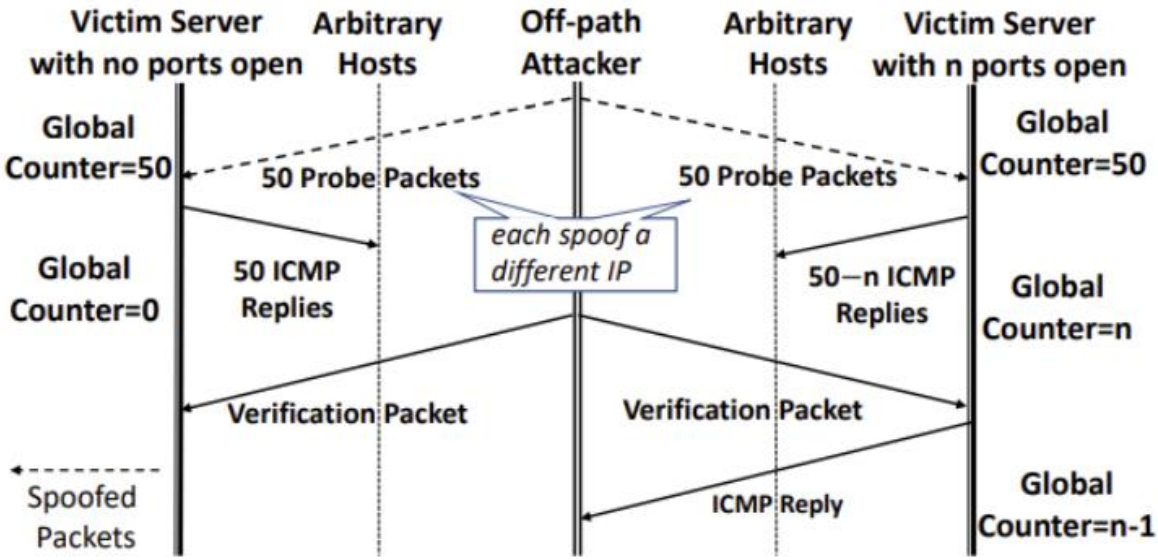


Figure 3: Fast Port Scanning of an Open Source Port

推测私有端口号-使用connect() API绑定

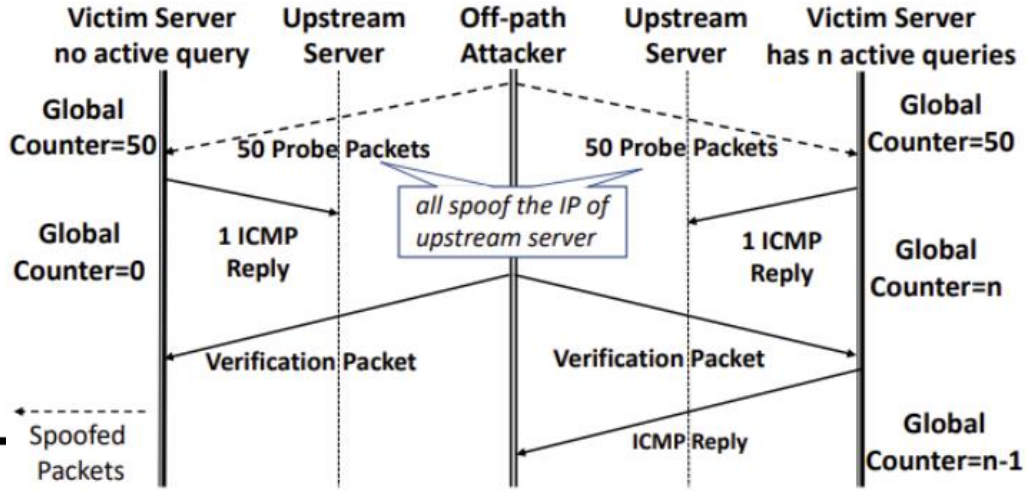
探测过程

•使用上游DNS服务器的源IP发送伪造的UDP数据包；如果它命中正确的源端口，则不会生成ICMP回复；否则就会生成一个ICMP回复

•可以使用与侧通道相同的全局ICMP速率限制来推断是否触发了此类ICMP消息

•分析了ICMP速率限制实现的源代码之后，我们发现全局速率限制是在每IP速率限制之前检查的；即使每IP速率限制最终可能决定不应发送ICMP回复，一个数据包仍将接受全局速率限制检查，并扣除一个令牌

•可以忽略单个IP速率限制，因为它只确定是否生成了最终的ICMP应答



扩展攻击窗口——转发器

Answer	www.attacker.com	CNAME	www.victim.com
	www.victim.com	A	1.2.3.4

Figure 5: Example Rogue Response Acceptable by a Forwarder (the victim domain record also cached)

• 攻击者首先发送一个查询自己的域名请求给转发器，这个举动最终会触发上游的解析器查询由攻击者控制的权威名称服务器。该名称服务器被故意设置成不响应解析请求，所以转发器会一直开放端口等待回应

• 由于RFC 8499，解析器负责所有递归查询，并向转发器提供最终答案，包括任何CNAME记录引起的重定向，这是一种防止转发器重复解析器工作的设计；转发器完全信任解析器的结果

• 因此，转发器将接收到如图所示的恶意响应，并缓存这两个记录；这种策略非常有效，因为我们可以对转发器施加最大的等待时间（即创建最大可能的攻击窗口），还可以制造多个CNAME记录，在让解析器有取得进展的错觉的同时，重置等待时间，以便时攻击窗口变得更长

扩展攻击窗口——解析器

•前提：权威域名服务器以不同粒度限制记录查询速率以缓解DNS放大攻击，成为响应速率限制（RRL）；RRL功能允许按IP、按前缀甚至全局限制触发的响应——如果达到限制，则响应要么被截断，要么被丢弃

•如果攻击者能够以高于配置限制的速率（使用目标解析程序的IP）注入伪造的DNS查询，则可以恶意利用该功能使名称服务器静音

缓解、防御DNS缓存攻击

- 通过额外的随机性和密码学解决方案来缓解
  - DNSSEC：通过数字签名来保证DNS应答报文的真实性和完整性
  - 0x20编码（增加随机性）域名随意大小写
  - DNS cookie：要求客户端和服务器交换一些非路径攻击者未知的额外秘密
- 关闭侧信道
  - 完全不允许传出ICMP回复，代价是丢失一些网络故障诊断和排除功能
  - ICMP全局速率限制随机化
- 阻止扩展攻击窗口
  - 适当配置RRL（响应速率限制）超出后的行为：向解析器发送一个强烈的信号，表明发生了什么不好的事情，解析器应该立即做出反应
  - 更加积极地设置DNS查询超时，比如始终低于1s

想法

内网下的网络基础设施需要重点保护：

- DNS服务器、DNS缓存记录
  - DNS服务器可能成为放大攻击的攻击源，需要限制RRL，但不放大流量
  - DNS缓存的正确性，内网DNS软件升级+合适的管理政策（开启必要的安全措施）
- 依靠DNS建立的PKI，怎么防护