MaginotDNS

绕过 DNS 缓存防御的马奇诺防线

段海新，李想
清华大学网络研究院
duanhx@tsinghua.edu.cn

帽子：Whitehat，Blackhat

演讲者介绍

段海新，清华大学网络研究院 教授

- **主要研究方向**
  - 网络安全，漏洞挖掘，网络测量
  - 网络地下产业检测
- **主要组织或活动**
  - 蓝莲花， XCTF， InForSec，DataCon

李想，清华大学网络研究院 博士生

- **主要研究方向**
  - 网络攻防，DNS安全、IPv6安全
- **主要成果**
  - 第一作者四大安全顶会论文4篇
  - 安全漏洞：170+ CVE
  - Blackhat 、DNS-OARC 报告人
  - XMap 扫描器

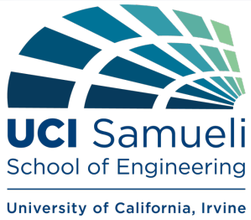
MaginotDNS 攻击:

绕过 DNS 缓存防御的马奇诺防线

The Maginot Line: Attacking the Boundary of DNS Caching Protection

[发表于 USENIX Security 2023]

2023-10-23



DNS 是互联网多种安全机制的基石

- **Web 和CDN：**
  - CNAME，内容路由
- **邮件路由： DNS MX 记录**
- **公钥证书**
  - **➔HTTPS 向CA申请公钥证书**
  - **➔ CA依赖DNS来签发证书**
- **口令恢复**
  - **➔ 发邮件，➔ 依赖DNS**

MaginotDNS 攻击可以直接污染整个顶级域名，比如 .com 和 .net

通过此种方式，
顶级域名下的所有域名均可被劫持

域名解析流程


末梢解析器（stub）
DNS递归服务器
（企业、ISP、公共）
权威DNS服务器
DNS
Cache
.
com
Test.com
根（Root）
顶级域
（TLD）
二级域
（SLD）
所有用户共享缓存记录

DNS缓存污染攻击

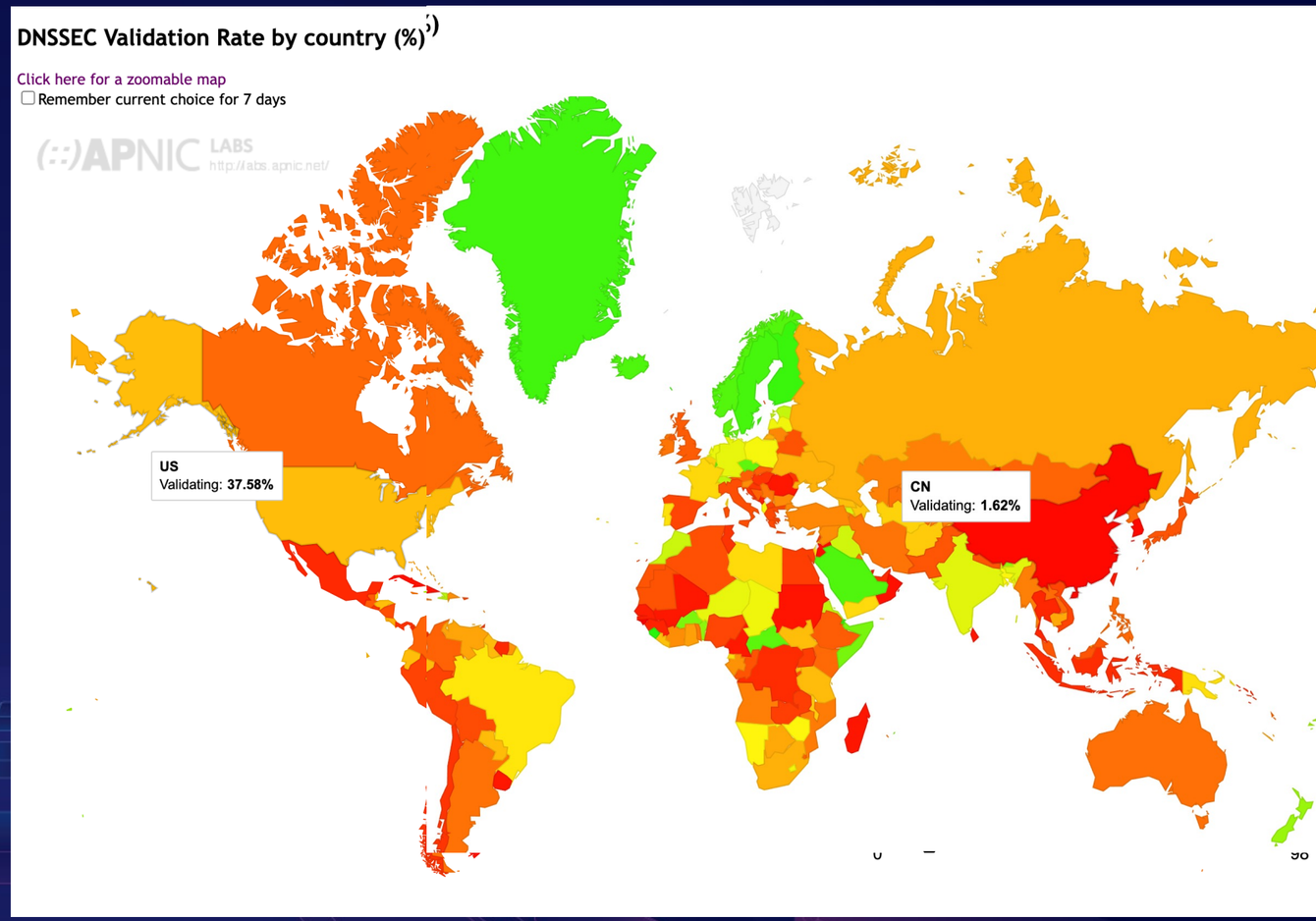
假设

- 不能监听网络流量，不是老大哥
- 不能入侵权威服务器，但可拥有自己的域名， attacker.com

通过网络系统或协议漏洞篡改缓存中的域名记录，从而实现钓鱼、中间人等攻击

DNSSEC 作为通用的解决方案？

- DNSSEC 彻底的解决方案
- 但1995年提出以来，部署缓慢
  - 30年， 中国1.62%
  - 剩下的98% 还需要多少年?
- 没有DNSSEC的缓解方案
- 我们的缓存污染攻击，也是为了促进DNSSEC的部署

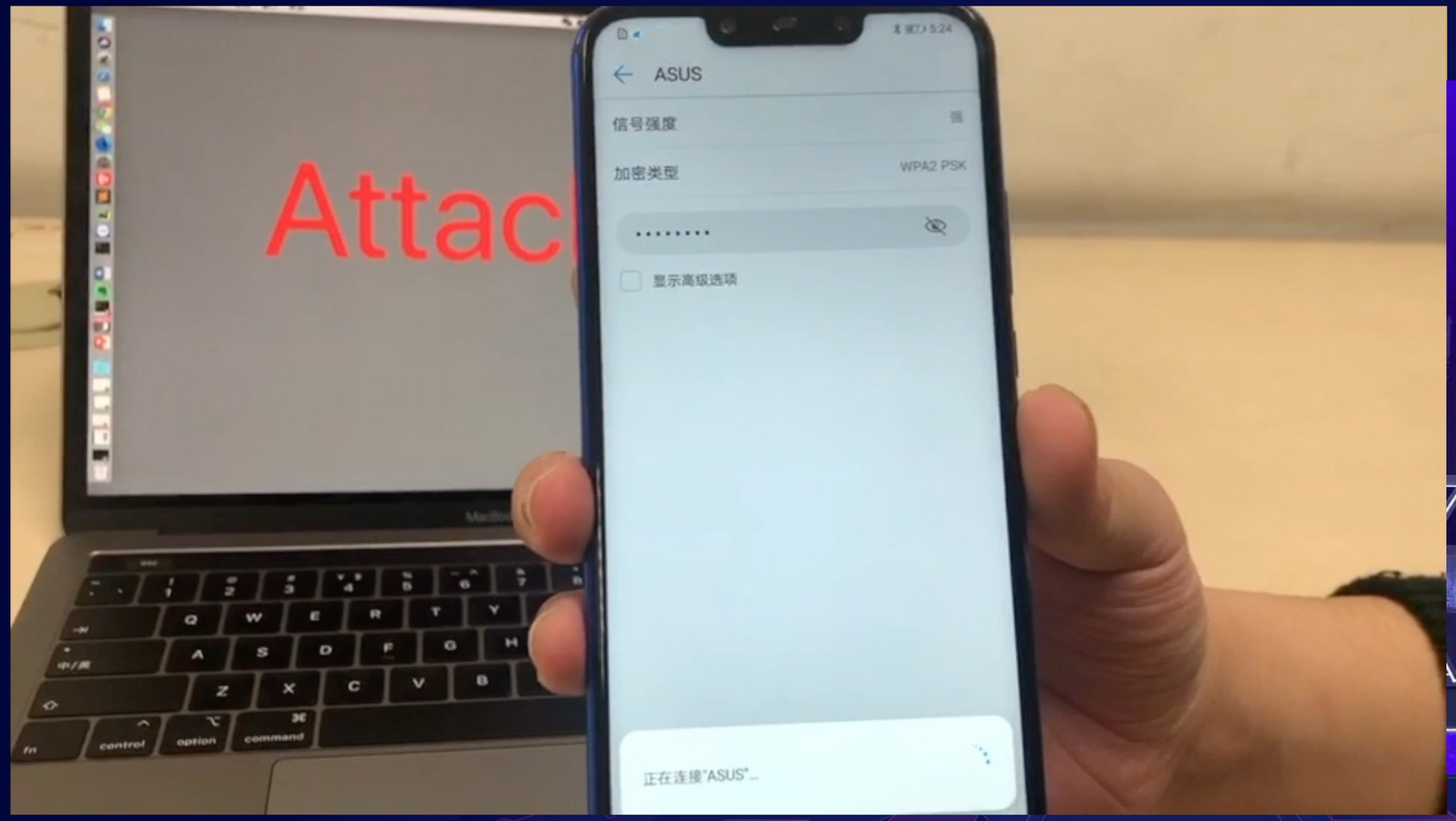


缓存污染攻击案例：巴西，2011

“要访问 Google.com 你需要安装 Google Defence”

攻击者用DNS缓存污染做钓鱼，长达10个月未被发现，影响数百万用户

GeekPwn， 2018



域名缓存污染攻击的前世今生

1997 Kashpureff Attack

2002 Birthday Attack

2008 Kaminsky Attack

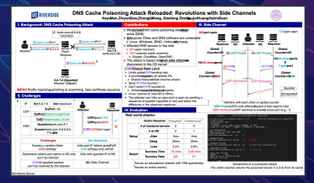
2013 Fragmentation Attack

2020 Attack on Forwarders

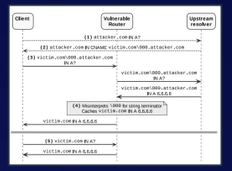
2020 SADDNS Attack

2021 Attack via Escaped Chars

2021 SADDNS v2 Attack



2022 Attack via Escaped Chars v2



2023 MaginotDNS Attack

攻击1：Kashpureff 攻击，1997

- **历史背景：** 90年代域名的战争——InterNIC 与 AlterNIC的冲突
  - **攻击方案：** 直接从**权威服务器处**回复虚假响应，解析器相信并**缓存所有回复**的资源记录
  - **漏洞成因：** **缺乏数据验证**

攻击者
客户端

Step1: Recursive query for
www.alternic.net/A

Step 2: Iterative query for
www.alternic.net/A

Step 3: Response including bogus
internic.net/NS RR

ISP resolver

Cache
InterNIC NS 1.2.3.4
www.internic.net A 5.6.7.8

Step 4:
Recursive query for
www.internic.net/A

Step 5:Bogus Response
www.internic.net A 5.6.7.8

ALTERNIC.NET
alternic.nic

攻击者控制
的服务器

1.2.3.4

攻击1：Kashpureff 攻击，1997

- **历史背景：** 90年代域名的战争——InterNIC 与 AlterNIC的冲突
  - **攻击方案：** 直接从**权威服务器处**回复虚假响应，解析器相信并**缓存所有回复**的资源记录
  - **漏洞成因：** **缺乏数据验证**

攻击者
客户端

Step1: Recursive query for
www.alternic.net/A

攻击者控制
的服务器

```
;; QUESTION SECTION:
;www.alternic.net.          IN  A

;; ANSWER SECTION:
www.alternic.net. IN A     198.58.118.167

;; AUTHORITY SECTION:
internic.net. 86400 IN NS ns.alternic.net.

;; ADDITIONAL SECTION:
ns.alternic.net. 86400 IN A 1.2.3.4
```

ISP resolver

Step 5:Bogus Response
www.internic.net A 5.6.7.8

1.2.3.4

防御1：域名辖区原则（Bailiwick Rules），1997

- **RFC2181：Clarifications to the DNS Specification**
  具体策略：在接收响应回复时对其合法性进行检查
  - **辖区原则**：回复资源记录的域名属于对应权威域名服务器**管辖区域的子域名**
  - **执行方式**：**符合辖区原则即被缓存**，否则被移除

攻击2：Dan Kaminsky， 2008

- **Dan Kaminsky 攻击，2008, Blackhat**
  - **漏洞成因**：仅依靠 16-bit 的 TXID 防御

Query ID	QR	Opcode	AA	TC	RD	RA	Z	rcode
Question count							Answer count	
Authority count							Addl. Record count	
DNS question or answer data								

Step 1: Recursive query for **www123.mybank.com/A**

If TXID matching, success!

Step 2: **TXID=1001**: Iterative query for **www123.mybank.com/A**

Step 4: Response

TXID=1001	
QD	www123.mybank.com A?
AN	(empty)
AU	mybank.com NS ns.mybank.com
AR	ns.mybank.com A 1.1.1.1

TXQID=1000 TXID=1001 TXID=1002

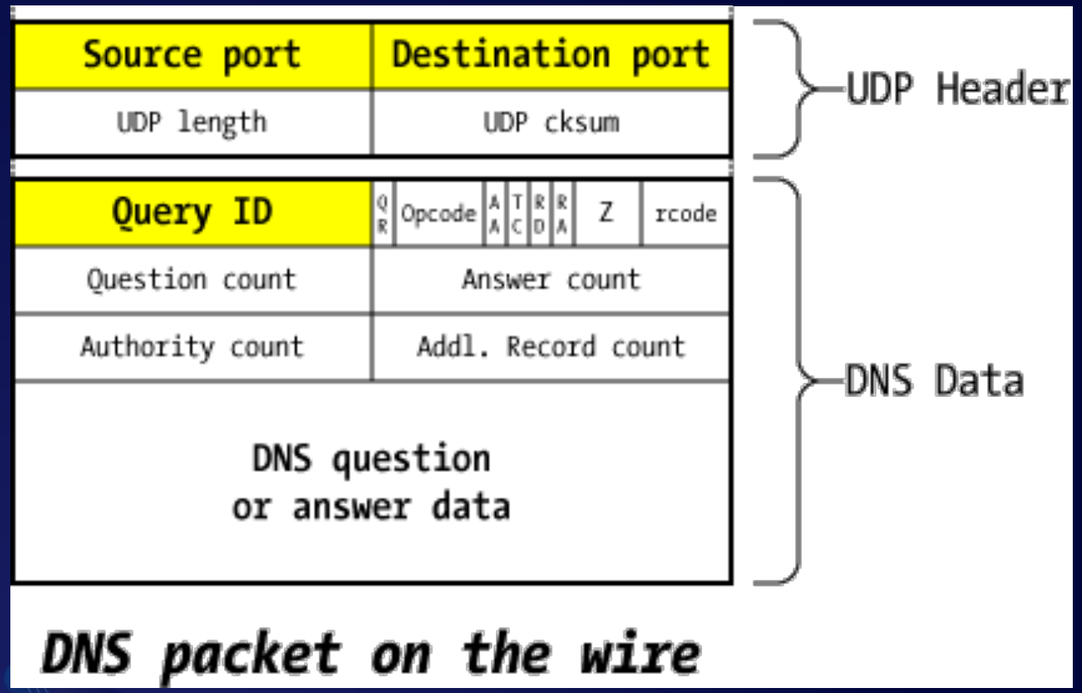
Step 3: Response

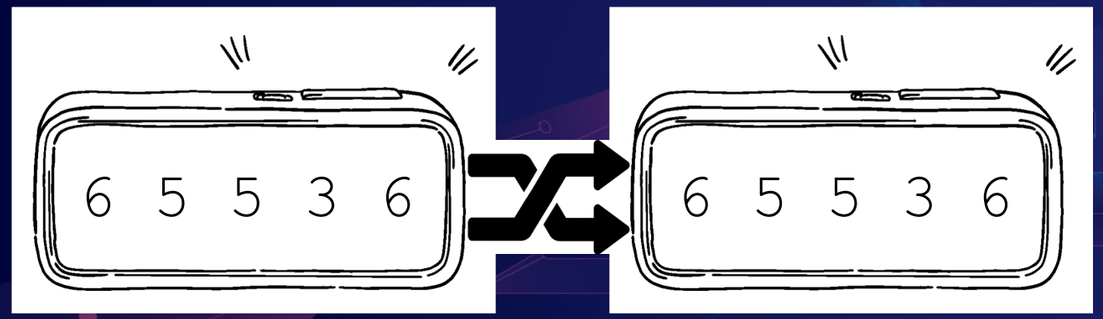
TXID=XXXX	
QD	www123.mybank.com A?
AN	(empty)
AU	mybank.com NS ns.mybank.com
AR	ns.mybank.com A 6.6.6.6

If TXID not matching, start the attack again with another domain: www456.mybank.com

防御方案：DNS 查询源端口随机化

- **用于抵御 Kaminsky 攻击**
  - **具体策略：提升校验字段熵值空间的大小**，增加猜解的难度
  - **熵值空间：16位源端口** x 16位 TXID = 32位校验值
  - **抵御效果：**32位空间使得几乎无法暴力猜解

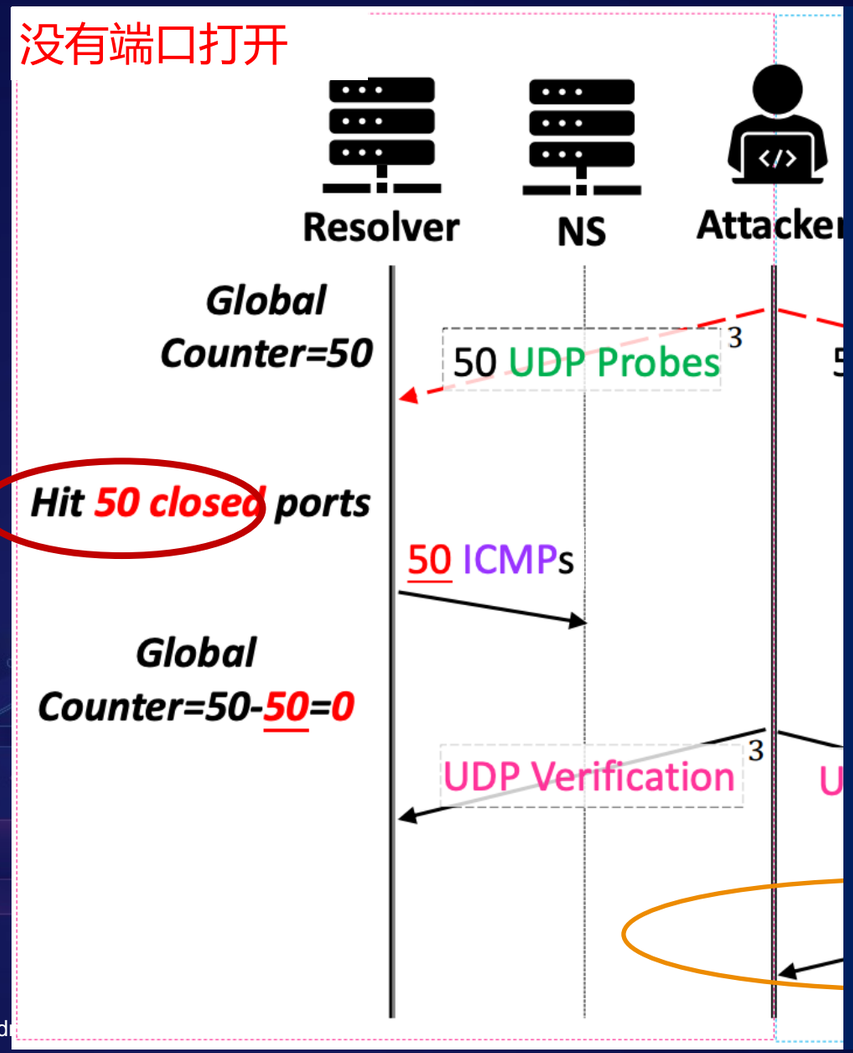




2^{32} = 4,294,967,296

攻击3: SAD DNS 侧信道泄露源端口号

- 钱志云@UCR, **DNS Cache Poisoning Attack Reloaded: CCS 2020 Best Paper**
  - **攻击方案**：利用 **Linux OS 侧信道**猜测 DNS 查询源端口
  - **根因**：操作系统 **ICMP 全局计数器**泄露源端口使用状态



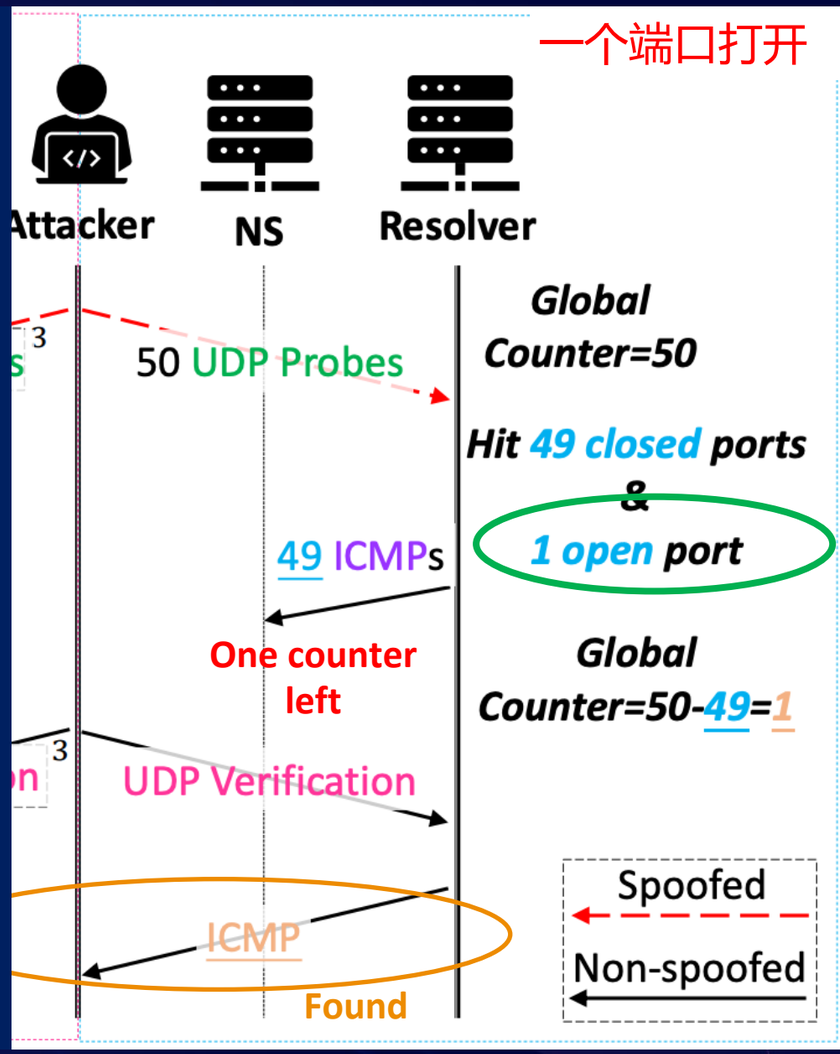
No counter left

引用：www.sadd

攻击3: SAD DNS 侧信道泄露源端口号

- 钱志云@UCR, **DNS Cache Poisoning Attack Reloaded: CCS 2020 Best Paper**
  - **攻击方案**：利用 **Linux OS 侧信道**猜测 DNS 查询源端口
  - **漏洞成因**：操作系统 **ICMP 全局计数器**泄露源端口使用状态

No counter left

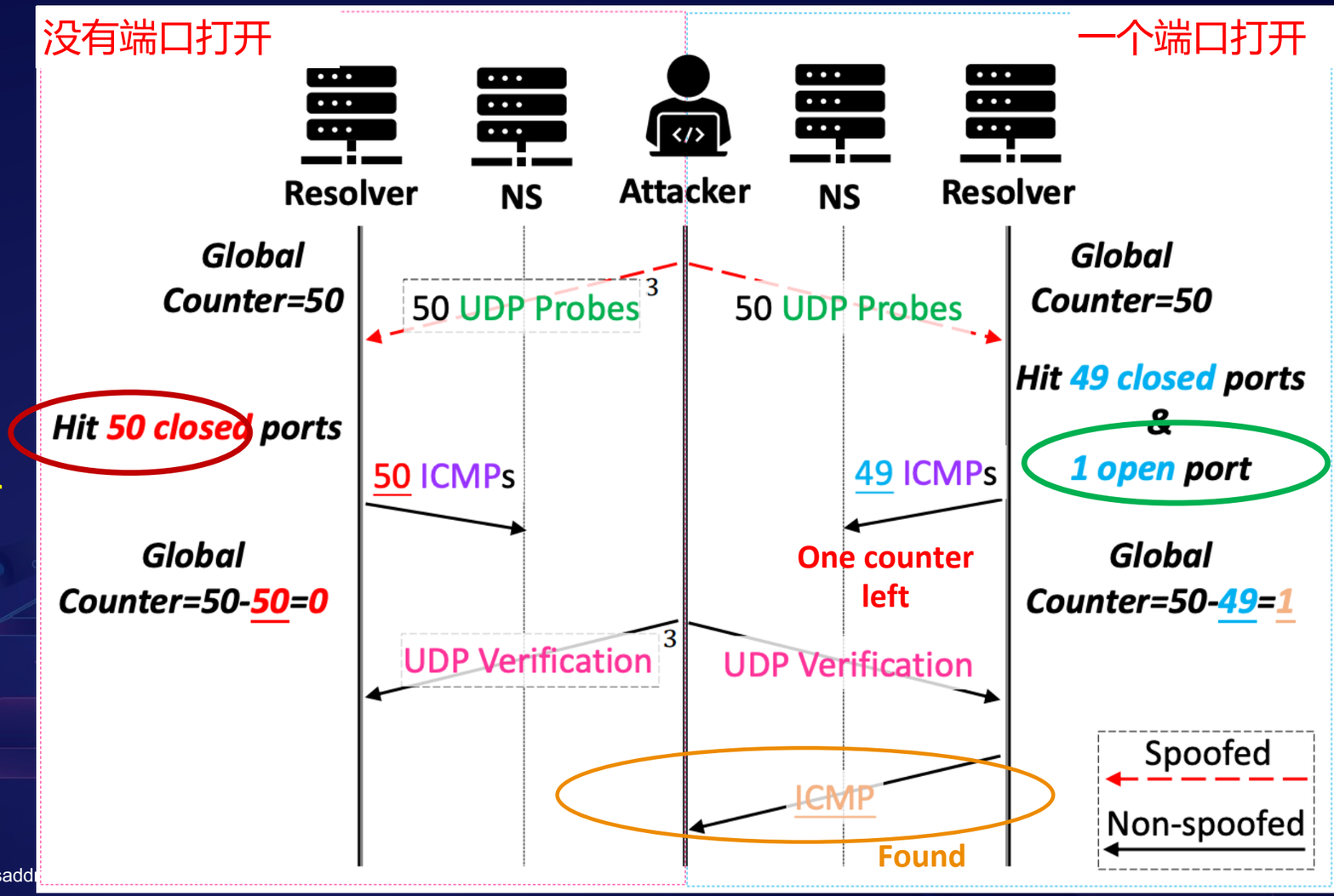


引用：www.sa

攻击3: SAD DNS 侧信道泄露源端口号

- 钱志云@UCR, **DNS Cache Poisoning Attack Reloaded: CCS 2020 Best Paper**
  - **攻击方案**：利用 **Linux OS 侧信道**猜测 DNS 查询源端口
  - **漏洞成因**：操作系统 **ICMP 全局计数器**泄露源端口使用状态

No counter left



防御3：ICMP 全局计数器随机化

- **用于抵御基于侧信道的域名缓存污染攻击**
  - **ICMP 全局计数器增长随机化**
    - 已被 Linux 内核采纳应用

```
icmp: randomize the global rate limiter

Keyu Man reported that the ICMP rate limiter could be used
by attackers to get useful signal. Details will be provided
in an upcoming academic publication.

Our solution is to add some noise, so that the attackers
no longer can get help from the predictable token bucket limiter.

Fixes: 4cdf507d5452 ("icmp: add a global rate limitation")
Signed-off-by: Eric Dumazet <edumazet@google.com>
Reported-by: Keyu Man <kman001@ucr.edu>
Signed-off-by: Jakub Kicinski <kuba@kernel.org>
```

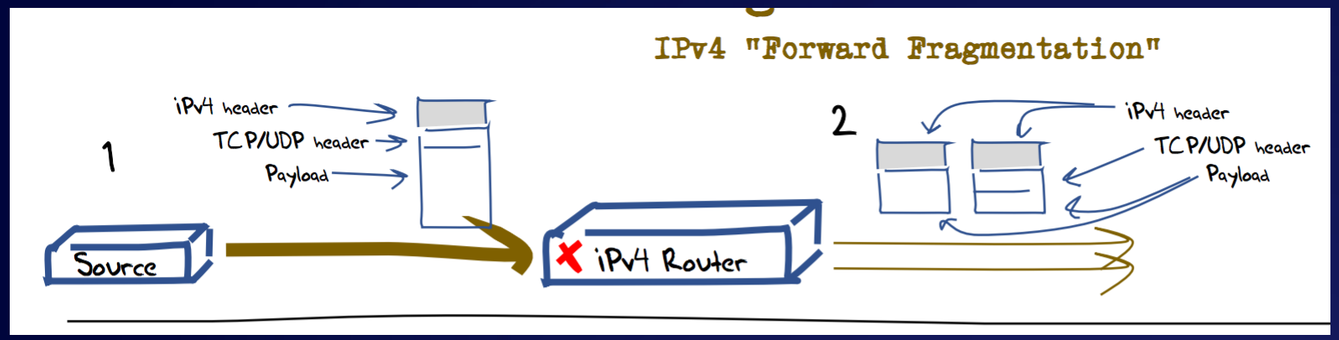
```
credit = min_t(u32, icmp_global.credit + incr, sysctl_icmp_msgs_burst);
if (credit) {
        credit--;
        /* We want to use a credit of one in average, but need to randomize
         * it for security reasons.
         */
        credit = max_t(int, credit - prandom_u32_max(3), 0);
        rc = true;
}
```

引用：git.ker

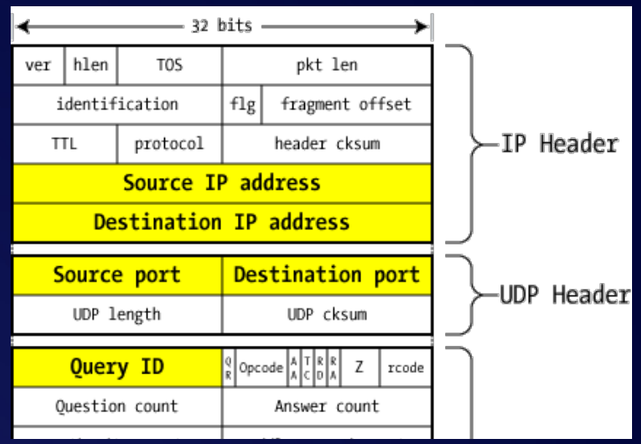
  - **减少域名查询解析的超时时间**
    - **该攻击需要一定的时间猜测源端口，减少超时时间可以提升攻击的难度**
    - **防止权威服务器被轻易的 mute**
  - **通用型方案**
    - DNSSEC、0x20

预备知识：IP 分片（fragementation）

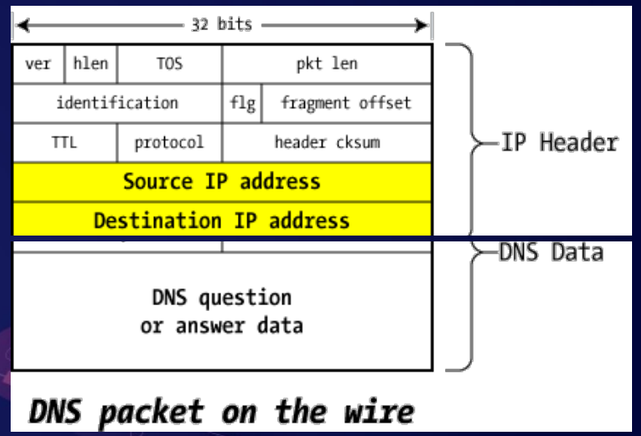
- 为什么要分片?



Frag1



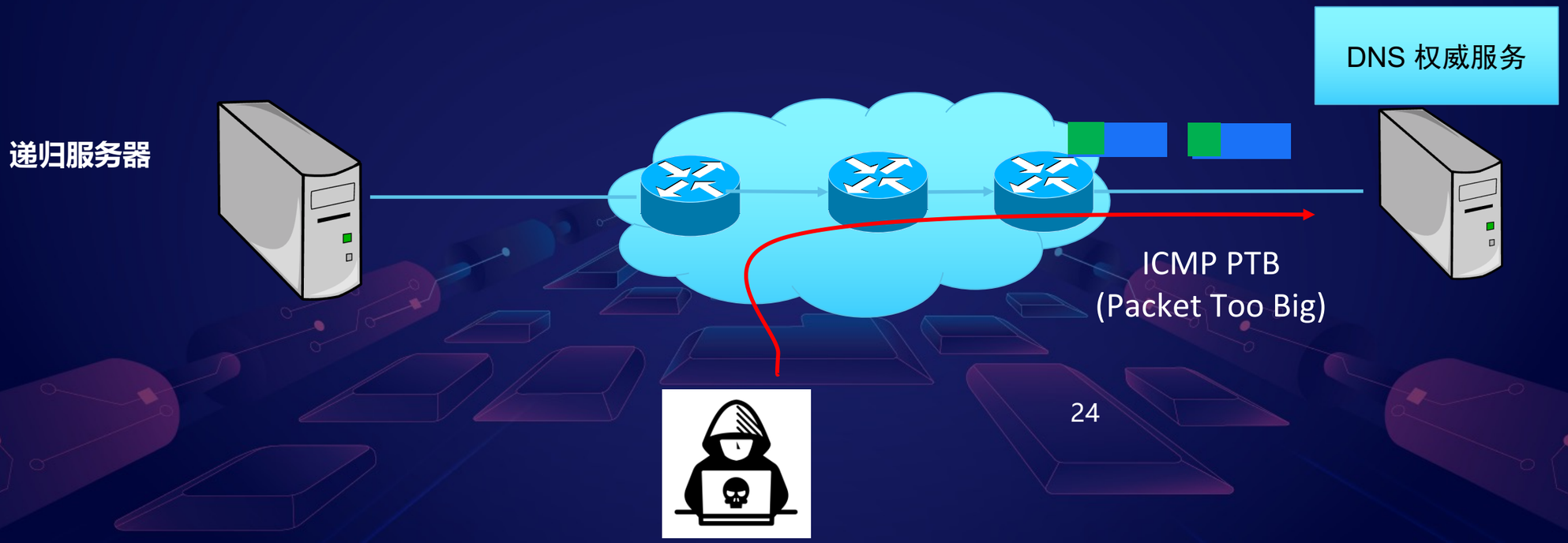
Frag2



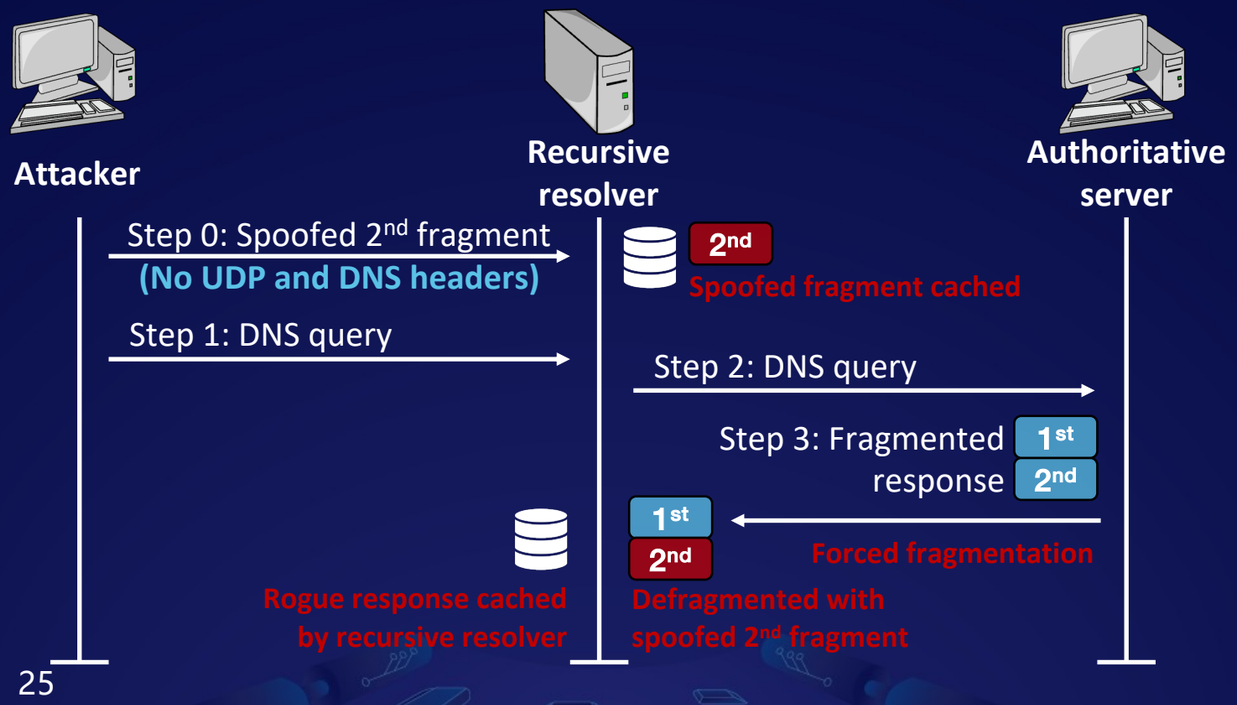
第二个分片中没有UDP 端口、DNS TXID

攻击3：基于分片的域名缓存污染

- **Herzberg，Shulrnan：Fragmentation Considered Poisonous， 2013**
  - **攻击方案：**利用**第二个分片不存在校验字段**的方式进行虚假回复的注入
  - **攻击效果：**解析器相信并缓存**分片重组后回复中的数据**
  - **漏洞成因：****接受较小的分片数据包**



攻击3：基于分片的域名缓存污染，2013



防御：Linux 内核已经不允许小的DNS响应分片

- 2020年，Alexa Top 10万 的域名服务器中只有0.7%可以把MTU将至528 Byte 以下
- 一般DNS响应 小于 512 Bytes，因此这种分片攻击不太可行了

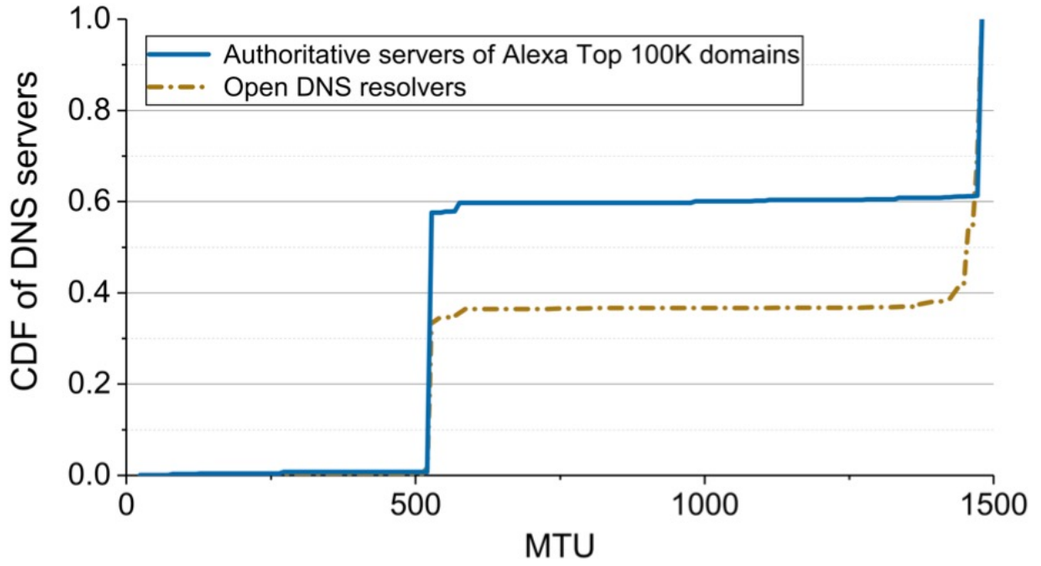
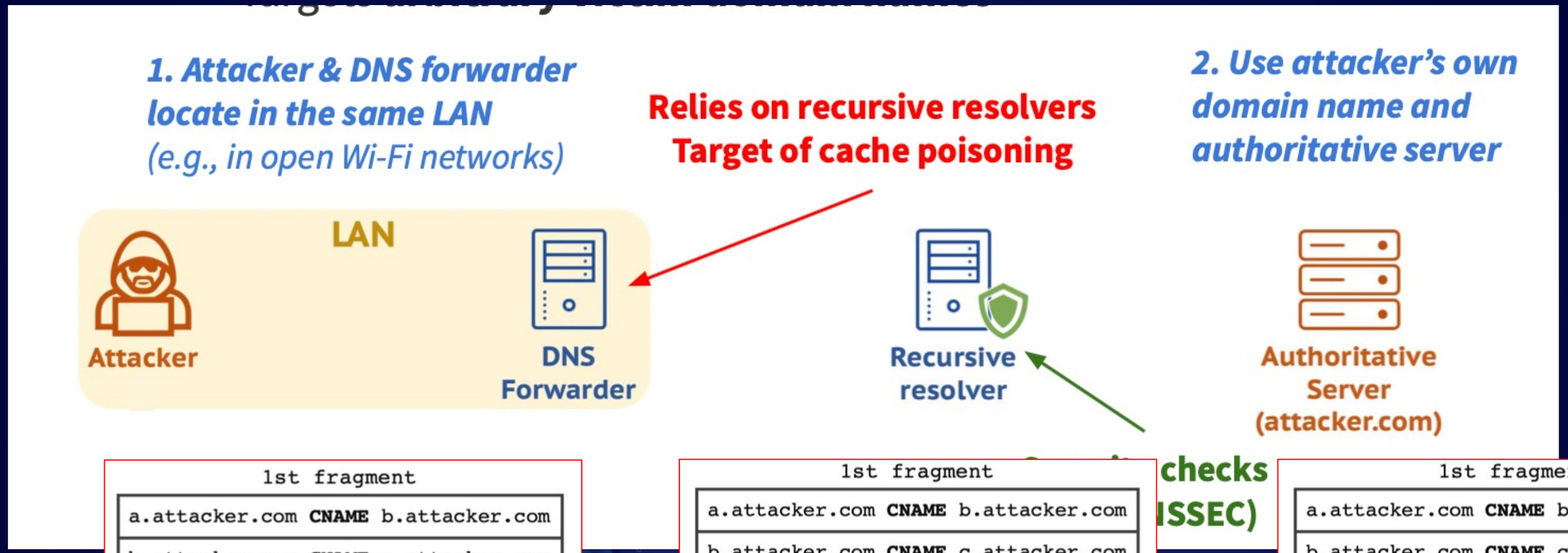


Figure 2: CDF of lowered MTU of a) authoritative servers of Alexa Top 100K domains, and b) 2M open DNS resolvers from an Internet-wide scan.

攻击4：CNAME 超大响应，转发服务器的缓存污染



1st fragment

a.attacker.com **CNAME** b.attacker.com
b.attacker.com **CNAME** c.attacker.com
c.attacker.com **CNAME** d.attacker.com
…

x.attacker.com **CNAME** y.attacker.com
y.attacker.com CNAME victim.com
victim.com A a.t.k.r

Spoofed 2nd fragment

1st fragment

a.attacker.com **CNAME** b.attacker.com
b.attacker.com **CNAME** c.attacker.com
c.attacker.com **CNAME** d.attacker.com
…

x.attacker.com **CNAME** y.attacker.com
y.attacker.com **CNAME** z.attacker.com
z.attacker.com **A** x.x.x.x

2nd fragment

1st fragment

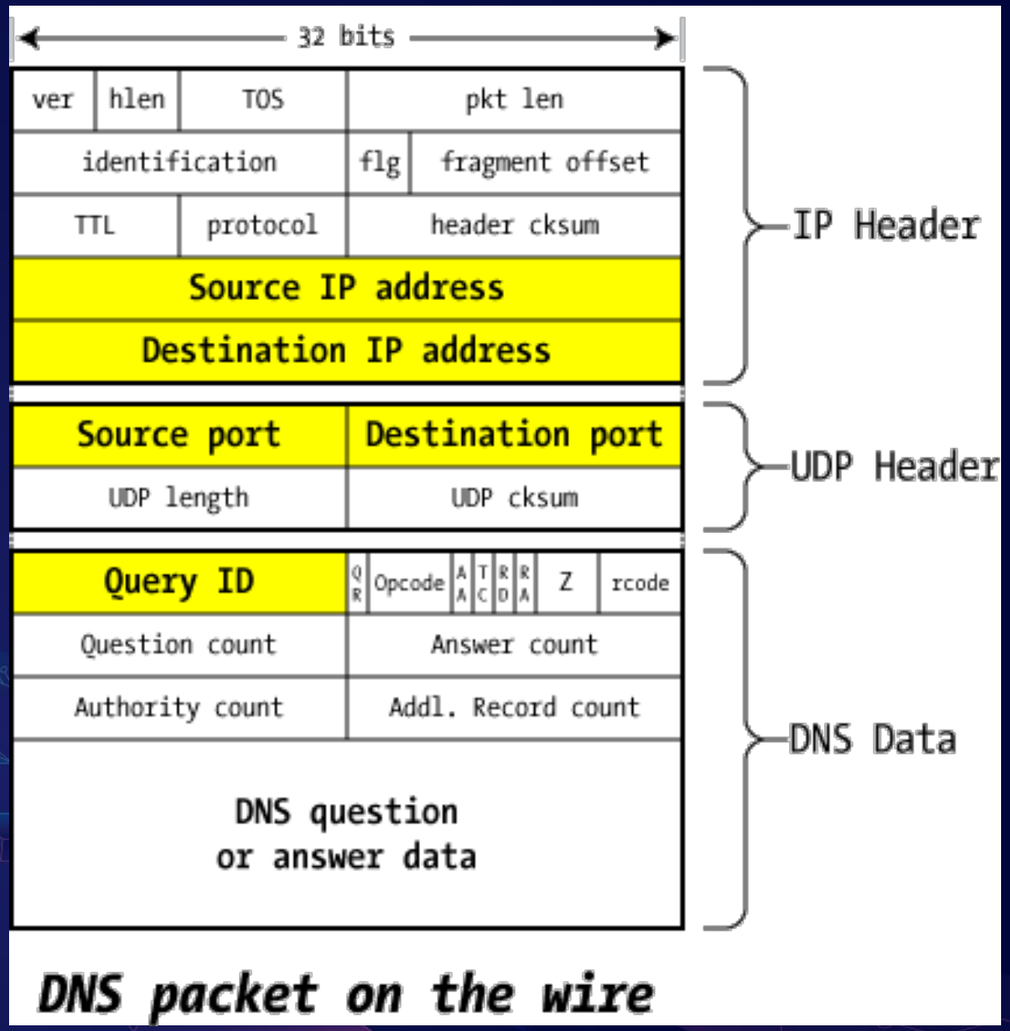
a.attacker.com **CNAME** b.attacker.com
b.attacker.com **CNAME** c.attacker.com
c.attacker.com **CNAME** d.attacker.com
…

x.attacker.com **CNAME** y.attacker.com
y.attacker.com **CNAME** z.attacker.com
z.attacker.com **A** x.x.x.x

2nd fragment

防御方案：限制分片分片大小、IPID随机化

- **用于抵御基于分片的域名缓存污染攻击**
  - **限制分片**
    - **尽量不分片或限制分片的条件**可以避免此类攻击
    - 回退到 TCP，限制分片的数量和重组的时间等
  - **IPID 随机化**
    - 无需猜测 UDP/DNS 的校验字段，但需要预测 IPID
    - **IPID 完全随机化**可以保障第二个虚假的分片不被重组
  - **其它方案**
    - 0x20编码，各个资源记录的域名均大小写随机化
    - DNS Cookie: 类似 SYN Cookie 的随机数密钥



开放递归服务器分布及防污染措施（2020年8月）

●DNSSEC验证

配置问题	世界	中国
支持DNSSEC	38.14%	13.88%
RRSIG过期	14.7%	4.79%
RRSIG缺失	33.4%	8.99%
RRSIG错误	13.72%	4.79%
DS缺失	13.75%	4.40%
DS错误	3.12%	4.80%
正确验证率	2.69%	4.04%

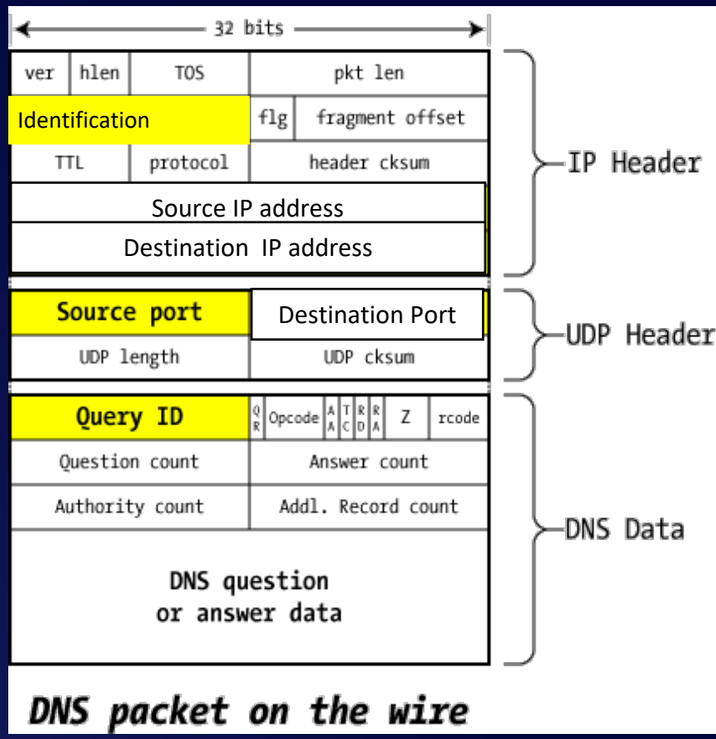
●非密码措施（各种随机化）

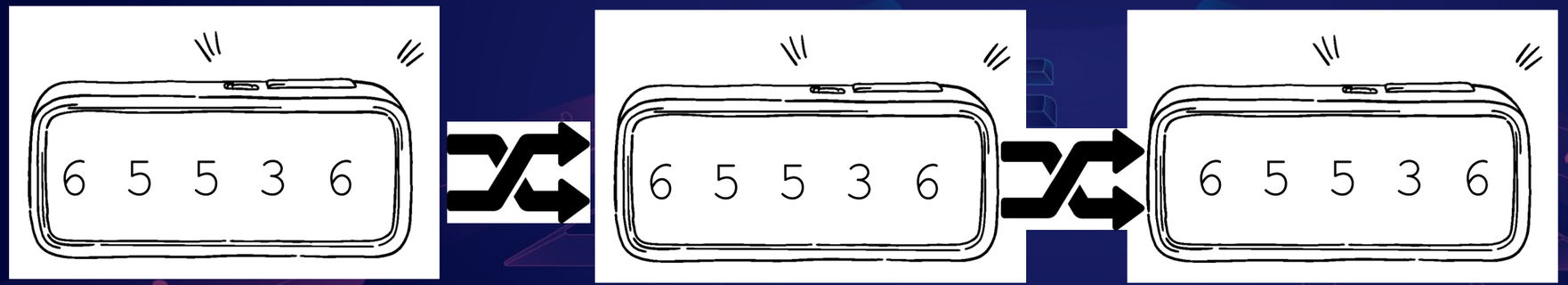
防缓存污染技术	世界	中国
端口随机化	99.33%	99.97%
TXID随机化	99.95%	99.99%
0x20编码	26.50%	17.70%
DNS Cookies	16.74%	12.62%

奇安信技术研究院司南系统2020年7月数据

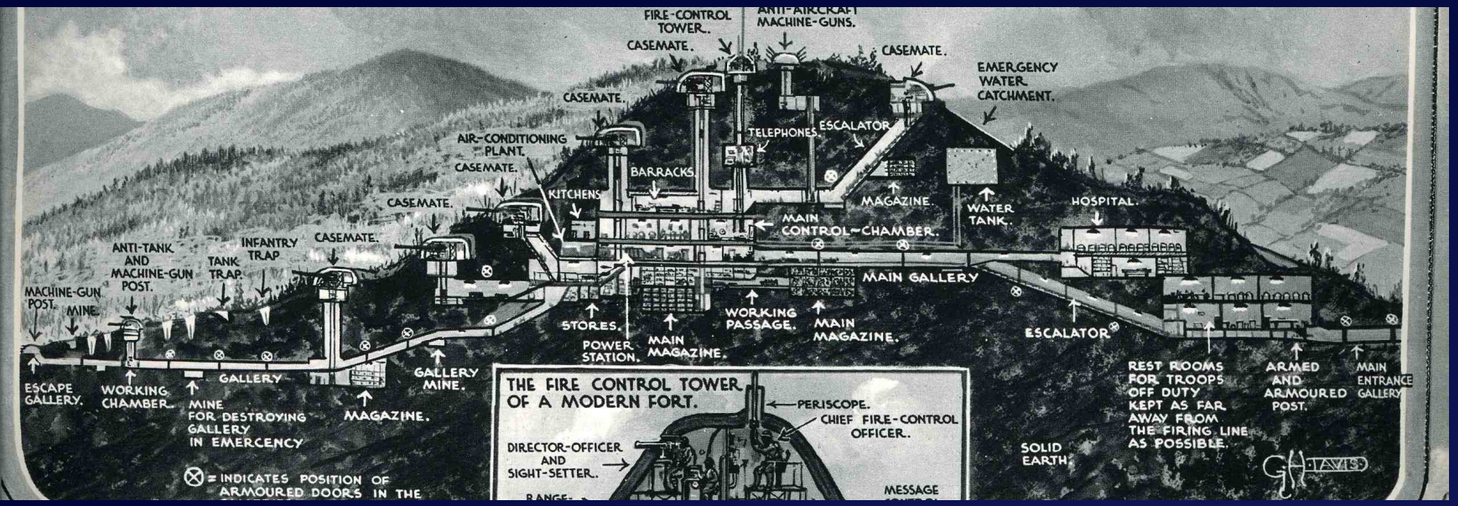
缓存防御的马奇诺防线

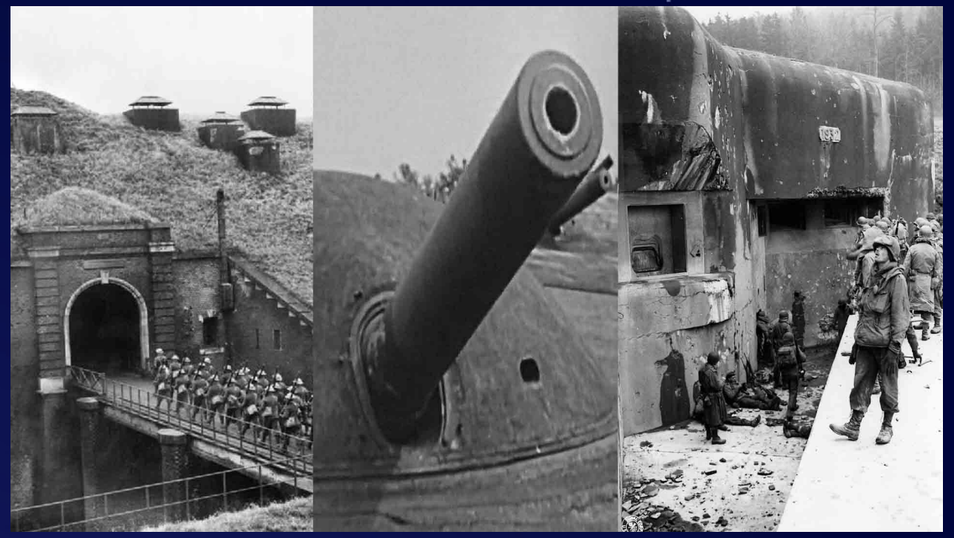
- IP ID 随机化: 16 bit
- UDP srcPort 随机化 16bit
- DNS TXID 随机化 16 bit





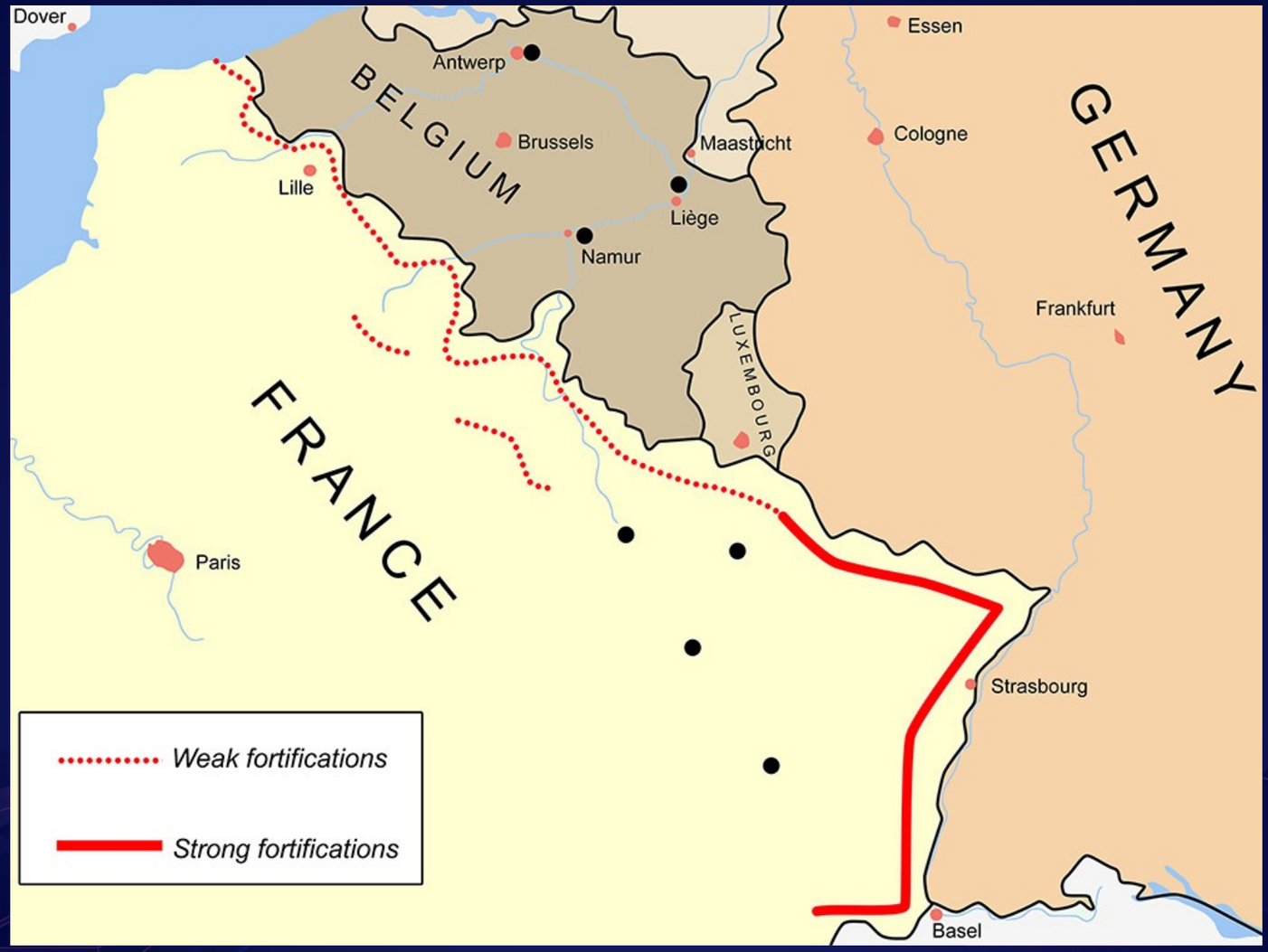
IP ID　　Source port　　TXID


FIRE-CONTROL TOWER.
ANTI-AIRCRAFT MACHINE-GUNS.
CASEMATE.
CASEMATE.
CASEMATE.
EMERGENCY WATER CATCHMENT.
TELEPHONES.
ESCALATOR.
AIR-CONDITIONING PLANT.
CASEMATE.
BARRACKS.
KITCHENS.
CASEMATE.
MAGAZINE.
WATER TANK.
HOSPITAL.
MAIN CONTROL-CHAMBER.
ANTI-TANK AND MACHINE-GUN POST.
INFANTRY TRAP
CASEMATE.
TANK TRAP.
MAIN GALLERY
MACHINE-GUN POST.
MINE.
STORES.
POWER STATION.
MAIN MAGAZINE.
WORKING PASSAGE.
MAIN MAGAZINE.
ESCALATOR.
ESCAPE GALLERY.
WORKING CHAMBER.
GALLERY
MINE FOR DESTROYING GALLERY IN EMERCENCY
MAGAZINE.
GALLERY MINE.
REST ROOMS FOR TROOPS OFF DUTY KEPT AS FAR AWAY FROM THE FIRING LINE AS POSSIBLE.
ARMED AND ARMOURED POST.
MAIN ENTRANCE GALLERY
SOLID EARTH.
⊗ = INDICATES POSITION OF ARMOURED DOORS IN THE
THE FIRE CONTROL TOWER OF A MODERN FORT.
PERISCOPE.
CHIEF FIRE-CONTROL OFFICER.
DIRECTOR-OFFICER AND SIGHT-SETTER.
MESSAGE

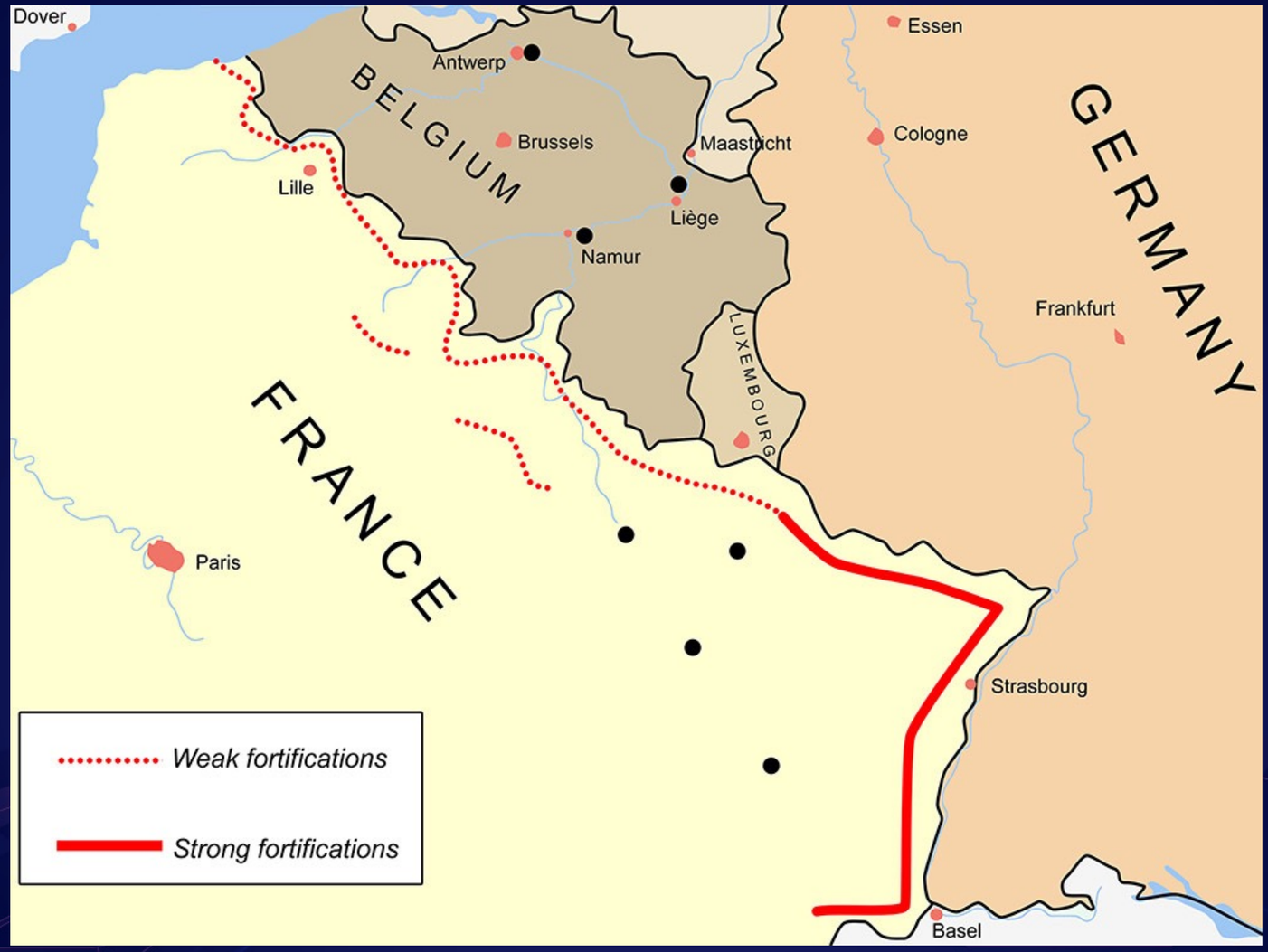



alamy - 2HXRBC0

固若金汤的马奇诺防线建好了



突破点在哪里呢?


Dover
Antwerp
BELGIUM
Brussels
Maastricht
Lille
Liège
Namur
LUXEMBOURG
Essen
Cologne
GERMANY
Frankfurt
FRANCE
Paris
Strasbourg
Basel
Weak fortifications
Strong fortifications

问题

25 年后，域名辖区原则的实现是否完美无缺?
是否有效抵御 Kashpureff 模式的域名缓存污染攻击?

MaginotDNS 利用全新的漏打破了该
“美好的愿望”。

MaginotDNS 攻击

- **概述**
  - 新型域名缓存污染攻击
  - 发表于国际网络安全顶级会议 **USENIX Security 2023**
  - **攻击目标：CDNS**（条件域名解析器）
  - **攻击效果：污染并劫持整个顶级域**，如 .com 和 .net
- **命名**
  - 利用域名缓存防护的漏洞，绕过防御的界限
  - 类似攻破马奇诺防线 → **MaginotDNS**

论文主页

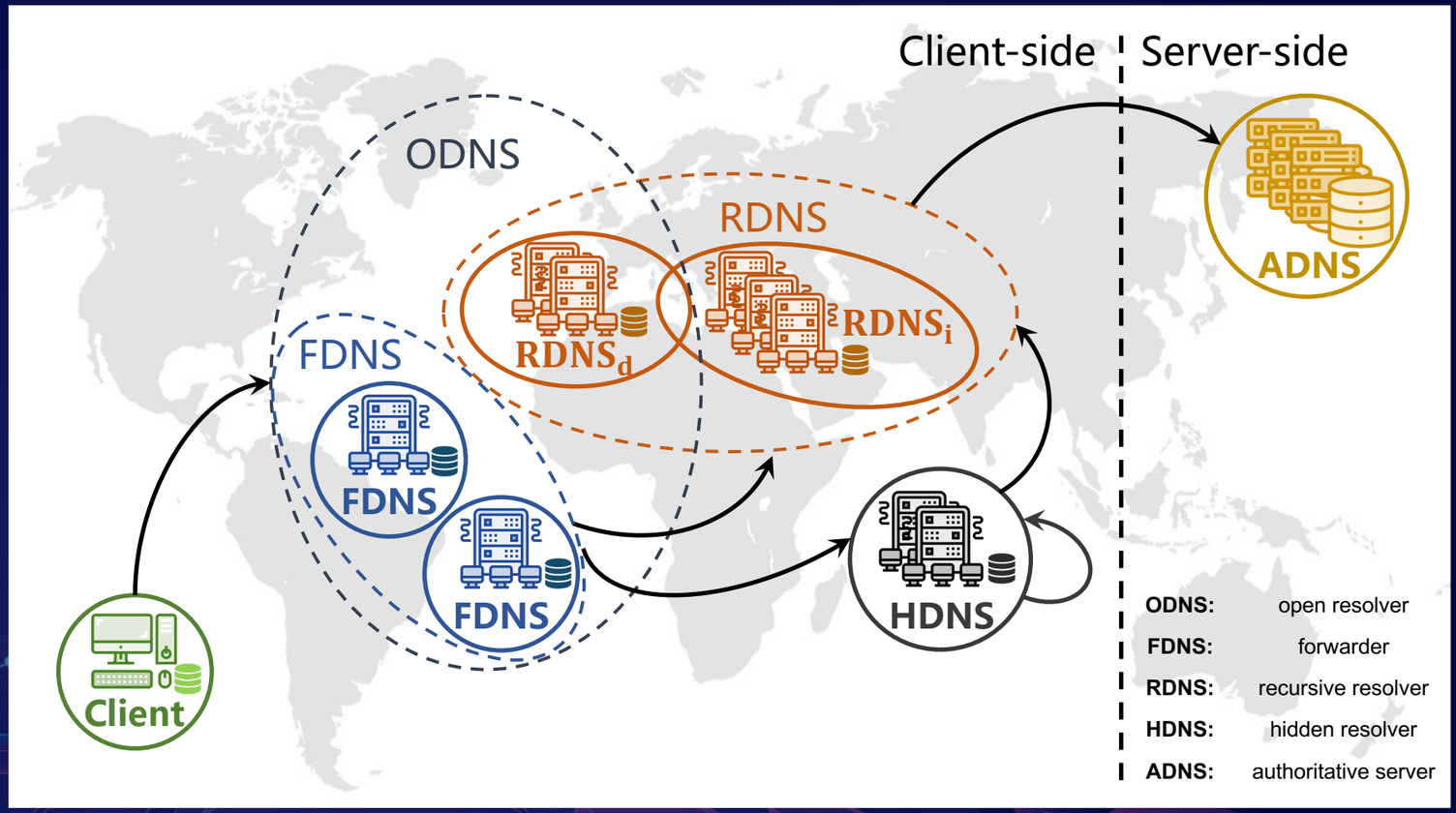
问题

什么是条件域名解析器(CDNS)（Conditional DNS）？

条件域名解析器**兼具转发查询和递归解析**功能。

域名解析器服务器生态，2013

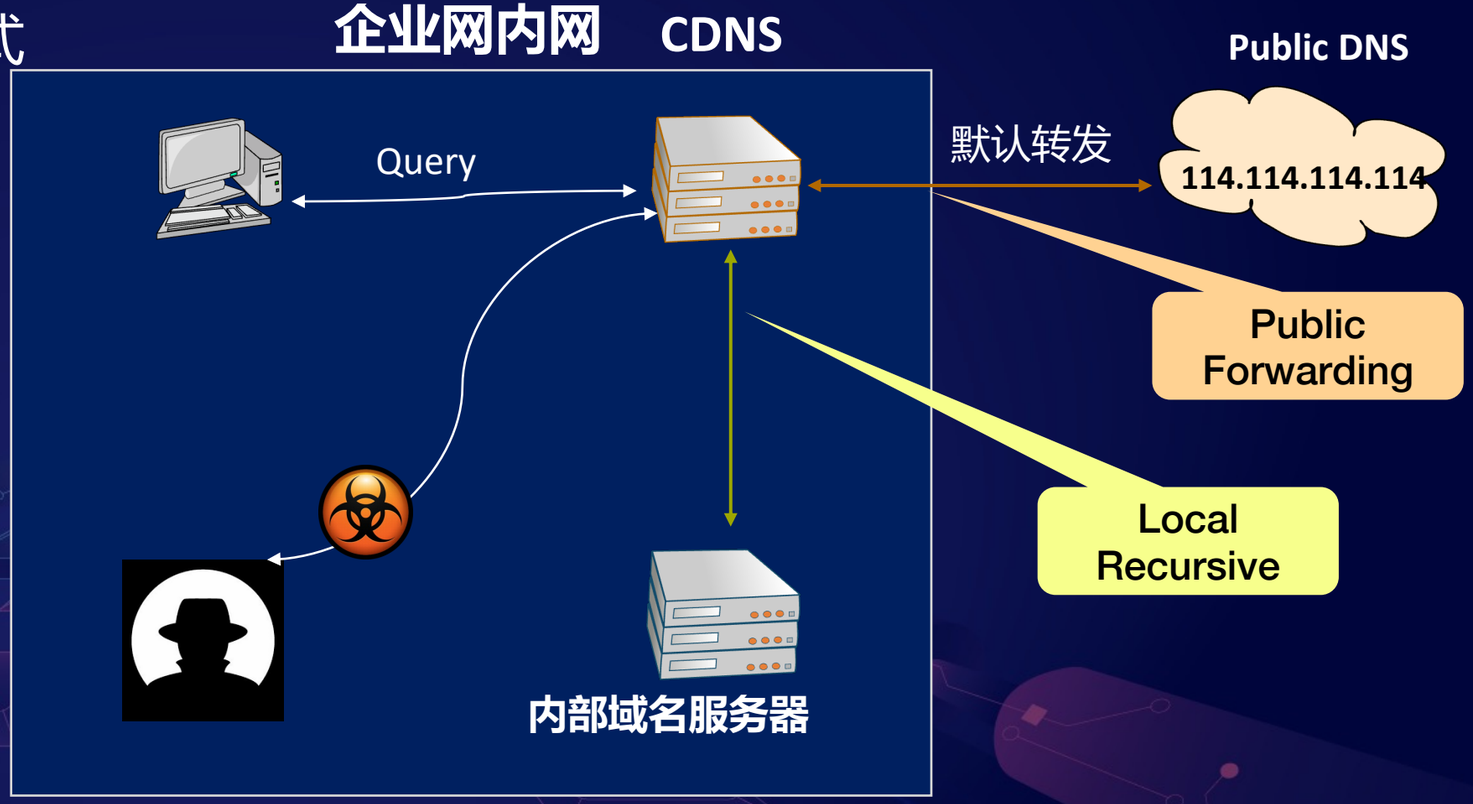
- **多个解析器角色**
  - 转发器、递归解析器
  - 客户端、权威服务器
- **组合的角色**
- **条件解析器**
  - 潜在的一种解析器
  - 并未被彻底研究过



On measuring the client-side DNS infrastructure,IMC 2013

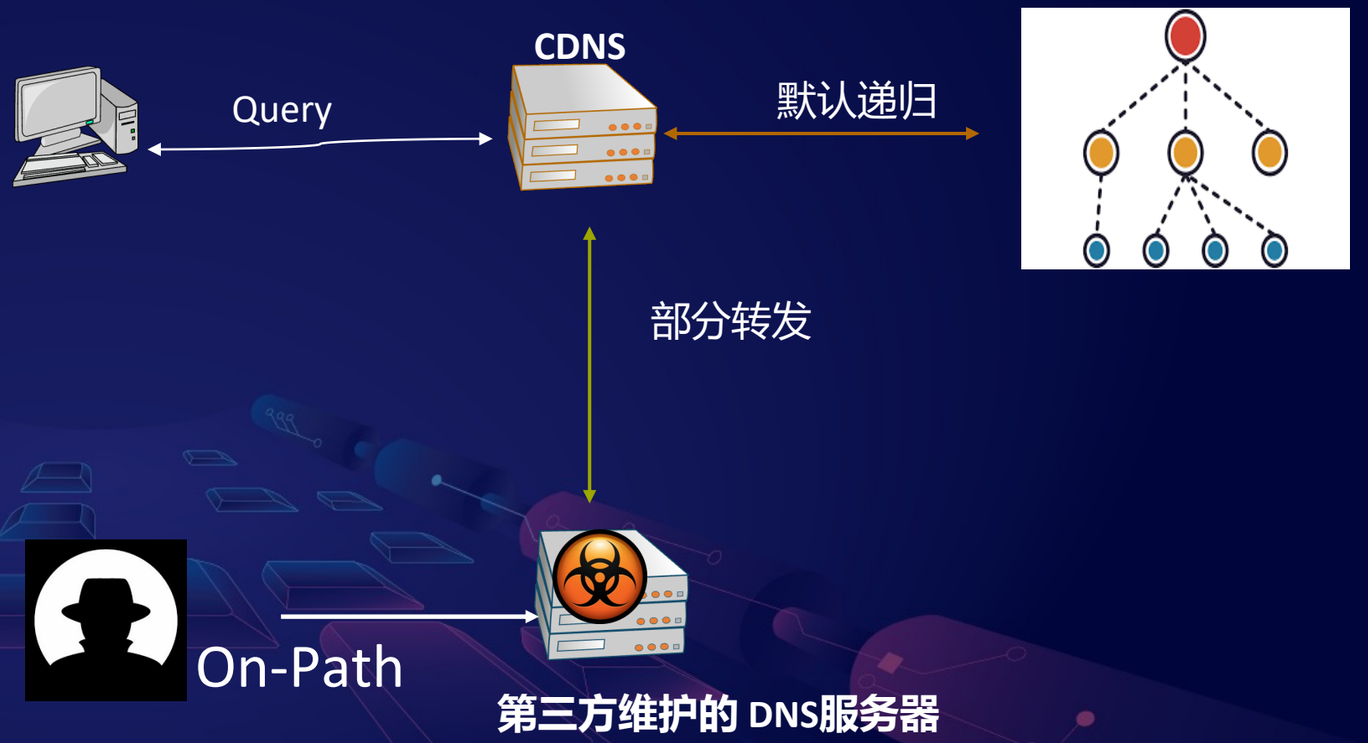
企业网场景中的CDNS 攻击模型

- **CDNS（条件域名解析器）**
  - 域名转发器 + 递归解析器（**共享缓存**）
  - 依据两个域名区域用以区分解析模式
    - Z_F：用于转发查询的域名区域
    - Z_R：用于递归查询的域名区域
- **企业内网场景**
  - 区分内外网域名解析
  - 默认转发
  - 内网域名递归，如mail.local



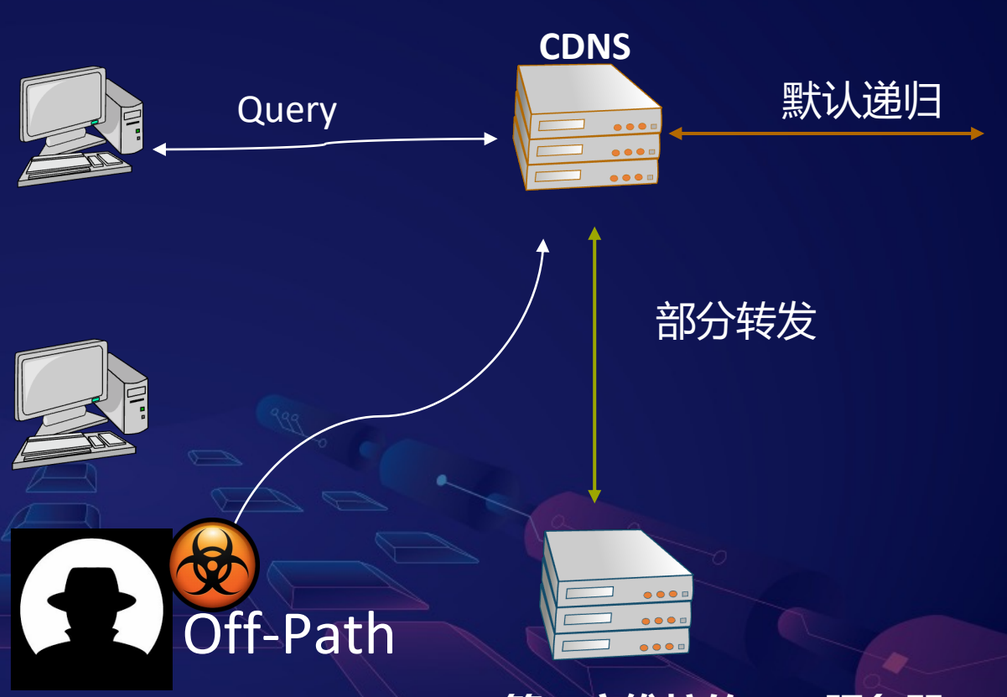
运营商 场景中的CDNS 攻击模型

- **CDNS（条件域名解析器）**
  - 域名转发器 + 递归解析器（**共享缓存**）
  - 依据两个域名区域用以区分解析模式
    - Z_F：用于转发查询的域名区域
    - Z_R：用于递归查询的域名区域
- **运营商：带宽或性能优化**
  - 性能优化：重定向到ISP内部的镜像节点
  - **带宽优化：降低ISP之间流量，节省带宽、费用**
  - 比如大流量消耗式的视频网站



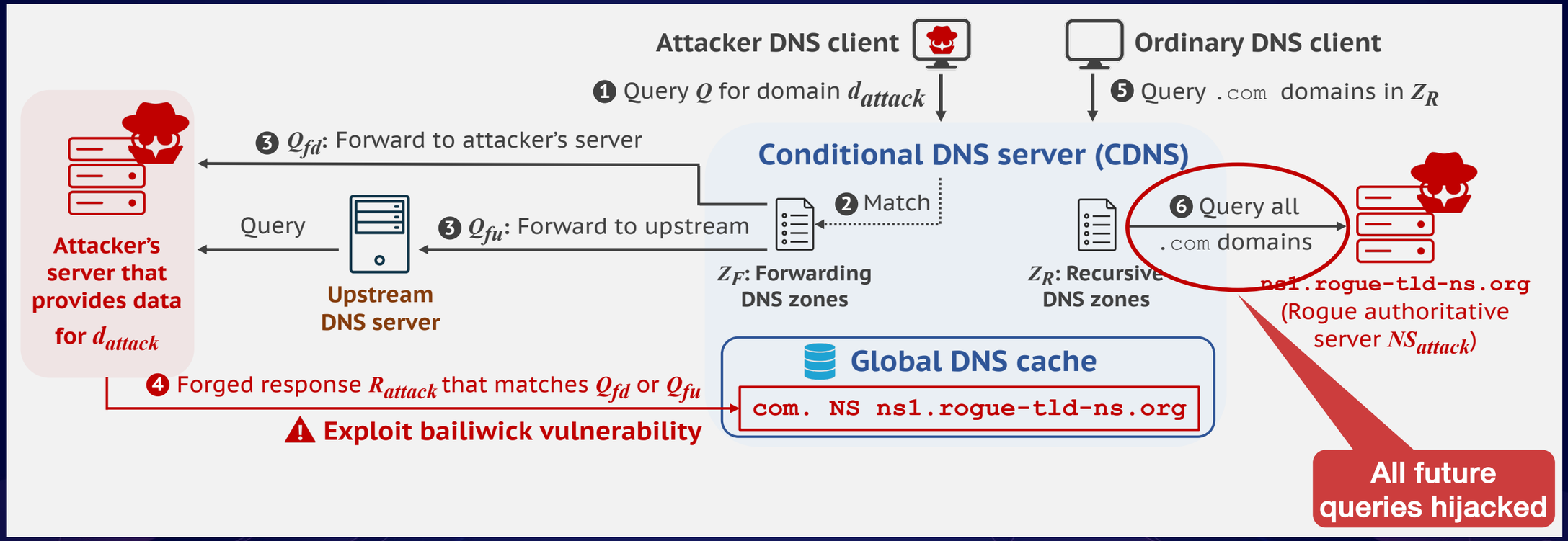
运营商 场景中的CDNS 攻击模型

- **CDNS（条件域名解析器）**
  - 域名转发器 + 递归解析器（**共享缓存**）
  - 依据两个域名区域用以区分解析模式
    - Z_F：用于转发查询的域名区域
    - Z_R：用于递归查询的域名区域
- **运营商：带宽或性能优化**
  - 性能优化：重定向到ISP内部的镜像节点
  - **带宽优化：降低ISP之间流量，节省带宽、费用**
  - 比如大流量消耗式的视频网站



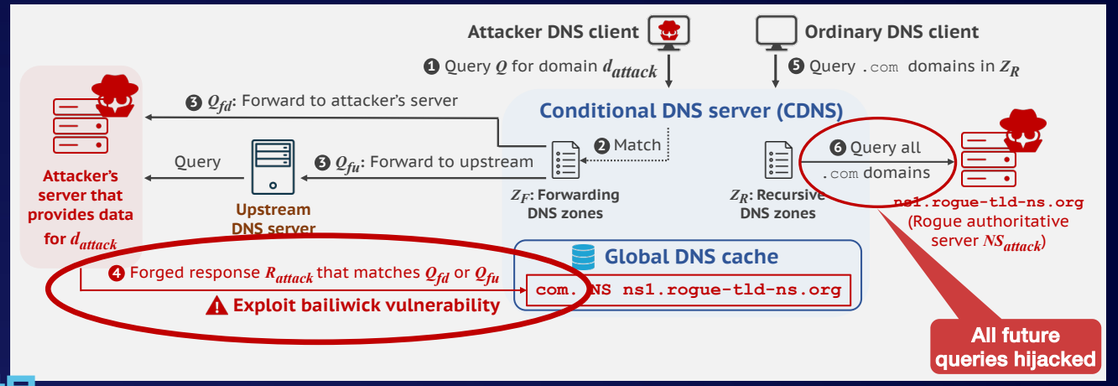
攻击概述（1/2）

- 攻击目标：可被访问的 CDNS
- 威胁模型：假设已获取 CDNS 的 Z_F & Z_R



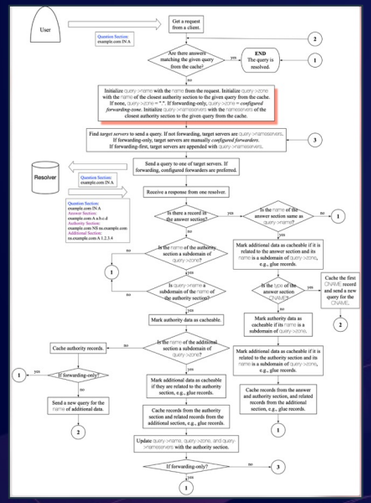
攻击概述（2/2）

- **漏洞缺陷**
  - 位于转发查询模式下
  - **接收转发查询回复中的所有内容**
- **攻击思路**
  - 递归查询模式的域名辖区原则实现**完好无损**
  - 转发查询模式的域名辖区原则实现**存在缺陷**
  - 递归查询和转发查询**共享全局缓存**
  - **利用防护不足的转发模式攻击防护严密的递归模式**
    - → **绕过域名辖区原则防护的界限，攻破域名缓存防御的马奇诺防线**

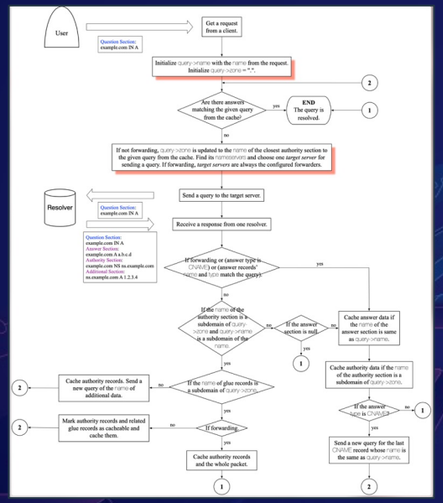


软件分析与测试

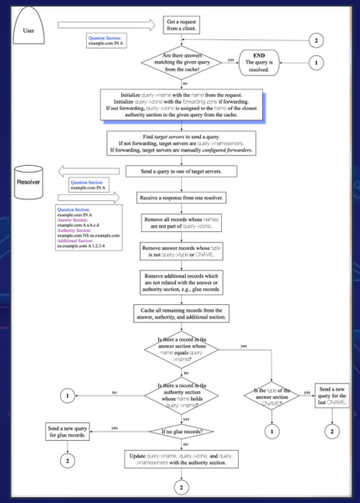
- **发现受影响的软件实现**
  - **研究思路**：开展域名辖区原则具体实现分析
  - **分析方法**：源码审阅、动态调试、灰盒测试
  - **分析目标**：**8 款主流 DNS 解析软件**，比如 **BIND** 和 **Microsoft DNS**



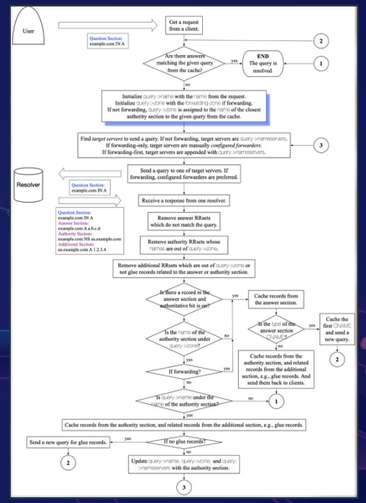
BIND



Knot



PowerDNS



Unbound

提取域名辖区原则实现的核心处理逻辑

漏洞成因 & 软件影响

- **通用型域名辖区原则**
  - 通过软件分析与测试，归纳出右图的**通用型检查逻辑**
- **漏洞成因**
  - 位于 `InitQuery` 函数：
    - `Qry.zone` **被设置为 root** →
    - 所有回复都符合域名辖区原则（均为 root 的子域名）
- **软件影响**

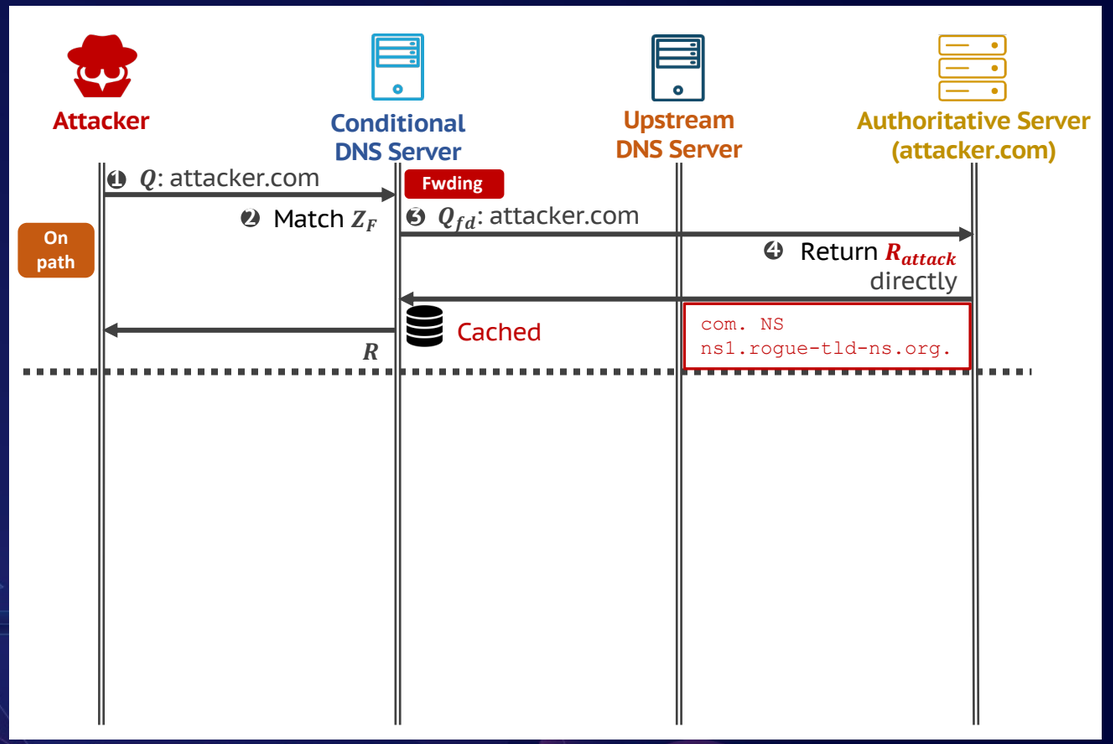
DNS Software	Forwarding	Recursive	Vulnerable
BIND9	Enabled	Enabled	**Yes**
Knot Resolver	Enabled	Enabled	**Yes**
Microsoft DNS	Enabled	Enabled	**Yes**
Technitium	Enabled	Enabled	**Yes**

Algorithm 1: DNS resolution process

```
input  : A DNS Request from clients
output : A DNS Reply to clients
1  main()
2      step_0: InitQuery(Q, Request)
3      step_1: if SeachCache(Q, Cache) then
4          goto final
5      step_2: FindServers(Q, TgtSvrs)
6      step_3: SendQuery(Q, TgtSvrs)
7      step_4: ProcessResponse(Q, R)
8          if ServerIsError(Q, R) then
9              goto step 3
10         if not MatchQuery(Q, R) then
11             goto final
12         SanitizeRecords(Q, R)
13         if IsReferral(Q, R) then
14             if not IsFwding() then
15                 UpdateQuery(Q)
16                 goto step 2
17         if IsCNAME(Q, R) then
18             UpdateQuery(Q)
19             goto step 1
20         CacheRecords(R, Cache)
21     final: ConstructReply(Reply)
22     return Reply
23 InitQuery(Q, Request)
24     initialize Q.name, Q.type, Q.zone
25     if IsFwding() then
26         ModifyFwdQuery(Q)
27 SanitizeRecords(Q, R)
28     for RR ∈ R do
29         if OutofBailiwick(RR) then
30             remove RR from R
31 UpdateQuery(Q, R)
32     update Q.name, Q.type, Q.zone
```

运营商（ISP）环境的On-path 攻击

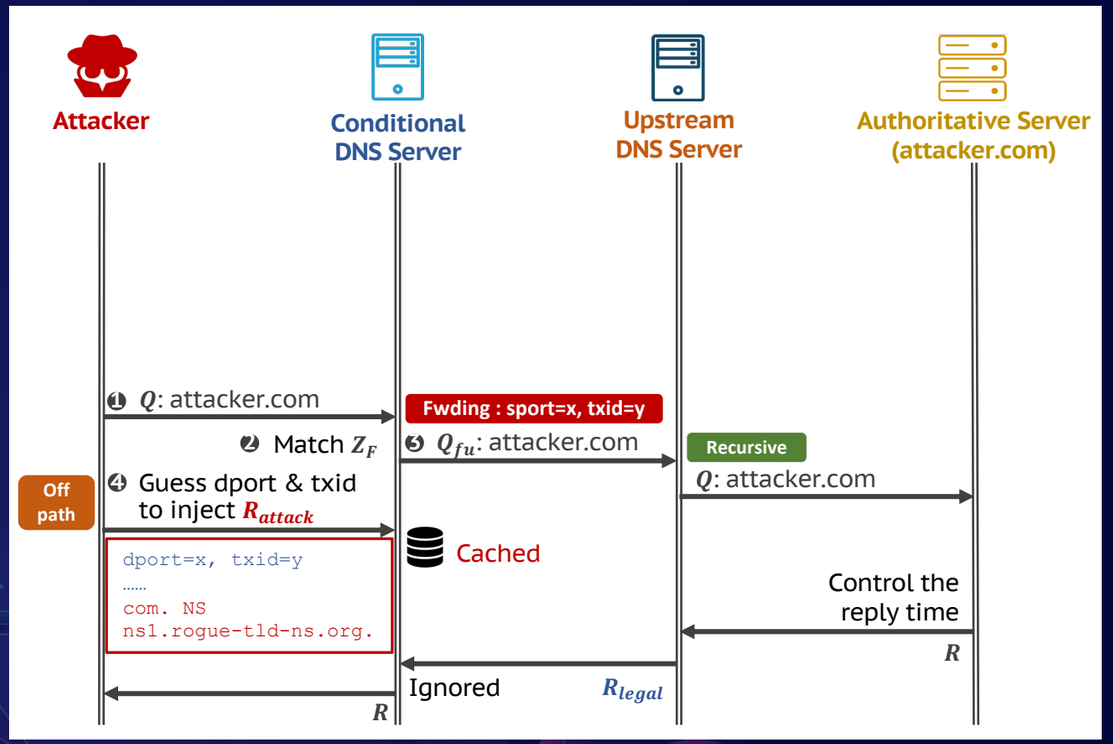
- 攻击者可以控制某个权威服务，比如与ISP合作的某视频网站
- 在权威侧直接回复虚假的响应
- **BIND、MS DNS、Knot 及 Technitium**



All future queries will be hacked.

Off-path 攻击（企业和ISP场景均适用）

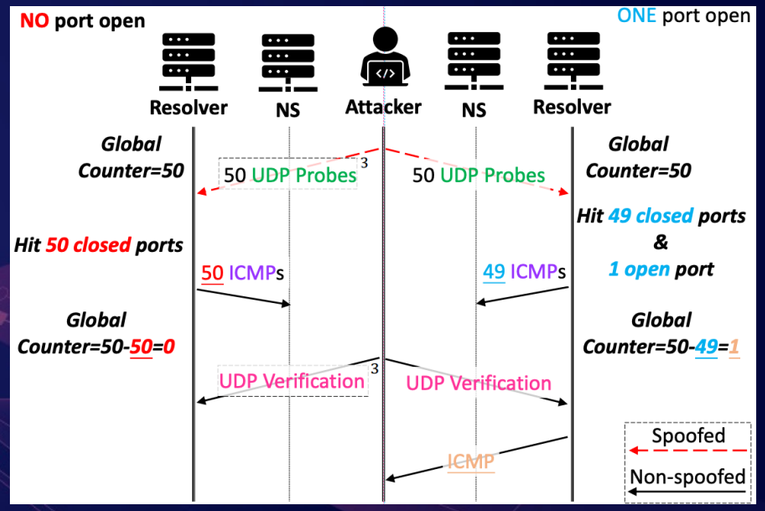
- 猜测用于验证的Source Port， TXID
- BIND9: 拓展 SADDNS 适用于 MaginotDNS
- Microsoft：全新的源端口随机化漏洞



All future queries will be hacked.

攻击验证（2/4）：Off-path 攻击 BIND9

- **猜测源端口**
  - 利用 **SADDNS** 攻击猜测源端口
  - 只有当源端口被使用时才处于开放状态，其余端口处于关闭状态
  - Linux 操作系统 **ICMP 速率限制侧信道**
- **我们的变种：生日悖论**
  - 源端口范围：32,768 - 60,999（28,232）
  - 解析超时时间：仅有 **1.2s**，多次尝试，**每轮猜测 50 个端口**
  - 3,600 轮后成功率（4320秒，1.2小时）：
    - $1-[(28,232-50)/28,232]^{3,600}=99.8\%$
- **暴力枚举 TXID： 65535， 只需100 ms**



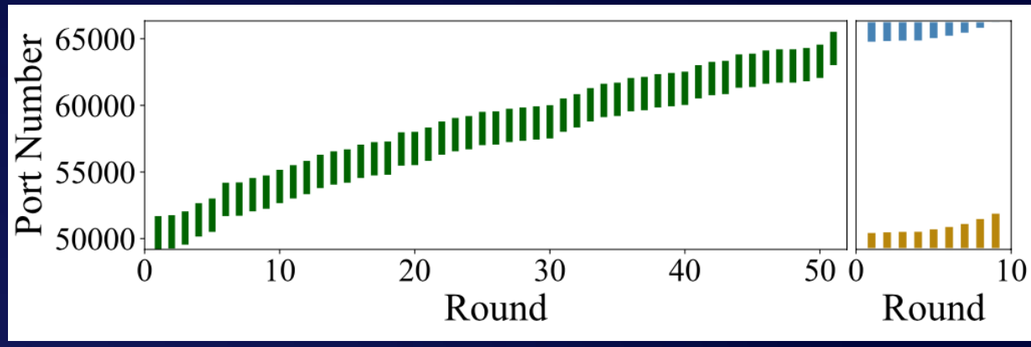
SADDNS

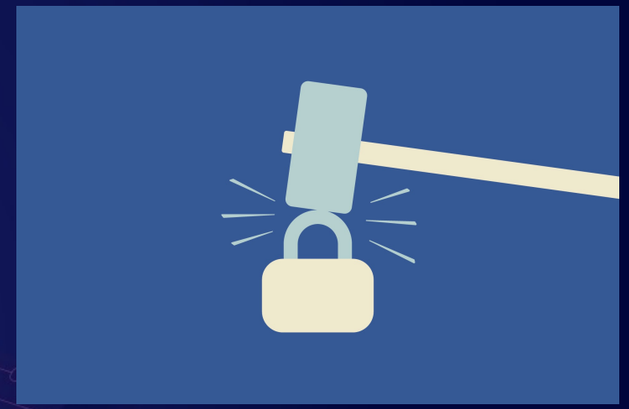
https://www.saddns.net/

Off-path 攻击 MS DNS

- **猜测源端口**
  - 经过实验，我们发现 **MS DNS 仅使用 ~2,500 个源端口进行解析**
  - 2,500 个端口均处于开放状态（SADDNS 无法奏效）
  - **可以直接暴力枚举 2,500 个端口**
- **我们的方案**
  - 源端口范围：预先探测 2,500 个端口的范围
  - 解析超时时间：**5s**（充足），多次尝试，**每轮猜测 20 个端口**
  - 1小时720 轮后成功率：$1-[(2,500-20)/2,500]^{720}=99.7\%$
- **暴力枚举 TXID**

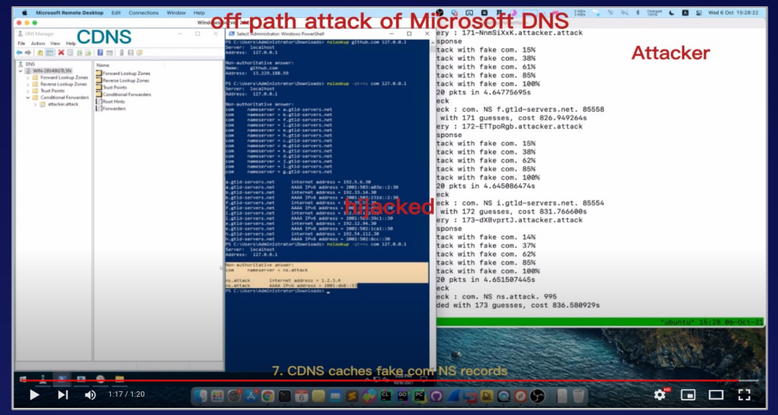
Microsoft DNS源端口范围示例





攻击验证（4/4）

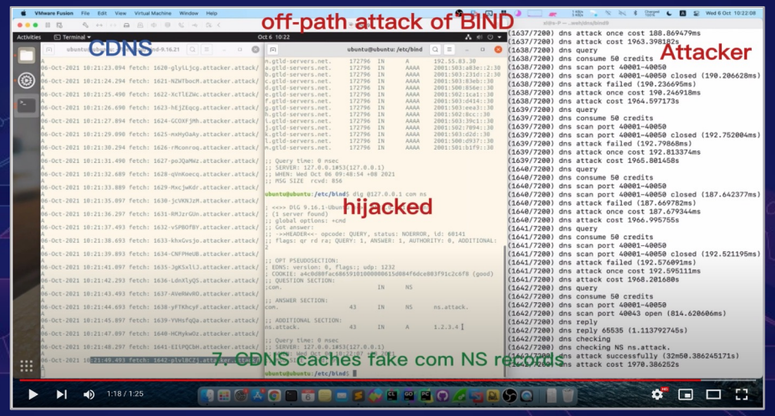
- **On-path 攻击**
  - 100% 成功率
- **Off-path 攻击**
  - Microsoft DNS：**平均 802s**
  - BIND9：**平均 790s**




```
Mon Aug  9 03:31:01 2021 : (2/360) dns query : 2-BatHkHSX.idealeer.com
Mon Aug  9 03:31:01 2021 : (2/360) dns response
Mon Aug  9 03:31:03 2021 : (2/360) dns attack with fake com. 15%
Mon Aug  9 03:31:04 2021 : (2/360) dns attack with fake com. 37%
Mon Aug  9 03:31:05 2021 : (2/360) dns attack with fake com. 60%
Mon Aug  9 03:31:06 2021 : (2/360) dns attack with fake com. 85%
Mon Aug  9 03:31:06 2021 : (2/360) dns attack with fake com. 100%
Mon Aug  9 03:31:06 2021 : to 202.112.238.57 : 1310720 pkts in 4.632276358s
Mon Aug  9 03:31:06 2021 : (2/360) dns check
Mon Aug  9 03:31:06 2021 : (2/360) dns check : com. NS gtld-servers.attack.
Mon Aug  9 03:31:06 2021 : dns attack succeeded with 2 guesses, cost 10.079395433s
```

Microsoft DNS 攻击日志

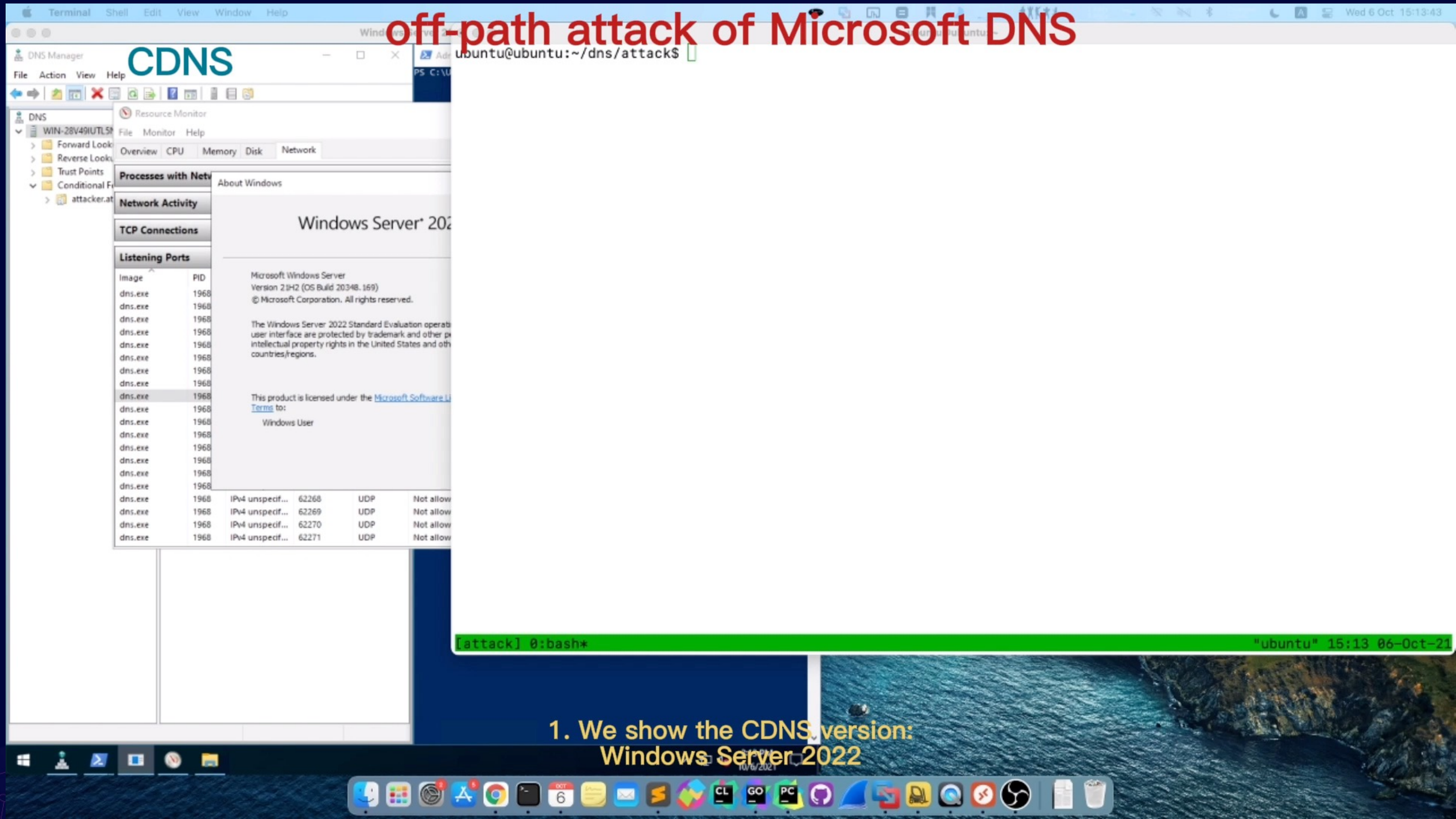
Watch videos here.




```
Thu Aug 26 23:10:53 2021 : (661/3600) dns querying
Thu Aug 26 23:10:53 2021 : (661/3600) dns consuming 50 credits
Thu Aug 26 23:10:53 2021 : (661/3600) dns scanning port 40001-40050
Thu Aug 26 23:10:54 2021 : (661/3600) dns scanning port 40020 open (651.902104ms)
Thu Aug 26 23:10:54 2021 : (661/3600) dns replying
Thu Aug 26 23:10:54 2021 : (661/3600) dns replying 65535 (928.938966ms)
Thu Aug 26 23:10:54 2021 : (661/3600) dns checking
Thu Aug 26 23:10:54 2021 : (661/3600) dns checking NS gtld-servers.attack.
Thu Aug 26 23:10:54 2021 : (661/3600) dns attack successfully (13m12.992182401s)
Thu Aug 26 23:10:54 2021 : (661/3600) dns attack cost (13m12.99219492s)
```

BIND9 攻击日志

off-path attack of Microsoft DNS

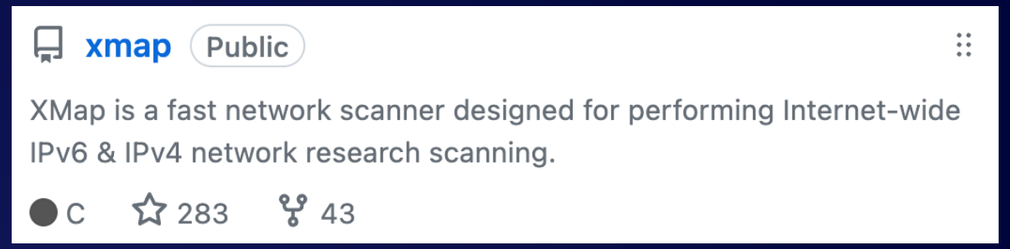


只是实验室里的攻击吗?

如何找到互联网上的条件域名服务器CDNS呢?

CDNS 规模可能超乎你的想象

- **使用自开发的 XMap 探测受影响的 CDNS**
  - **测量目标：** 1.2M 公网 DNS 解析器
  - **过滤条件：** 移除不适合测量的解析器
  - **测量方法：** **提出了新颖的方法**，发现了 **154,955 个 CDNS**
  - **软件识别：** 使用 DNS 软件指纹，识别到了 **54,949 个受影响的 CDNS**
    - 排除开启 DNSSEC 或 0x20 安全防护措施的解析器



CDNSes identified by probing	**154,955**	**41.8%**
– Version identifiable (in CDNS)	117,306	31.7%
– by `version.bind`	59,419	16.0%
– by `fpdns`	57,887	15.6%
– OS identified for BIND (in CDNS)	19,995	5.4%
– DNSSEC validation (in CDNS)	34,424	9.3%
– 0x20 encoding (in CDNS)	1,119	0.3%

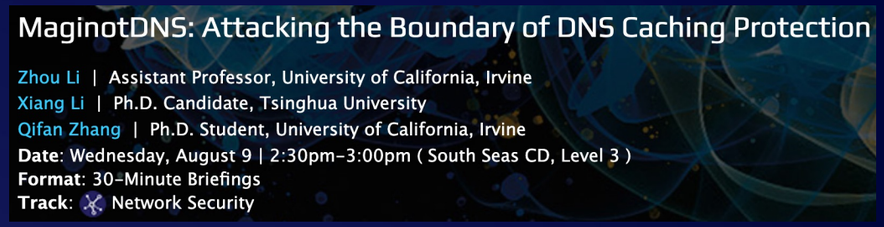
Vulnerable CDNSes	**54,949**	**14.8%**
– On-path attack possible*	**54,949**	**14.8%**
– BIND	24,287	6.6%
– Microsoft DNS	30,662	8.3%
– Off-path attack possible*	**48,539**	**13.1%**
– BIND (OS exploitable)	17,877	4.8%
– Microsoft DNS	30,662	8.3%
– Recursive-default	10,445	5.0%
– Forwarding-default	36,581	9.9%

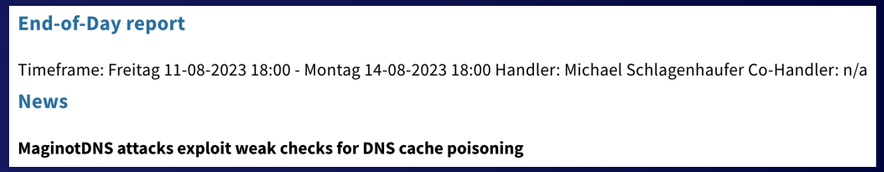
漏洞披露 & 缓解措施

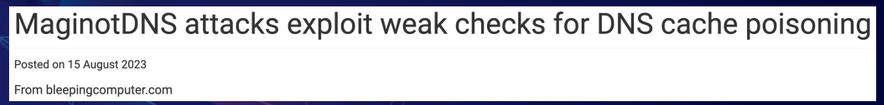
- **漏洞披露**
  - 受影响的厂商均已**承认并修复了该漏洞**，包括 BIND9、Knot、Microsoft 及 Technitium
  - 该漏洞共获批 **4 个 CVE 安全漏洞编号**，获得了微软的漏洞奖励
- **缓存措施**
  - **方式：** `Qry.zone` 应被配置为“被转发的域名”，而不是“root”
  - **效果：** 只有被转发域名的资源记录可以被接收
  - **状态：** 该方案**已被受影响的厂商所采纳应用**
  - **用户：** 应该及时更新旧版本的软件
  - **其它方案：** DNSSEC、0x20 等

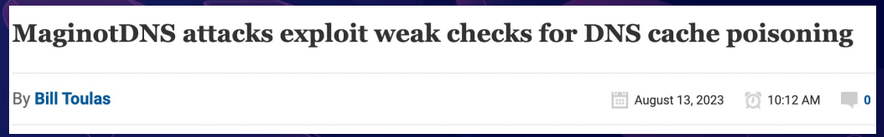
现实影响

- **工业界影响**
  - 在工业界安全顶会 Black Hat USA 2023 展示
- **政府大学影响**
  - 奥地利政府 CERT 每日安全公告
  - 瑞典政府 CERT 每周安全公告
  - 伯恩茅斯大学 CERT 安全公告
- **60+ 科技媒体报道**
  - 比如 BleepingComputer
- **APNIC 域名注册局博客约稿**
- **数字寰宇“大家讲堂”讲座邀请**









结语

- **全新的威胁模型**
  - 提出了一种长久被忽略的解析器角色：CDNS 及威胁模型
- **全新的安全漏洞及攻击方法**
  - 位于共存的解析模式和共享的缓存下
  - 古老的域名辖区原则机制，在众多软件中依旧存在重大安全缺陷
  - 提出新的攻击方式，用于利用该漏洞发起威力更大的域名缓存污染攻击
- **全新的 CDNS 发现方法及结果**
  - 提出了一种新颖的 CDNS 发现方法
  - 发现大规模受影响的真实 CDNS 解析器

预告

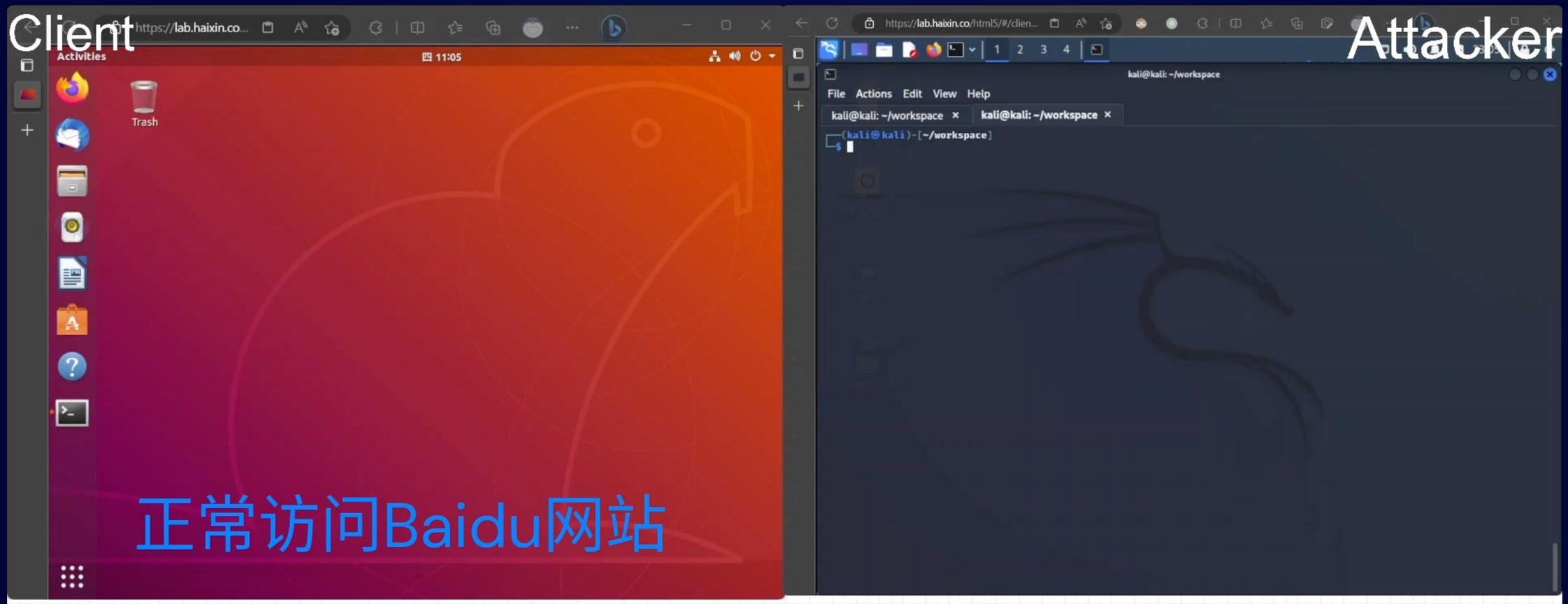
什么？该攻击还是太慢了？谁想要在 1s 内完成攻击？

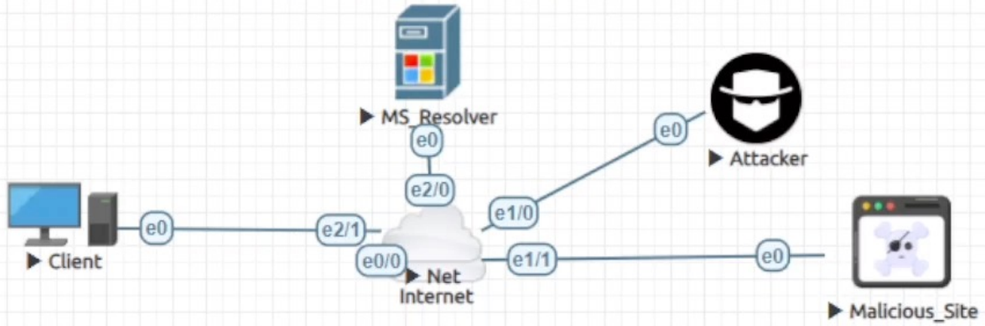
我们最新发表在国际网络安全顶级会议 **IEEE S&P 2024**
的**突门攻击**论文带来了你想要的，请看！

突门攻击：1s 内缓存污染 MS DNS

Client

Attacker





域名缓存污染攻击的前世今生

1997 Kashpureff Attack

2002 Birthday Attack

2008 Kaminsky Attack

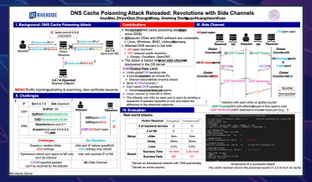
2013 Fragmentation Attack

2020 Attack on Forwarders

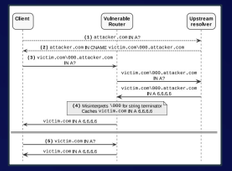
2020 SADDNS Attack

2021 Attack via Escaped Chars

2021 SADDNS v2 Attack



2022 Attack via Escaped Chars v2



2023 MaginotDNS Attack

突门攻击 TuDoor DNS Attack

IEEE S&P 2024