32ND USENIX SECURITY SYMPOSIUM

The Maginot Line: Attacking the Boundary of DNS Caching Protection

Xiang Li, Chaoyi Lu, Baojun Liu, Qifan Zhang,

Zhou Li, Haixin Duan, and Qi Li

Presenter: **Xiang Li**, Tsinghua University

August 2023

Attack Impact

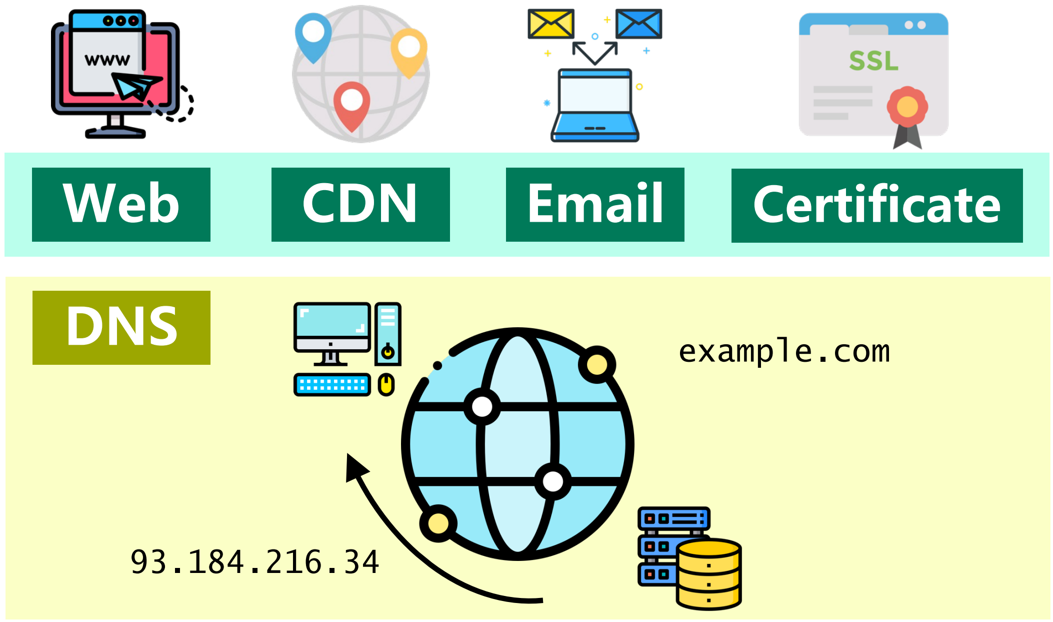
Our MaginotDNS attack could poison a whole TLD, e.g., .com and .net, at a time.

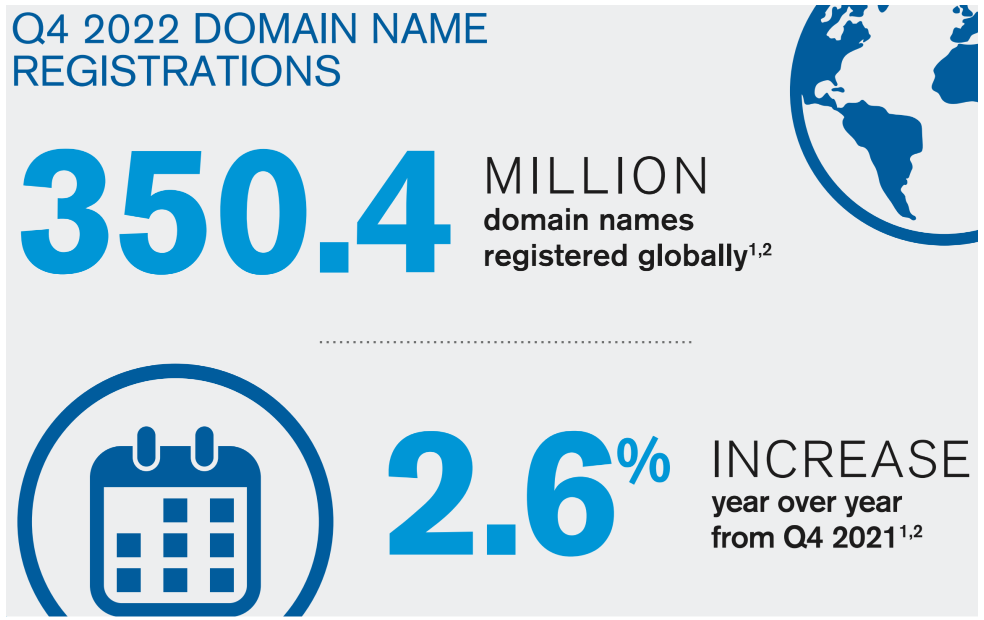
Thus, all domains under that TLD can be hijacked.

Domain Name System (DNS)

➢ DNS Overview

- ❑ Translating domain names to IP addresses
- ❑ Entry point of many Internet activities
- ❑ Domain names are widely registered



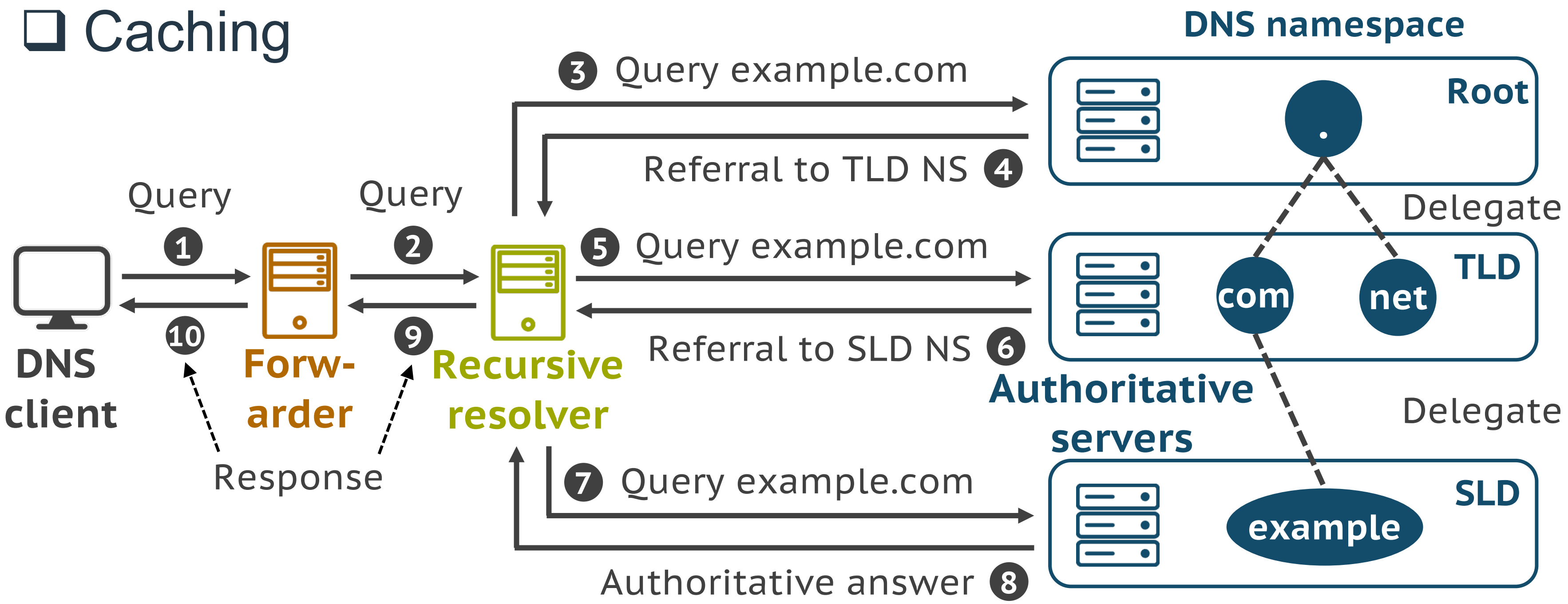


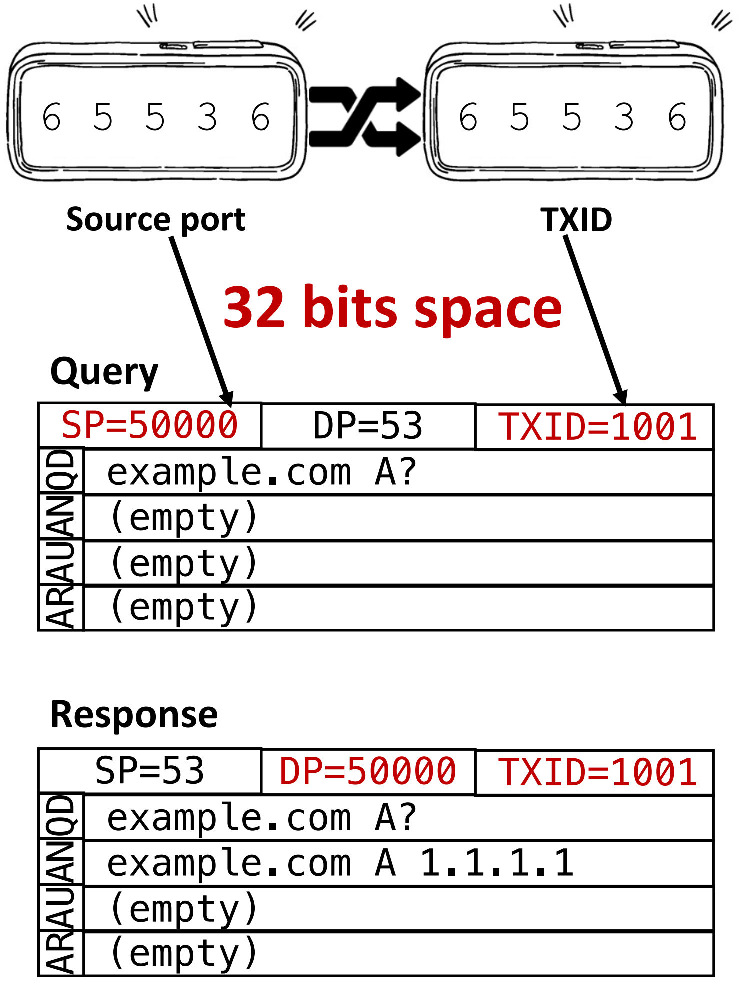
verisign.com/dnib

Domain Name System (DNS)

➢ DNS Resolution Process

- ❑ Primarily over UDP
- ❑ Iterative and recursive
- ❑ Caching





Takeaway

Since DNS is the cornerstone of the Internet, enabling multiple critical services and applications,

Attackers have long been trying to manipulate its response for hijacking via **cache poisoning attacks**.

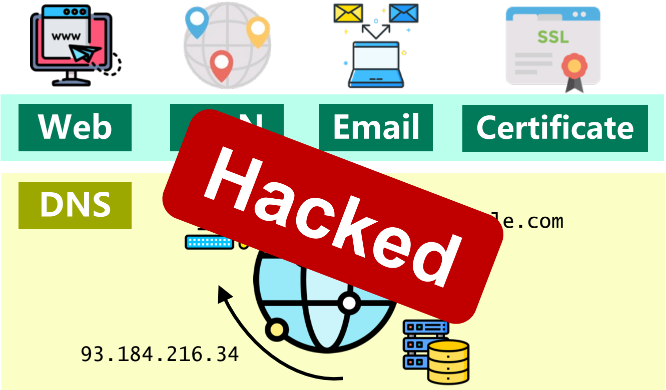
Question

What is DNS cache poisoning?

Since DNS is primarily over UDP, attackers want to **inject forged answers into resolvers' cache**.

DNS Cache Poisoning

- **Target**
  - ❑ Injecting forged answers into resolvers' cache
- **Taxonomy**
  - ❑ On-path, off-path
- **Technique**
  - ❑ Cat-and-mouse game



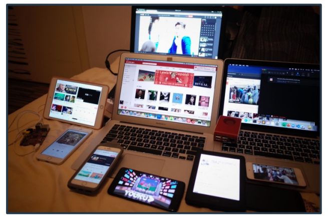
1997 Kashpureff Attack

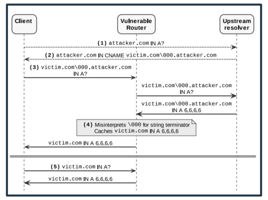
2002 Birthday Attack

2008 Kaminsky Attack

2013 Fragmentation Attack

2020 Attack on Forwarders

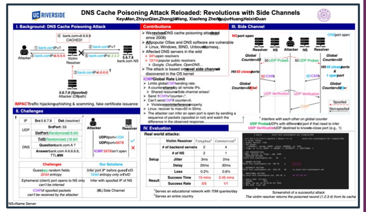




2020 SADDNS Attack

2021 Attack via Escaped Chars

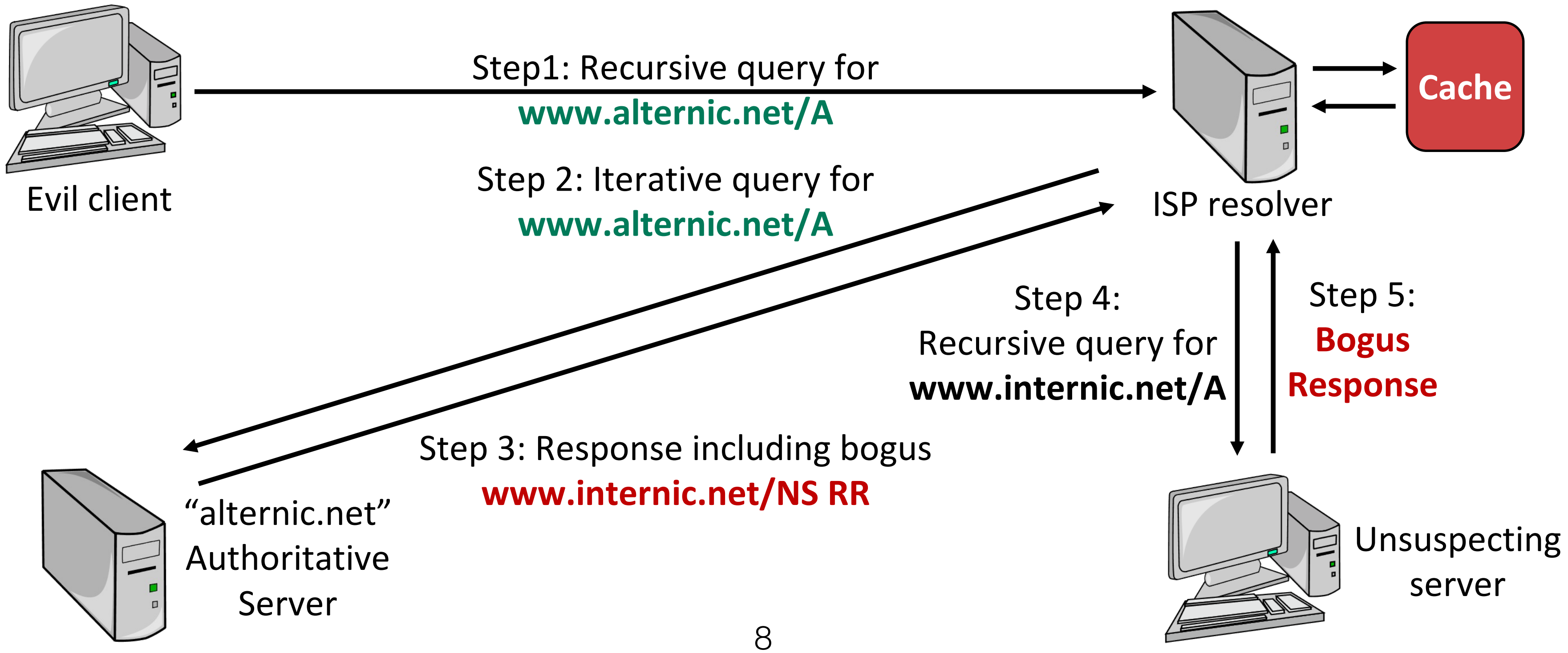
2021 SADDNS v2 Attack



2022 Attack via Escaped Chars v2

DNS Cache Poisoning

- **Kashpureff Attack (on-path, 1997)**
  - ❑ Method: returning forged responses from the authoritative
  - ❑ Result: resolver accepting all records in the response
  - ❑ Cause: lacking data verification (**bailiwick rules**)



DNS Bailiwick Rules

- **Mitigating the Kashpureff Attack**
  - The credibility checking when storing cache entries
  - Checking for "**in bailiwick**" in response data: **answer records must be from the same domain as the requested name**

```
$ dig example.com

;; ANSWER SECTION:
example.com. 86400 IN A 93.184.216.34

;; AUTHORITY SECTION:
mybank.com. 86400 IN NS ns.mybank.com.

;; ADDITIONAL SECTION:
ns.mybank.com. 86400 IN A 1.2.3.4
```

Bailiwick

In-bailiwick
Can be trusted

Out-of-bailiwick
Should be removed

Takeaway

After the Kashpureff attack, bailiwick checking is integrated into the resolver's implementation,

DNS cache poisoning on recursives from the on-path seems **impossible** to conduct from 1997.

Question

26 years later, does bailiwick checking work as desired after fixing the Kashpureff attack?

No. **MaginotDNS** breaks this guarantee with a new powerful **cache poisoning vulnerability**.

MaginotDNS Attack

- **What is the MaginotDNS attack**
  - Proposed by our **NISL** lab
  - A new powerful DNS cache poisoning attack against **CDNS resolvers**
  - Can be launched from either **on-path** or **off-path**
  - Can poison **arbitrary domains** including **TLDs**, such as .com and .net
- **Name**
  - Exploiting **vulnerabilities** of bailiwick checking to bypass itself
  - Working like breaking the **Maginot Line** → **MaginotDNS**

DNS
MAGINOT

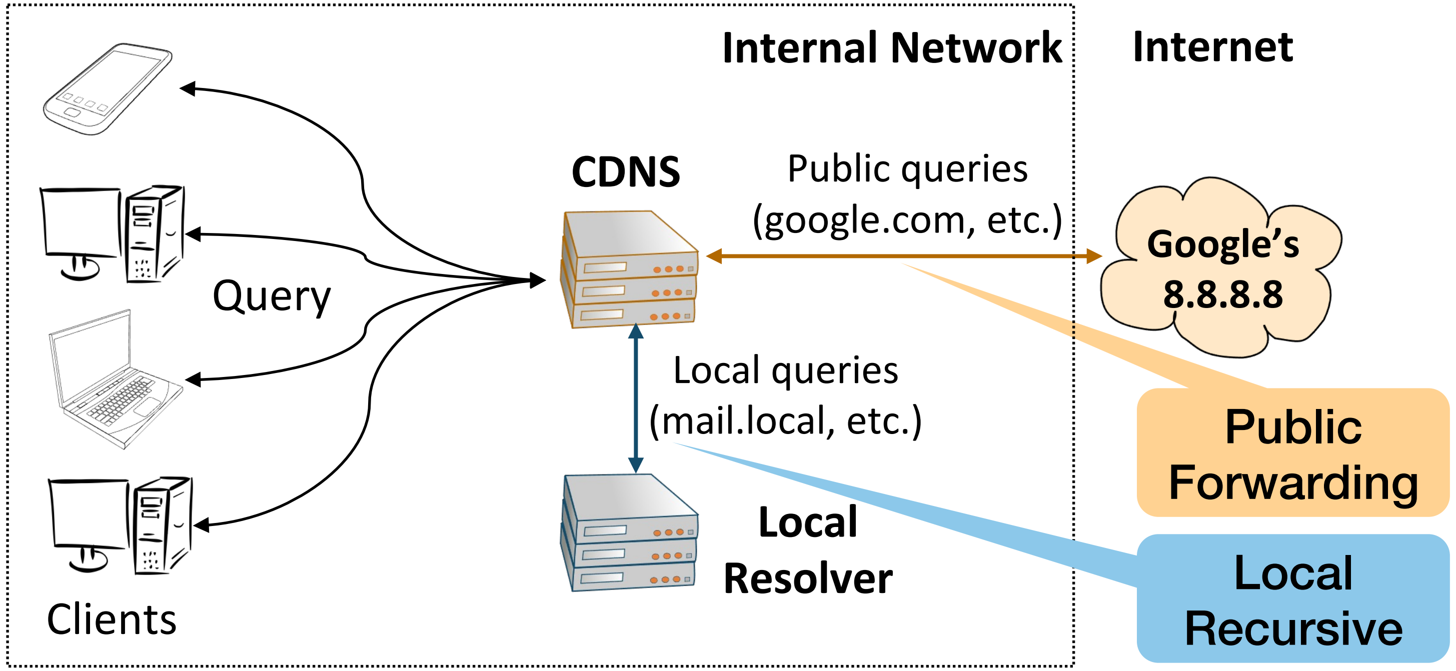
Question

What is the CDNS resolver?

A **conditional DNS resolver** with both **recursive** and **forwarding** query modes.

Attack Target: CDNS

- **Conditional DNS Resolver (CDNS)**
  - Forwarder + recursive resolver (shared cache)
  - 2 query zones used for different resolution
    - Z_F: domains for forwarding queries
    - Z_R: domains for recursive queries
- **Usage Scenarios**
  - Enterprise: splitting networks
  - ISP: reducing heavy traffic cost
  - (video-style domains)



Attack Overview of MaginotDNS

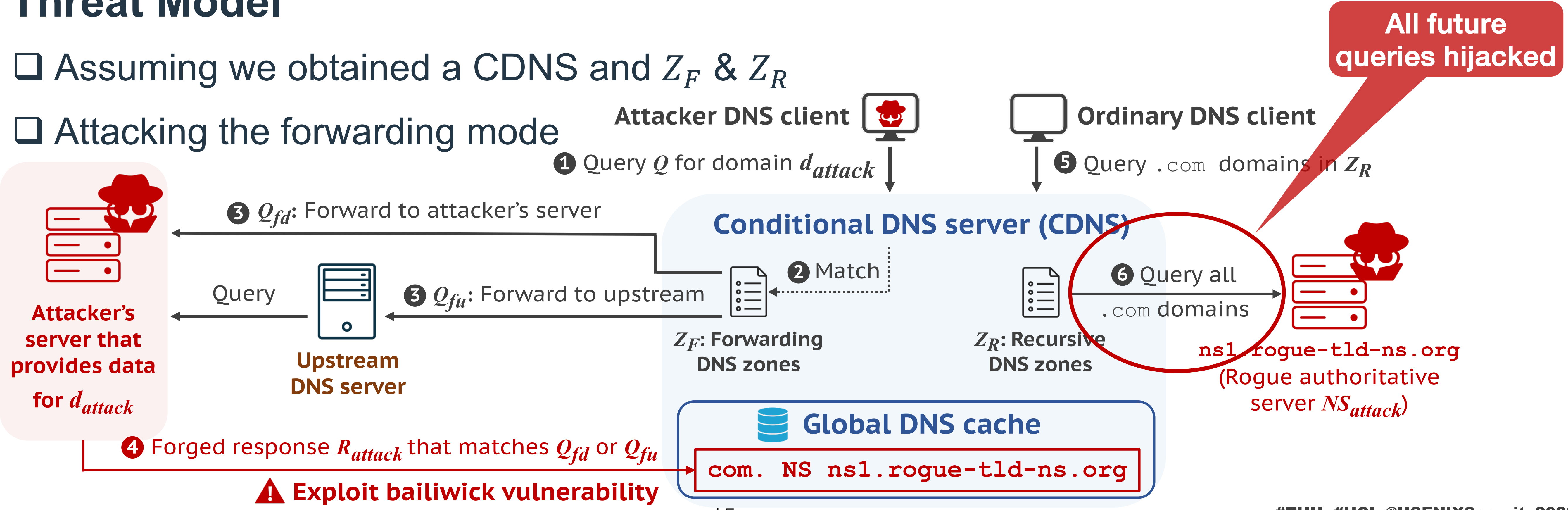
➢ Attack Target

❑ CDNS that can be accessed

➢ Threat Model

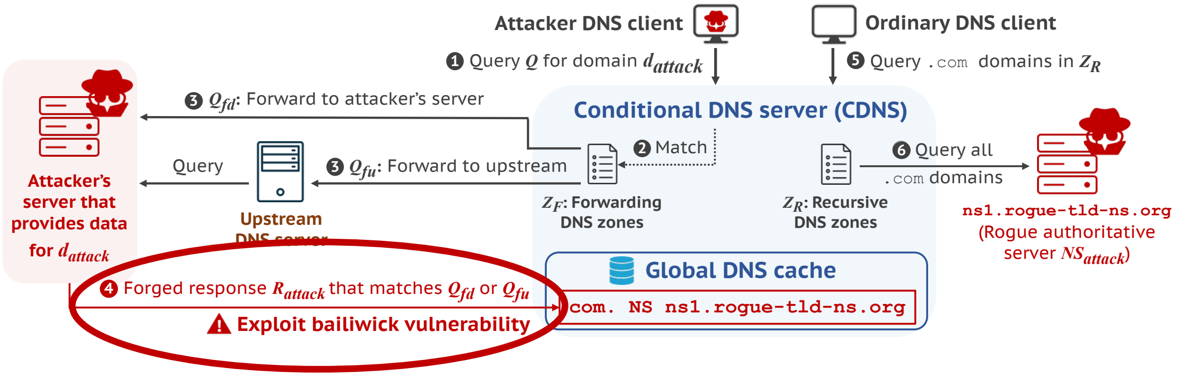
❑ Assuming we obtained a CDNS and Z_F & Z_R

❑ Attacking the forwarding mode



Attack Overview of MaginotDNS

- **Bailiwick Checking Vulnerability**
  - In the forwarding mode
  - **Accepting all records in a forwarding res.**

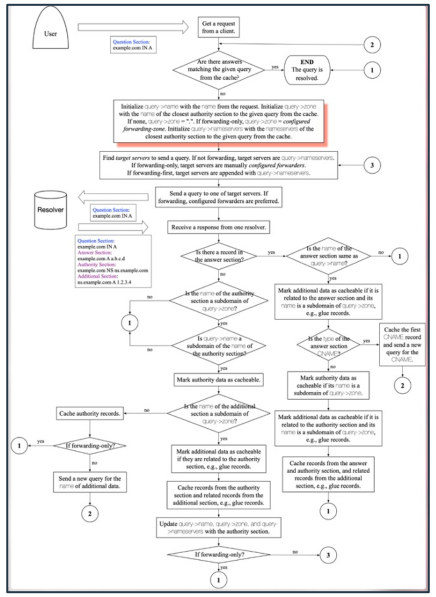


- **Exploiting Idea**
  - Bailiwick checking of the recursive mode is **well implemented**
  - But the **forwarding** mode is not.
  - Since they share the **same global DNS cache**
  - We can **exploit the weak forwarder** to attack the well-protected recursive
    - → **Breaking the boundary of DNS caching protection**

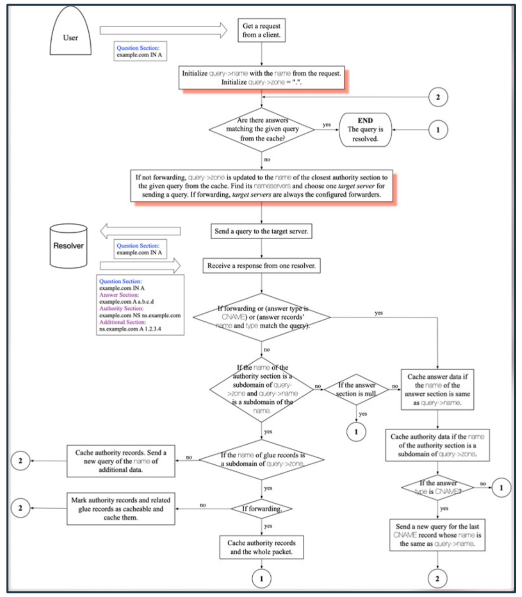
Software Analysis

➢ Finding Vulnerable Software

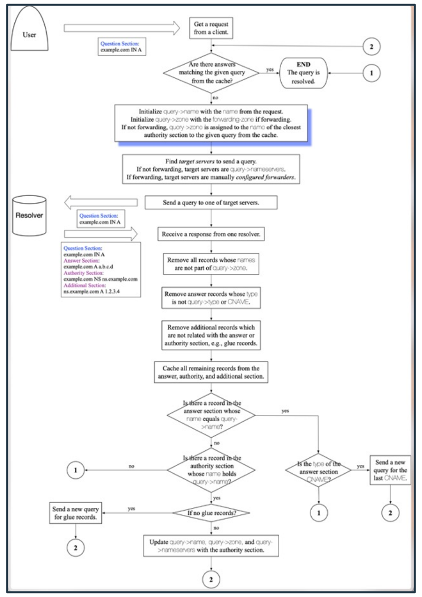
- ❑ In depth **bailiwick checking implementation** analysis
- ❑ Via source code review, debugging, and testing
- ❑ 8 mainstream DNS software, e.g., BIND and Microsoft DNS



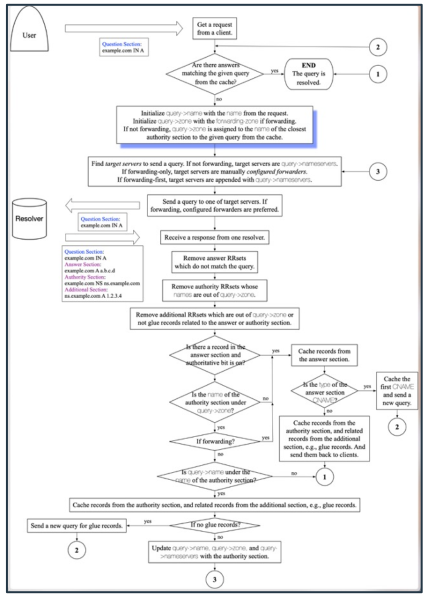
BIND



Knot



PowerDNS



Unbound

Extracting bailiwick checking implementations

Root Cause & Vulnerable Software

- **General Bailiwick Checking Logic**
  - Summarized by us
- **Root Cause**
  - In the `InitQuery` function:
    - `Qry.zone` is set to root → all records is **in-bailiwick** (root's subdomains)
- **Vulnerable Software**

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

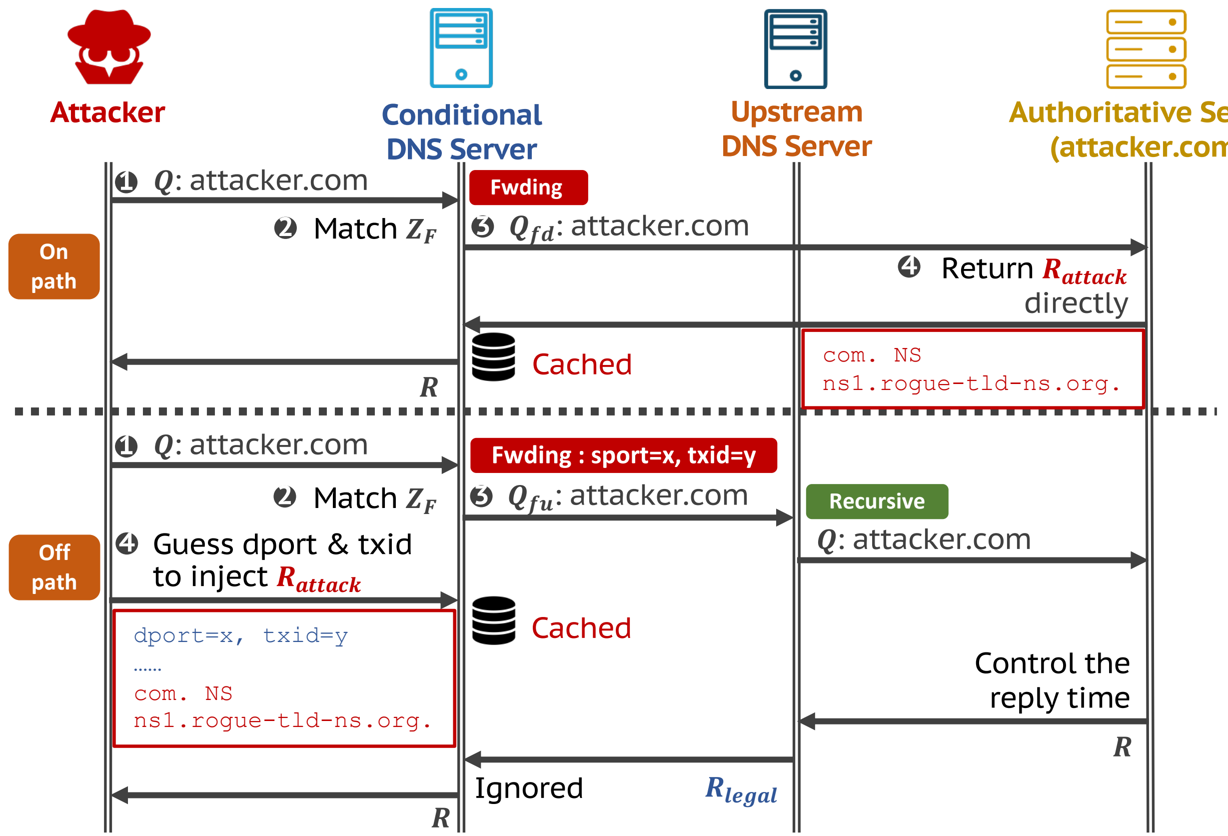
Algorithm 1: DNS resolution process

```
input  : A DNS Request from clients
output : A DNS Reply to clients
main()
    step_0: InitQuery(Q, Request)
    step_1: if SeachCache(Q, Cache) then
        goto final
    step_2: FindServers(Q, TgtSvrs)
    step_3: SendQuery(Q, TgtSvrs)
    step_4: ProcessResponse(Q, R)
        if ServerIsError(Q, R) then
            goto step 3
        if not MatchQuery(Q, R) then
            goto final
        SanitizeRecords(Q, R)
        if IsReferral(Q, R) then
            if not IsFwding() then
                UpdateQuery(Q)
                goto step 2
        if IsCNAME(Q, R) then
            UpdateQuery(Q)
            goto step 1
        CacheRecords(R, Cache)
    final: ConstructReply(Reply)
    return Reply
InitQuery(Q, Request)
    initialize Q.name, Q.type, Q.zone
    if IsFwding() then
        ModifyFwdQuery(Q)
SanitizeRecords(Q, R)
    for RR ∈ R do
        if OutofBailiwick(RR) then
            remove RR from R
UpdateQuery(Q, R)
    update Q.name, Q.type, Q.zone
```

Attack Steps of MaginotDNS

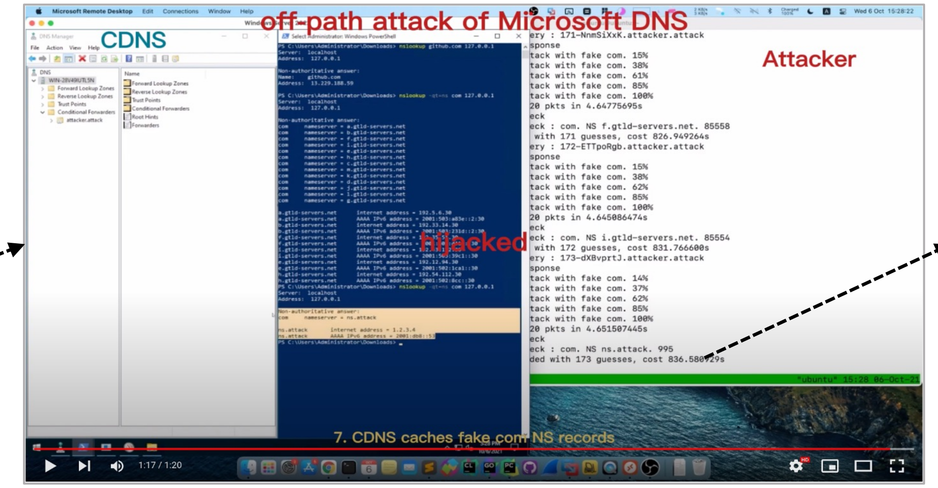
- **On-path Attack**
  - ❑ Returning fake responses directly
  - ❑ **BIND**, **MS DNS**, **Knot**, and **Technitium**
- **Off-path Attack**
  - ❑ Guessing src port & TXID with birthday attack
  - ❑ **Microsoft**: our found **new port vulnerability**
  - ❑ **BIND9**: extending the SADDNS attack

All future queries will be hacked.



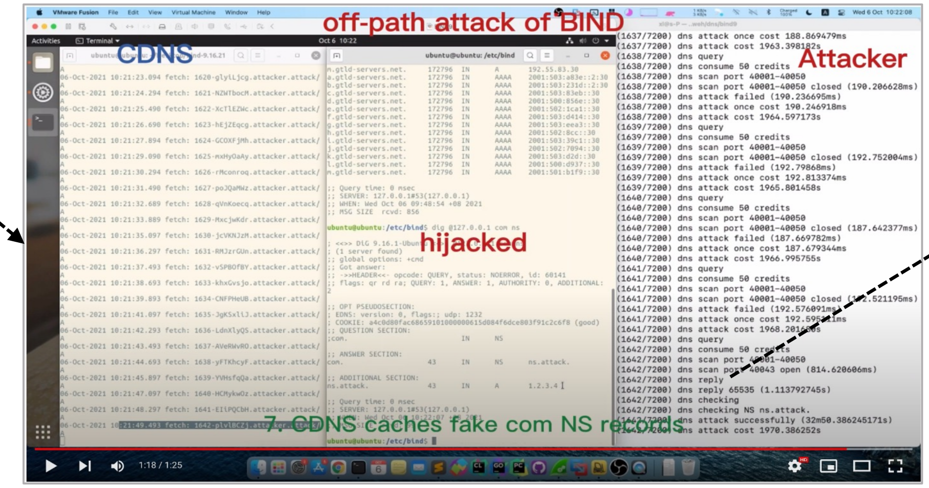
MaginotDNS Attack Demos

- **On-path Attack**
  - The result is determinative
- **Off-path Attack**
  - Microsoft: **avg. 802s**
  - BIND9: **avg. 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
```

Log of Attacking Microsoft




```
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)
```

Log of Attacking BIND9

Watch videos here.

Vulnerable CDNS Population

xmap Public

XMap is a fast network scanner designed for performing Internet-wide IPv6 & IPv4 network research scanning.

C ☆ 260 43

➢ Measurement with XMap

- ❑ We collected **1.2M resolvers**
- ❑ Removing not-applicable ones, such as violating NR or multiple caches
- ❑ Applying our **new method** to identify **154,955 CDNSes**
- ❑ Using **software fingerprints** to locate **54,949 vulnerable CDNSes**
  - Resolvers with DNSSEC or 0x20 are filtered out

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%

Discussion & Mitigation

- **Vulnerability Disclosure**
  - Confirmed and fixed by **all affected software**: BIND9, Knot, Microsoft, & Technitium
  - **4 CVE-ids** published & **Bounty** awarded by Microsoft
- **Root Cause**
  - Poor forwarding bailiwick checking implementation
    - `Qry.zone` is set to root → all records is **in-bailiwick** (root's subdomains)
- **Mitigation Solution**
  - `Qry.zone` should be set to the forwarded domain in Z_F
  - Then only records under forwarded domain are acceptable
  - Have been adopted by affected software

Conclusion

- **New Threat Model**
  - A new resolver role: CDNS
- **New Attack Surface, Vulnerabilities, & Attacks**
  - Mixed roles and shared cache
  - Inconsistency of DNS implementation
  - Old DNS mechanism
  - New Vulnerabilities & Attacks
- **New Methodology & Results**
  - CDNS identifying method
  - Numbers of vulnerable CDNSes

Wrap-up

Thanks for listening!
Any questions?

Xiang Li, Tsinghua University

x-l19@mails.tsinghua.edu.cn

Paper

Tool