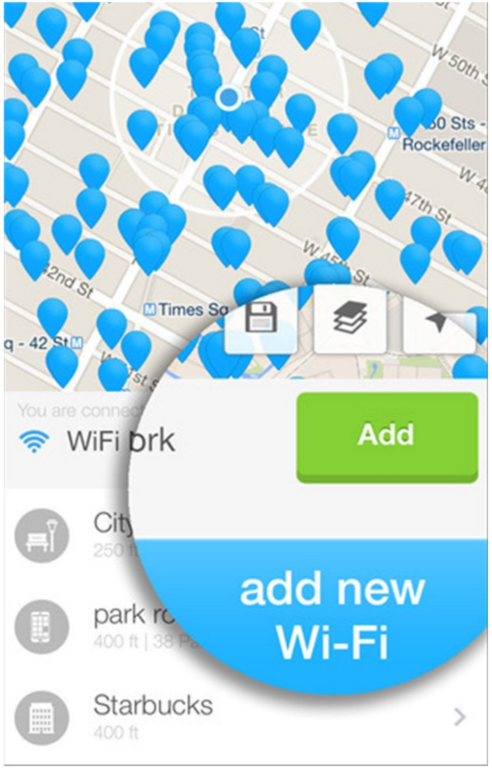
# Poison Over Troubled Forwarders:
# A Cache Poisoning Attack Targeting DNS Forwarding Devices

Xiaofeng Zheng, Chaoyi Lu, Jian Peng, Qiushi Yang, Dongjie Zhou, Baojun Liu, Keyu Man, Shuang Hao, Haixin Duan and Zhiyun Qian

Presenter: **Xiang Li** Tsinghua University

# Internet Access & Common Devices

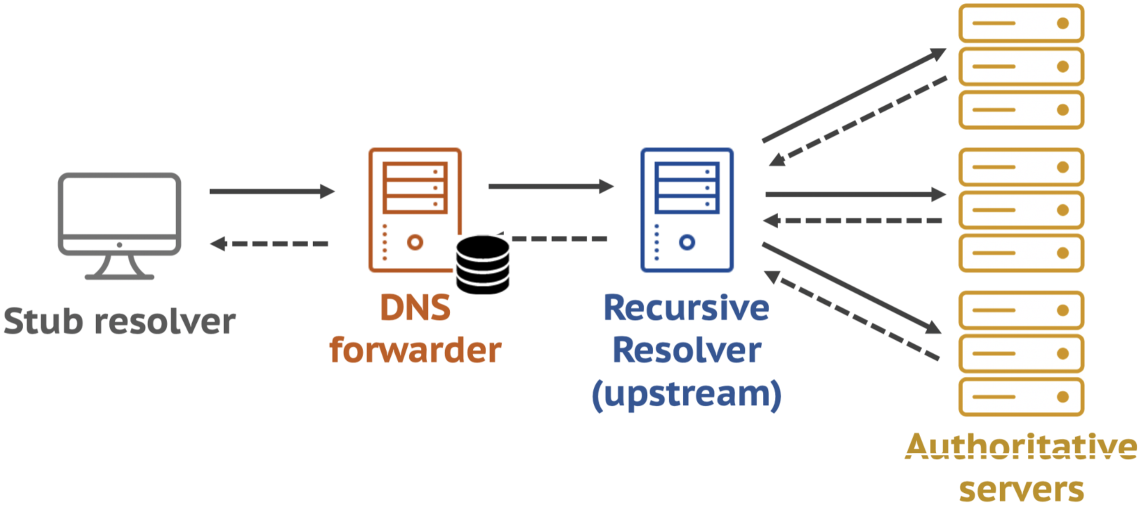


# How does DNS work on these routers and WI-FI networks?

They serve as DNS forwarders

# DNS Forwarder

- Devices standing in between stub and recursive resolvers
  - E.g., home routers, open Wi-Fi networks
  - **Gateways** of access control
  - **Load balancers** for upstream servers

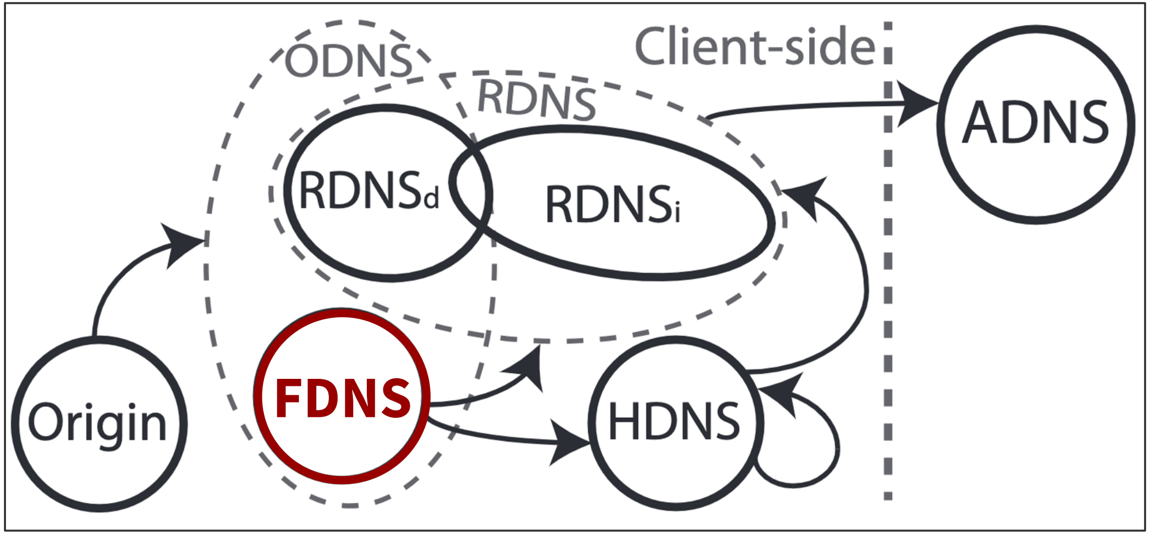


# DNS Forwarder: Prevalent Devices

- Prevalent devices
  - IMC ’14
    - **32M**, 95% are forwarders
  - IMC ’15
    - **17.8M**, 76.4% are residential devices
  - **Enabled by various software and routers**
    - BIND, Unbound, Knot Resolver, and PowerDNS
    - TP-Link, D-Link, and Linksys

# DNS Forwarder: Prevalent Devices

- Prevalent devices




  - **Part of the complex DNS infrastructure**

Kyle Schomp, Tom Callahan, Michael Rabinovich, Mark Allman. On measuring the client-side DNS infrastructure. IMC ‘13

# DNS Forwarder: Security

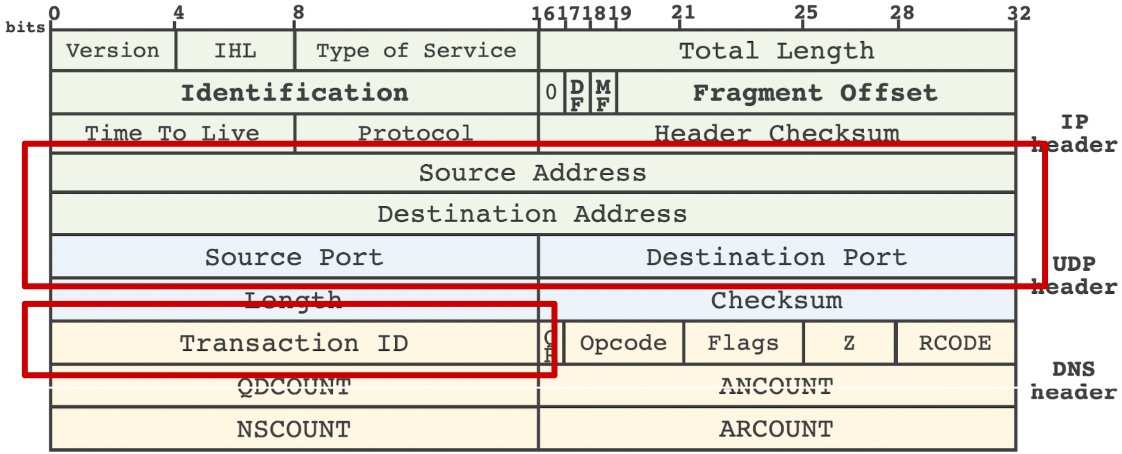
- Security status
  - Forwarder vs Recursive resolver
    - bailiwick check, DNSSEC validation
  - **Relies on the integrity of upstream resolvers**
  - Do not check too much by itself
  - E.g., fail to check the src port and TXID (PAM '14)
    - simple **cache poisoning attacks**
    - DoS attacks

# DNS Cache Poisoning Attacks

One of the most influential attacks targeting DNS resolvers

# DNS Cache Poisoning Attacks

- Forging a valid DNS response
  - Matching the DNS query's metadata
    - Address, Port, DNS transaction ID (TXID), Query name
  - Type 1: **Forging Attacks**
  - Type 2: **Defragmentation Attacks**



# DNS Cache Poisoning Attacks: Type 1

- Type 1: Forging Attacks
  - **Guessing the metadata**, e.g., TXID, src port
    - e.g., the BIND Birthday Attack, **the Kaminsky Attack**
    - others, e.g.,
    - attack with NAT, DNS proxy attack, sock overloading
  - **Mitigation**
    - **randomize**, randomize, randomize (RFC 5452)
    - src port, TXID, qname

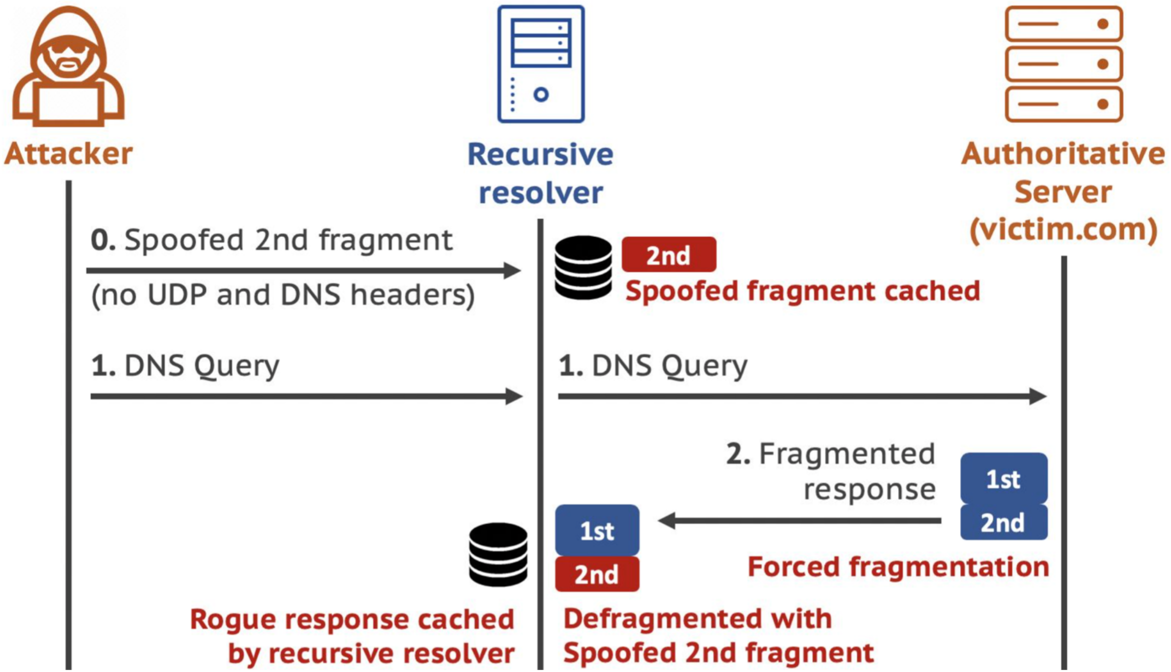
# Do randomization defenses end forging attacks?

Yes or No? Proud or Upset.

E.g., **SAD DNS Attack** with side-channels

# DNS Cache Poisoning Attacks: Type 2

- Type 2: Defragmentation Attacks
  - **Circumventing the metadata**, e.g., TXID, src port



# DNS Cache Poisoning Attacks: Type 2

- Type 2: Defragmentation Attacks
  - **Forcing a fragmentation**
  - Lower the MTU → difficult now
    - **0.7%** Alexa Top 100k domains is willing to reduce the MTU to < 528 bytes
    - **0.3%** of 2M open resolvers can reduce the MTU to < 512 bytes
  - Use the DNSSEC records → cannot target arbitrary domains
    - Non-validating recursive resolvers
    - DNSSEC deployment is still low
    - The attack **only works for DNSSEC-signed domains**

# Our New Defragmentation Attack

Targeting DNS forwarders

Motivation

Threat Model

Attack Workflow

Experiment

Discussion

# Threat Model: Overview

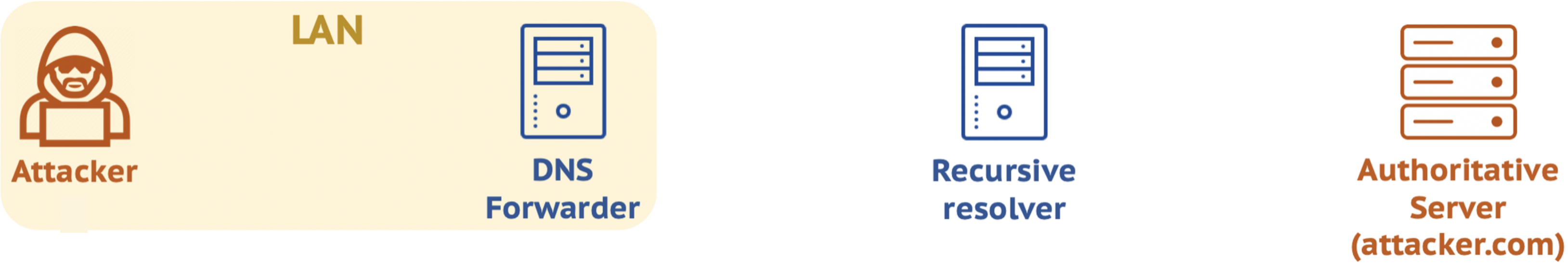
- Defragmentation attacks targeting DNS forwarders
    - **Reliably** force DNS response fragmentation
    - Target **arbitrary victim domain names**

# Threat Model: Overview

- Defragmentation attacks targeting DNS forwarders
  - **Reliably** force DNS response fragmentation
  - Target **arbitrary victim domain names**

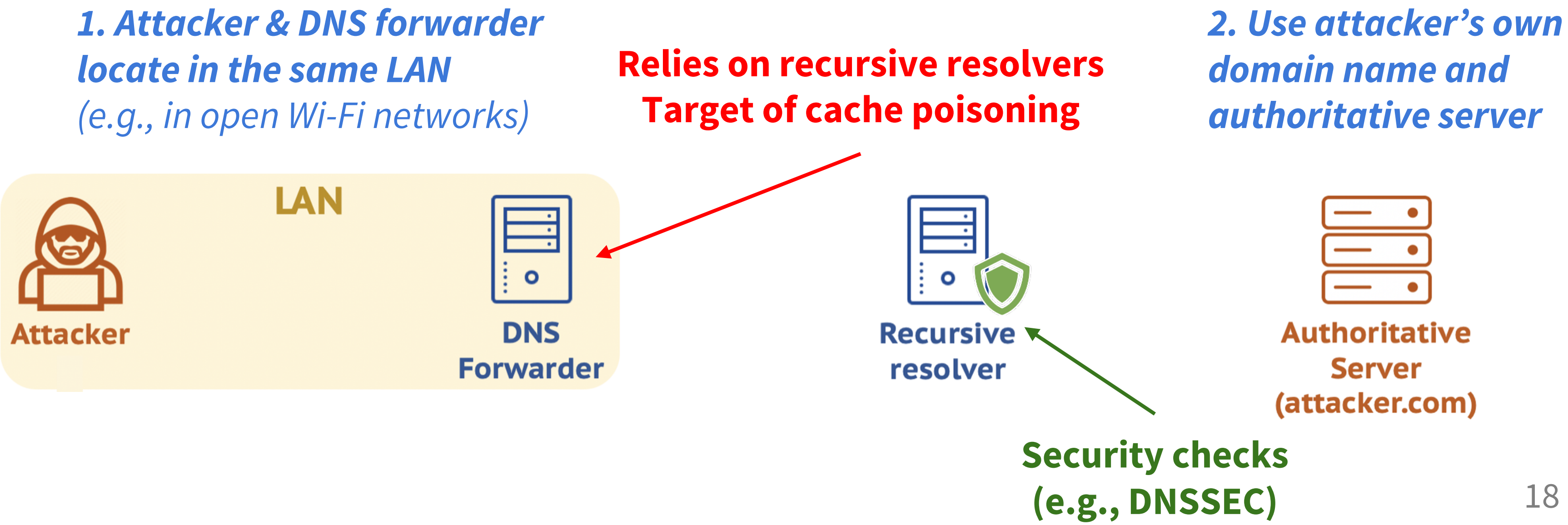
***1. Attacker & DNS forwarder locate in the same LAN***
*(e.g., in open Wi-Fi networks)*

***2. Use attacker's own domain name and authoritative server***



# Threat Model: Insight on Forwarder Roles

- Defragmentation attacks targeting DNS forwarders
  - **Reliably** force DNS response fragmentation
  - Target **arbitrary victim domain names**



Motivation

Threat Model

Attack Workflow

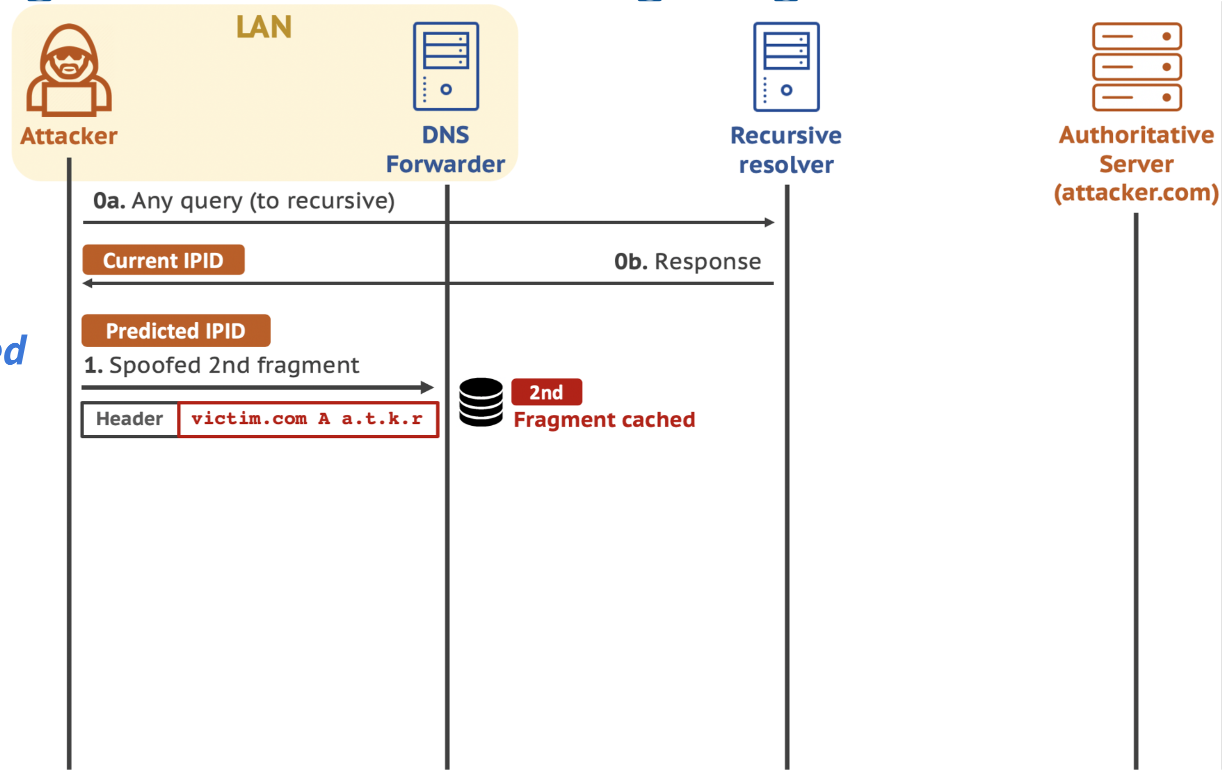
Experiment

Discussion

# Flow of Defragmentation Attack: Step 0&1

- Defragmentation attacks targeting DNS forwarders

*1. Craft spoofed 2nd fragment*

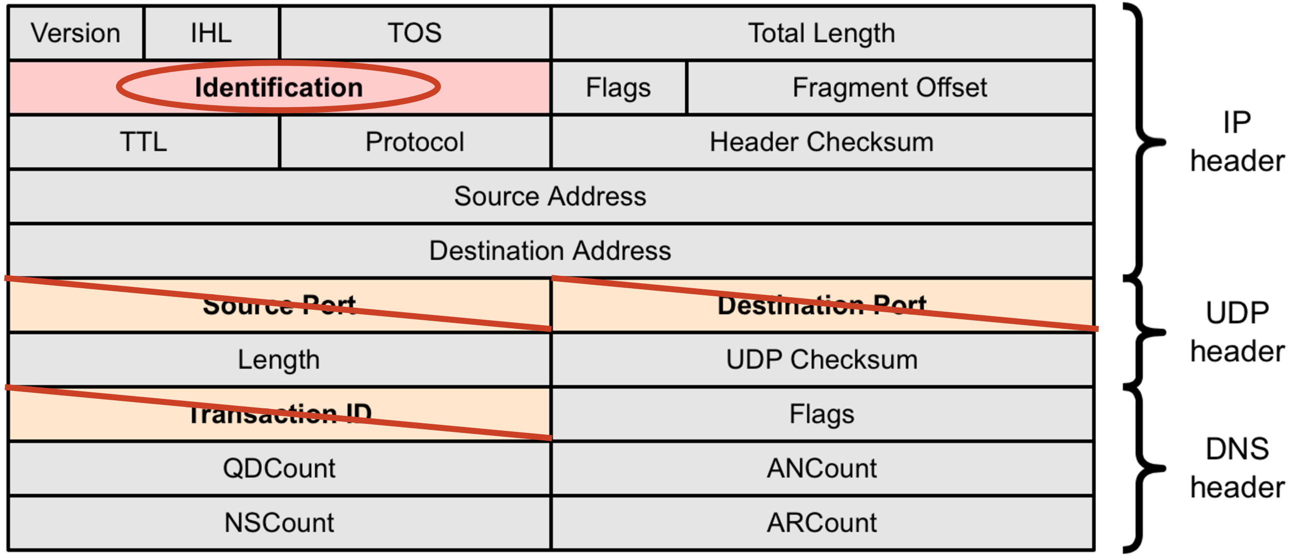


# Crafting Spoofed 2nd Fragment

Challenge: guessing the IPID

# Crafting Spoofed 2nd Fragment

- No UDP and DNS headers in the 2nd fragment
- IPID Prediction is needed
  - The IPIDs of the 2nd and 1st fragment should agree



# Crafting Spoofed 2nd Fragment

- IPID assignment algorithms
  - Global IPID Counter
  - Random IPID Counter
  - **Hash-based IPID Counter**
    - key: <src IP, dst IP>
    - increased number: [1, the number of system ticks]
- Predicting the hash-based IPID
  - same “NAT-ed” public src address
  - send the 2nd fragment quick

# Crafting Spoofed 2nd Fragment

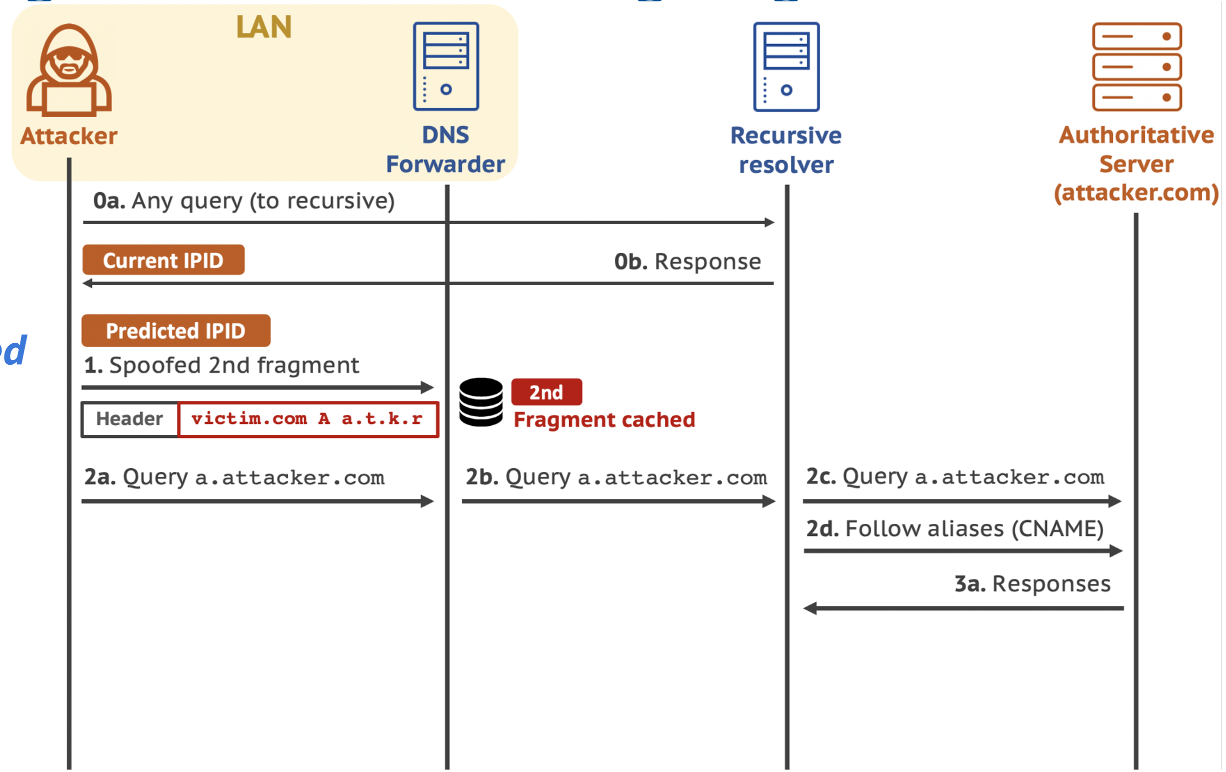
- Predictable IPID measurement results
  - Incremental IPID counter
    - **Open DNS resolvers**: 4.2M
  - Hashed-based IPID counter
    - **OS**: Windows 10 (version 1909), ubuntu (5.3.0.29-generic)
    - **Public DNS services**:
    - Cloudflare, Quad9, Comodo, OpenDNS, Norton
- Other header fields
  - Fragment offset
  - IP source address
  - UDP checksum

# Flow of Defragmentation Attack: Step 2

- Defragmentation attacks targeting DNS forwarders

*1. Craft spoofed 2nd fragment*

*2. Issue a DNS query*

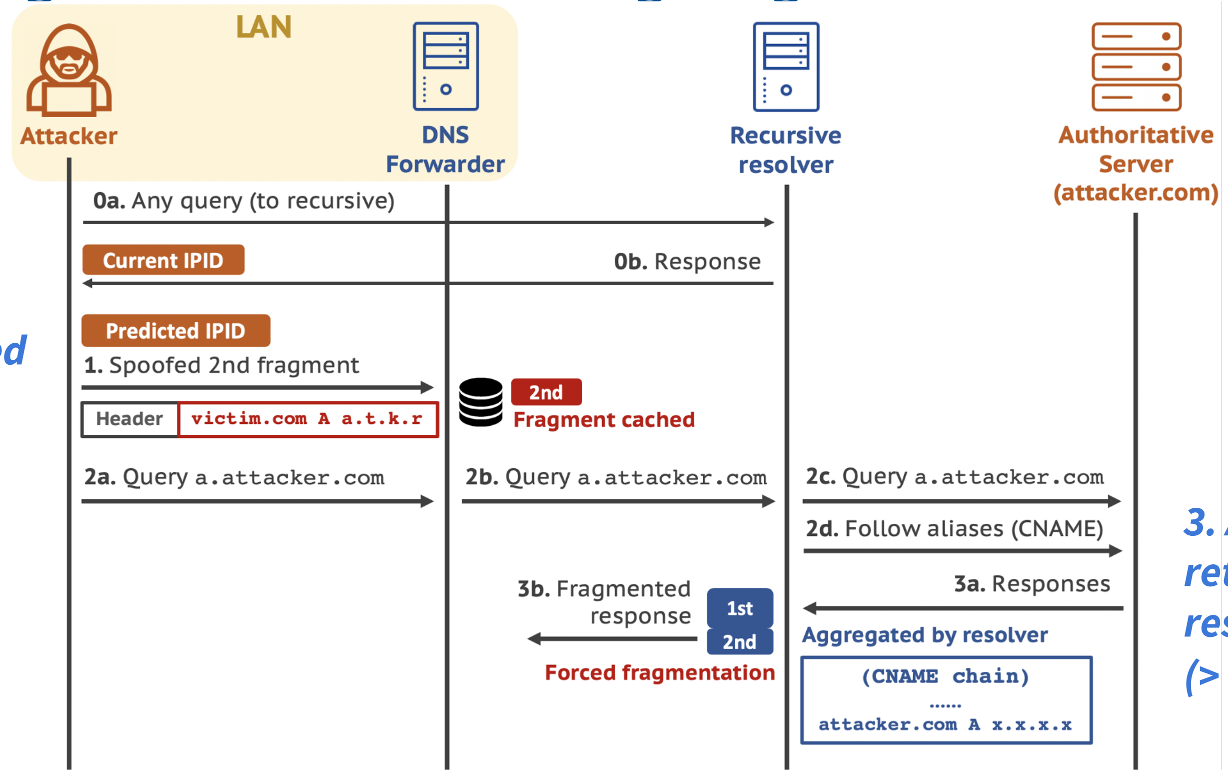


# Flow of Defragmentation Attack: Step 3

- Defragmentation attacks targeting DNS forwarders

*1. Craft spoofed 2nd fragment*

*2. Issue a DNS query*



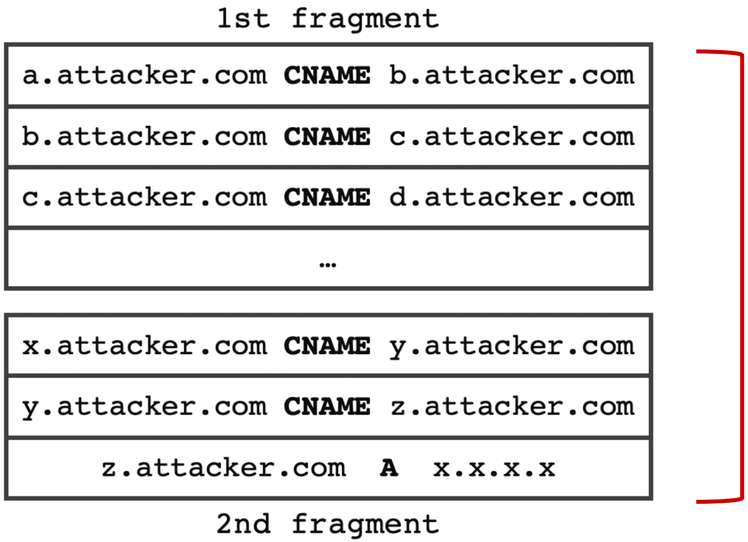

*3. Authoritative returns oversized response (> Ethernet MTU)*

# Forcing a fragmentation of the DNS Response

Via oversized DNS responses

# Attacker’s Oversized DNS Response

- CNAME chain
  - Use dummy **CNAME records** to enlarge attacker’s DNS response

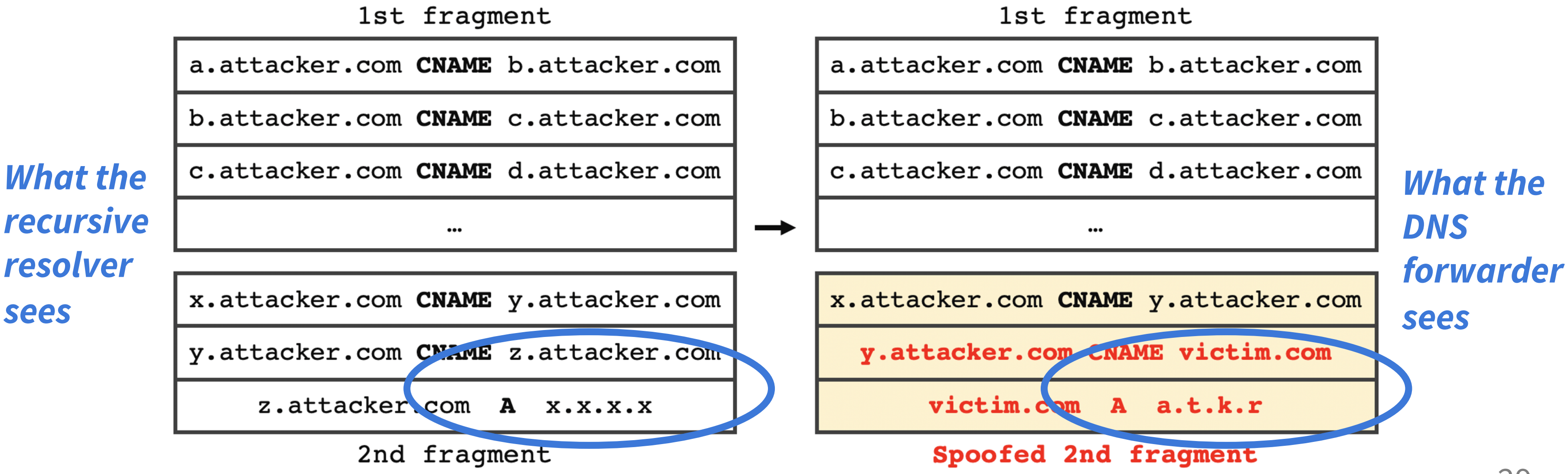



**> 1,500 Bytes (Ethernet MTU)**

**Always produce fragments**

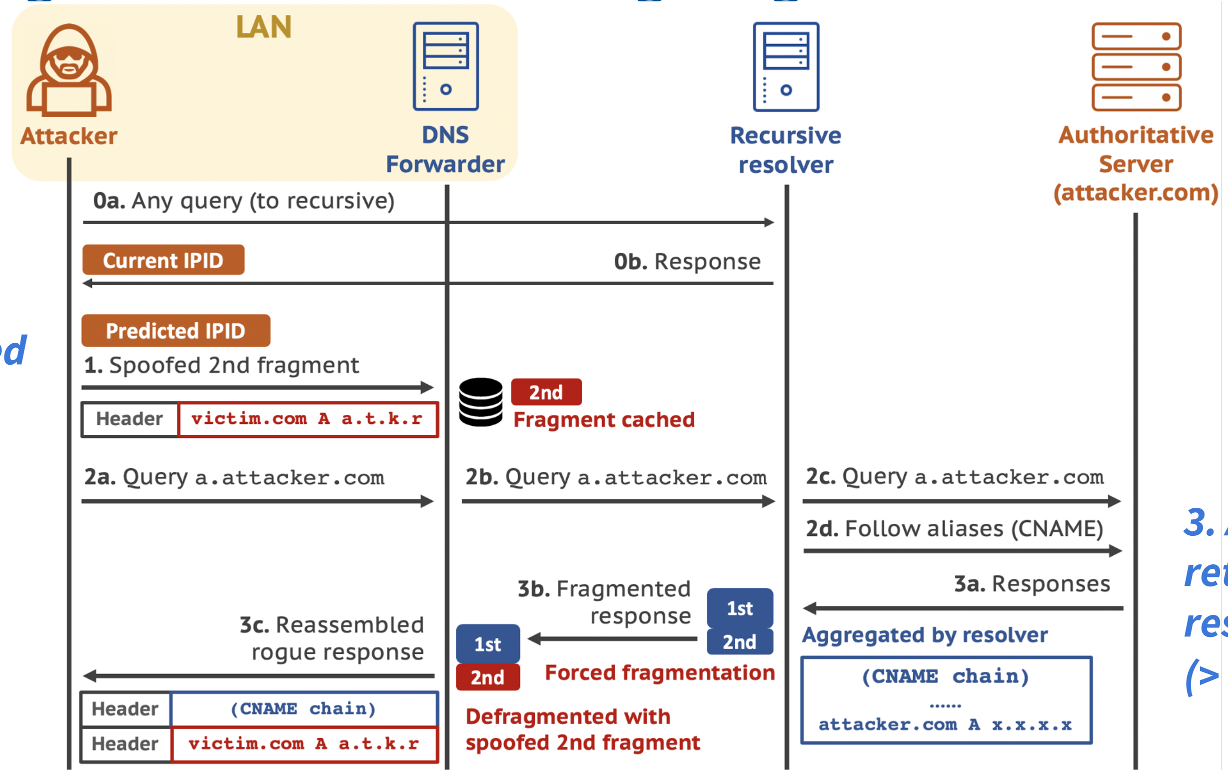
# Attacker's Oversized DNS Response

- CNAME chain
  - Use dummy **CNAME records** to enlarge attacker's DNS response
  - Use CNAME to **point attacker's domain to any victim**



# Flow of Defragmentation Attack: Step 4

- Defragmentation attacks targeting DNS forwarders




*1. Craft spoofed 2nd fragment*

*2. Issue a DNS query*

*3. Authoritative returns oversized response (> Ethernet MTU)*

*4. Defragment by forwarder*

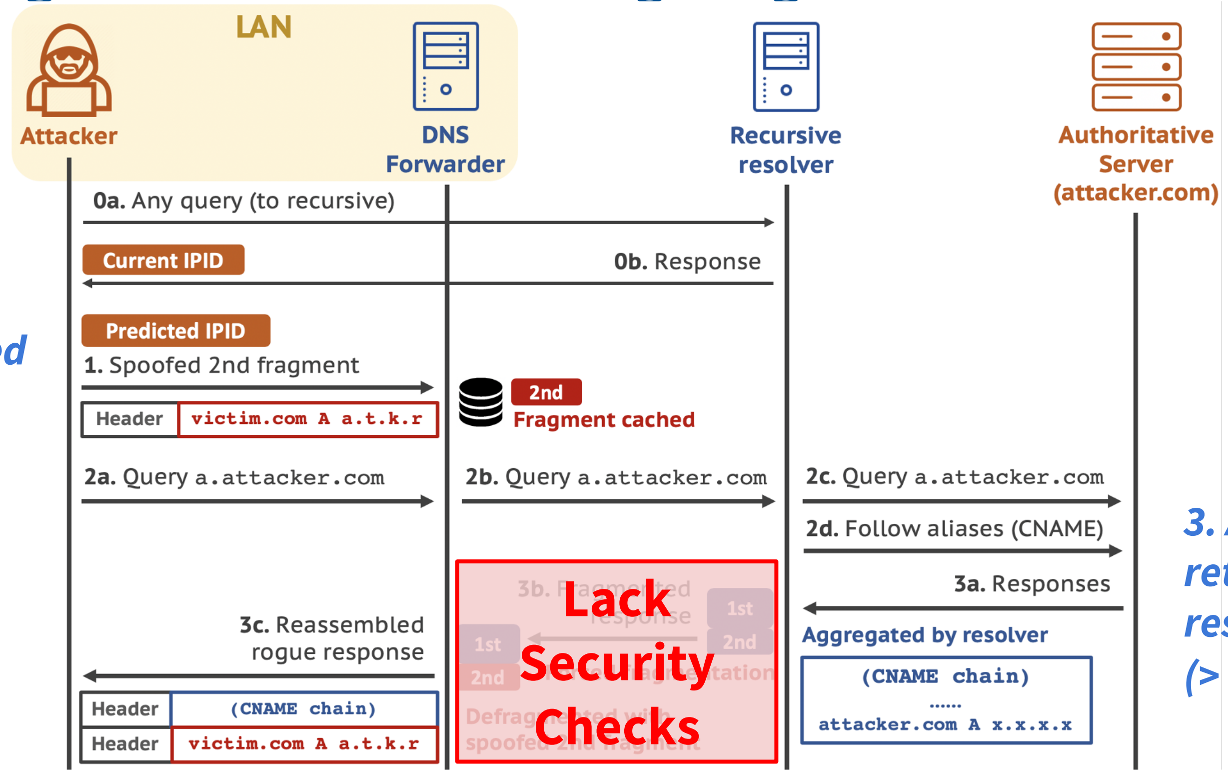
# Flow of Defragmentation Attack: Bingo

- Defragmentation attacks targeting DNS forwarders

*1. Craft spoofed 2nd fragment*

*2. Issue a DNS query*

*4. Defragment by forwarder*




*3. Authoritative returns oversized response (> Ethernet MTU)*

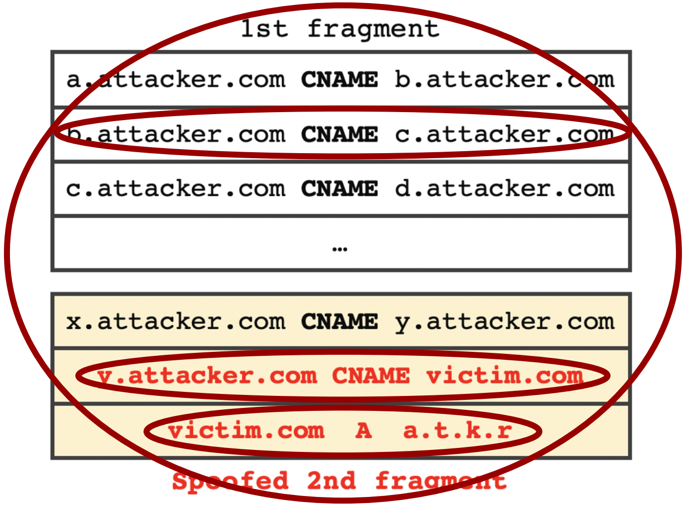
# Conditions of Successful Attacks

# Conditions of Successful Attacks: C1

- EDNS(0) support
  - Allows transfer of DNS messages **> 512 Bytes over UDP**
  - To force a fragmentation
  - **Is being increasingly supported** by DNS software
    - BIND, Knot DNS, Unbound, and PowerDNS
  - Is supported by most recursive resolvers

# Conditions of Successful Attacks: C2

- DNS caching by record
  - ~~Caching the answers as a whole~~
    - a.attacker.com A a.t.k.r
  - **Caching the answers by record**
    - a.attacker.com CNAME b.attacker.com
    - …
    - victim.com A a.t.k.r



# Conditions of Successful Attacks: others

- No active truncation of DNS response
  - Ensures that the entire oversized response is transfered
- No response verification
  - DNS forwarders rely on upstream resolvers
  - No “re-query” for the aliases

Motivation

Threat Model

Attack Workflow

Experiment

Discussion

# Which DNS software is vulnerable?

# Vulnerable DNS Software

- Test results
  - **2 kinds of popular DNS software** are vulnerable
  - **dnsmasq** (used by OpenWRT), **Microsoft DNS**
  - others
    - DNRD caches DNS responses as a whole
    - BIND, Unbound, Knot, and PowerDNS **re-query** the CNAME chain

| Software | Version | EDNS(0) & No truncation | Cache by Record | No Verification | Vulnerable |
|---|---|---|---|---|---|
| dnsmasq | 2.7.9 | ✓ | ✓ | ✓ | ✓ |
| MS DNS | 2019 | ✓ | ✓ | ✓ | ✓ |

# Vulnerable Home Routers

- Test results
  - 16 models are tested (by real attacks in controlled environment)
  - **8 models** are vulnerable
  - others
    - either do not support EDNS(0) or truncate the large response
    - **no one re-queries the aliases**

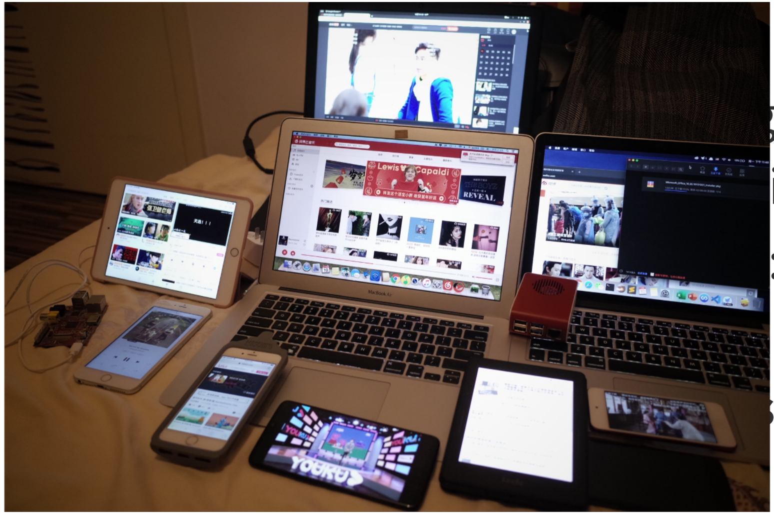
| Brand | Model | EDNS(0) | No Truncation | Cache by Record | Vulnerable |
|---|---|---|---|---|---|
| D-Link | DIR 878 | ✓ | ✓ | ✓ | ✓ |
| ASUS | RT-AC66U B1 | ✓ | ✓ | ✓ | ✓ |
| Linksys | WRT32X | ✓ | ✓ | ✓ | ✓ |
| Motorola | M2 | ✓ | ✓ | ✓ | ✓ |
| Xiaomi | 3G | ✓ | ✓ | ✓ | ✓ |
| GEE | Gee 4 Turbo | ✓ | ✓ | ✓ | ✓ |
| Wavlink | A42 | ✓ | ✓ | ✓ | ✓ |
| Volans | VE984GW+ | ✓ | ✓ | ✓ | ✓ |

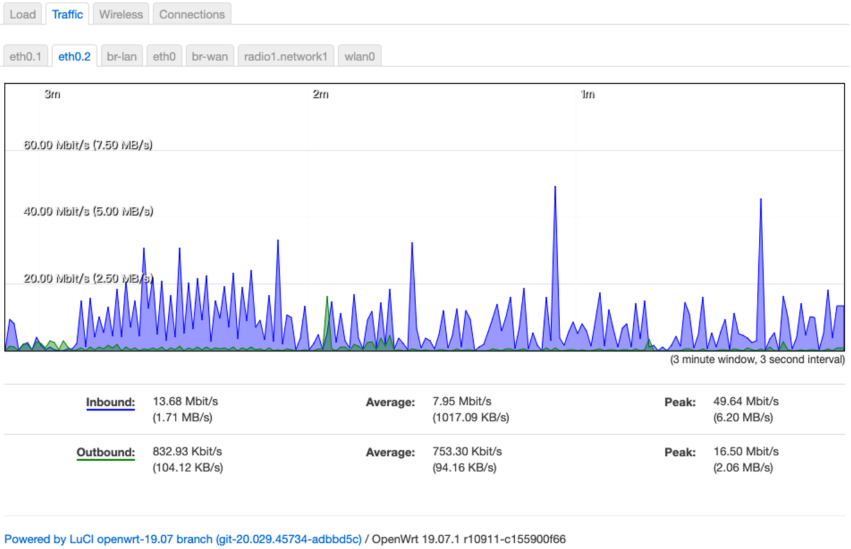
# Real Attacks

- Complex network experiment
  - Home router: OpenWRT with dnsmasq
  - Client and attacker
    - in the same LAN
    - plus 13 other clients, e.g., mobile phones and tablets
    - 7.95Mbps/753.3Kbps of inbound/outbound traffic
  - Upstream recursive resolver: Norton public resolver
  - Authoritative resolver
  - **It takes 58s to complete a successful attack**

# Real Attacks

- Complex network experiment
    - Home router: OpenWRT with dnsmasq
    - Client and attacker

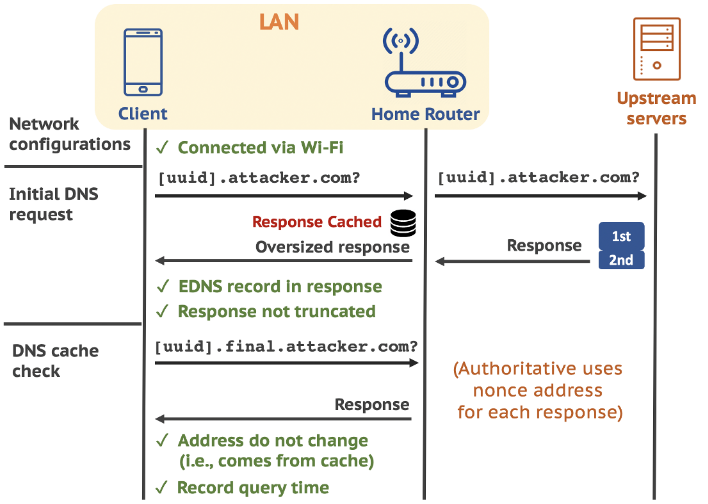


# How many real-world devices are affected potentially?

# Measuring Clients Potentially Under Risk

- Collect vantage points
  - Implement measurement code in a network diagnosis tool
  - **20K clients**, mostly located in China
- Check the forwarder conditions
  - Ethical considerations: no real attack
  - 40% do not support EDNS(0) yet
  - **Estimated vulnerable clients: 6.6%**



# Responsible Disclosure

- Responsible Disclosure
  - Submitting reports and connecting via emails
  - **ASUS** and **D-Link** release firmware patches
    - Caching the responses as a whole
  - **Linksys** accepts the issue via BugCrowd platform
  - **Microsoft** confirms the issue via Microsoft Bounty Program

Motivation

Threat Model

Attack Workflow

Experiment

**Discussion**

# Mitigation

- Mitigation for DNS forwarders
  - **DNS caching by response (short-term solution)**
    - Cache the responses as a whole
  - **0x20 encoding on DNS records**
    - Encode names and aliases in all records
  - **Perform response verification**
    - DNSSEC
    - Re-query all names and aliases
      - Should the forwarder do verification?
      - **Lack clear guidelines of DNS forwarders**

# So, what are DNS forwarders?

What role should they play?

What features should be supported?

# DNS Forwarder Specifications

- RFC 1034
  - No discussion on DNS forwarding
- Now, multiple layers of server
  - stub resolver, **forwarder**, recursive resolver, authoritative resolver
- Different RFCs, different names
  - RFC 2136, 2308, 3597, 5625, 7626, 7871, 8499
- Two definitions of “forwarder”
  - **D1:** Serve as upstream servers of recursive resolvers
  - **D2:** Stand between stub resolvers and recursive resolvers

# DNS Forwarder Specifications: D1

- Definition 1
  - **Serve as upstream servers of recursive resolvers**
- Uses
  - Be leveraged to access authoritative servers
  - Have better Internet connection or bigger cache ability

| RFC | Title | Description |
|---|---|---|
| **2136** | Dynamic Updates in the Domain Name System (DNS UPDATE) | When a **zone slave** forwards an UPDATE message…, enter the role of “**forwarding server**”. |
| **2308** | Negative Caching of DNS Queries (DNS NCACHE) | … a bigger cache which may be **shared amongst many resolvers**. |
| **7626** | DNS Privacy Considerations | ... these **forwarders** are like resolvers. |

# DNS Forwarder Specifications: D2

- Definition 2
    - **Stand between stub resolvers and recursive resolvers**
- Uses
    - Take queries from clients, pass the requests on to another server

| RFC | Title | Description |
|---|---|---|
| **3597** | Handling of Unknown DNS Resource Record (RR) Types | … **forwarders** used by the client. |
| **5625** | DNS Proxy Implementation Guidelines | (DNS) proxies are usually simple DNS **forwarders** …, **relies on an upstream resolver** ... |
| **7871** | Client Subnet in DNS Queries | **Forwarding Resolvers**, … Recursive Resolver handles the query |
| **8499** | DNS Terminology | stand between stub resolvers and recursive servers. |

# DNS Forwarder Implementations

- Lack clear guidelines of DNS forwarders
  - The term of DNS forwarders is updated by RFC 8499
  - There are no implementation details -> **diverse implementations**
- What should a DNS forwarder do
  - How to **handle** DNS responses
  - Whether should they **cache**
  - Whether should they "**re-query**" some responses
- Only RFC 5625: DNS Proxy
  - DNS proxies should be as transparent as possible
  - Forward DNS packets (up to 4,096 octets)

# Implementation guidelines of the DNS forwarder are needed.

To guarantee better security

- An attack targeting DNS forwarders
- Affects forwarder implementations extensively
- Call for more attention on DNS forwarder security

## Any Questions?

zxf19@mails.tsinghua.edu.cn

# Poison Over Troubled Forwarders:
# A Cache Poisoning Attack Targeting DNS Forwarding Devices

Xiaofeng Zheng, Chaoyi Lu, Jian Peng, Qiushi Yang, Dongjie Zhou, Baojun Liu,
Keyu Man, Shuang Hao, Haixin Duan and Zhiyun Qian
**Xiang Li**