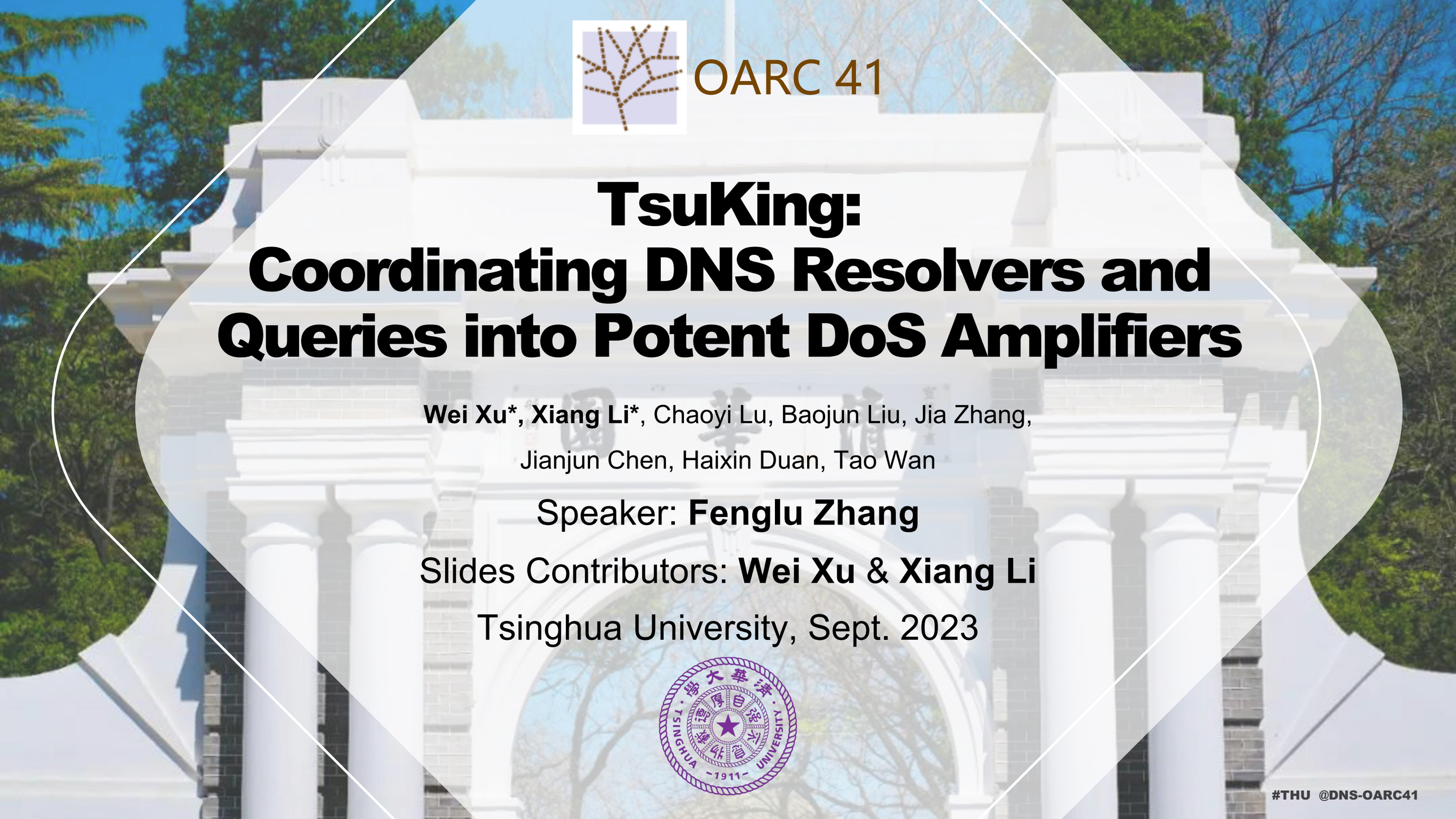


OARC 41

TsuKing: Coordinating DNS Resolvers and Queries into Potent DoS Amplifiers

Wei Xu*, **Xiang Li***, Chaoyi Lu, Baojun Liu, Jia Zhang,

Jianjun Chen, Haixin Duan, Tao Wan

Speaker: **Fenglu Zhang**

Slides Contributors: **Wei Xu** & **Xiang Li**

Tsinghua University, Sept. 2023

Attack Impact

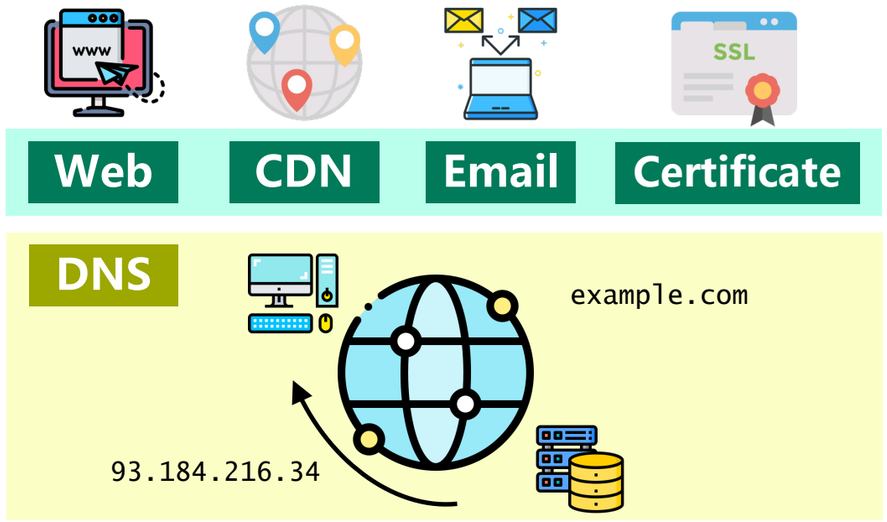
Our TsuKing attack could achieve at least a thousand-fold amplification of DNS packets.

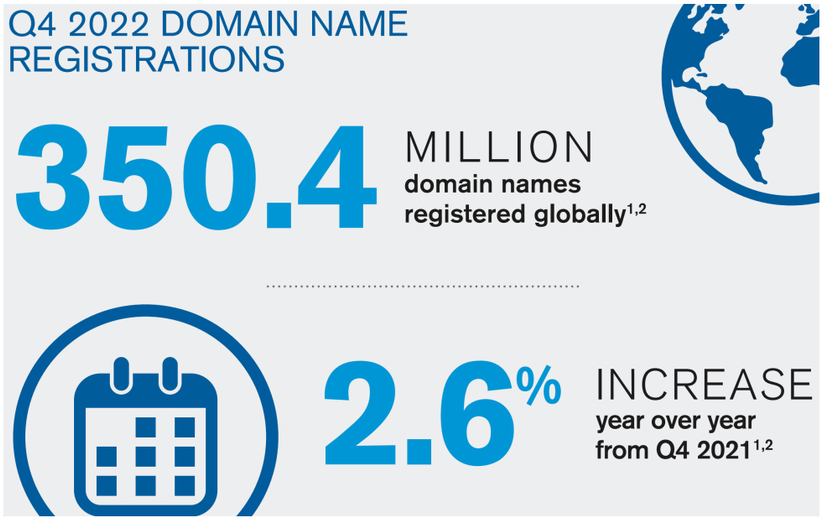
Root cause: DNS protocol non-compliance.

Domain Name System (DNS)

➢ DNS Overview

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



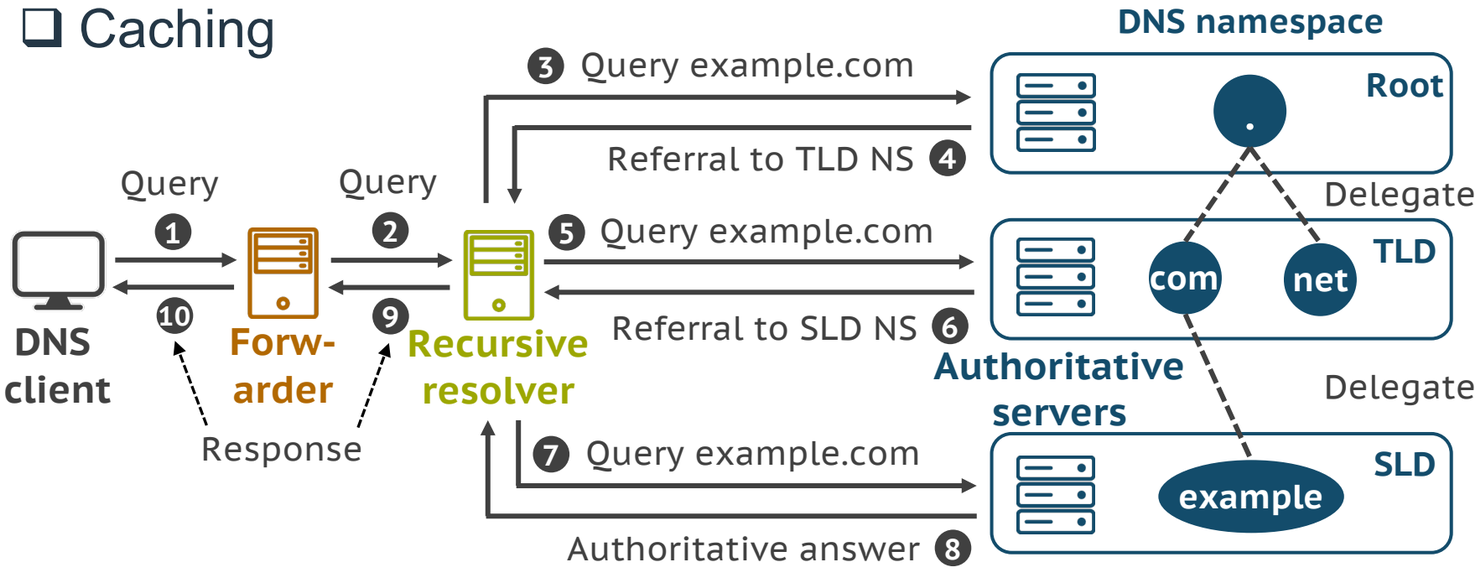


verisign.com/dnib

Domain Name System (DNS)

➢ DNS Resolution Process

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



Query

SP=50000	DP=53	TXID=1001
QD	example.com A?	
AN	(empty)	
AU	(empty)	
AR	(empty)	

Response

SP=53	DP=50000	TXID=1001
QD	example.com A?	
AN	example.com A 1.1.1.1	
AU	(empty)	
AR	(empty)	

Takeaway

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

For a long time, attackers have been attempting to carry out traffic amplification attacks through DNS.

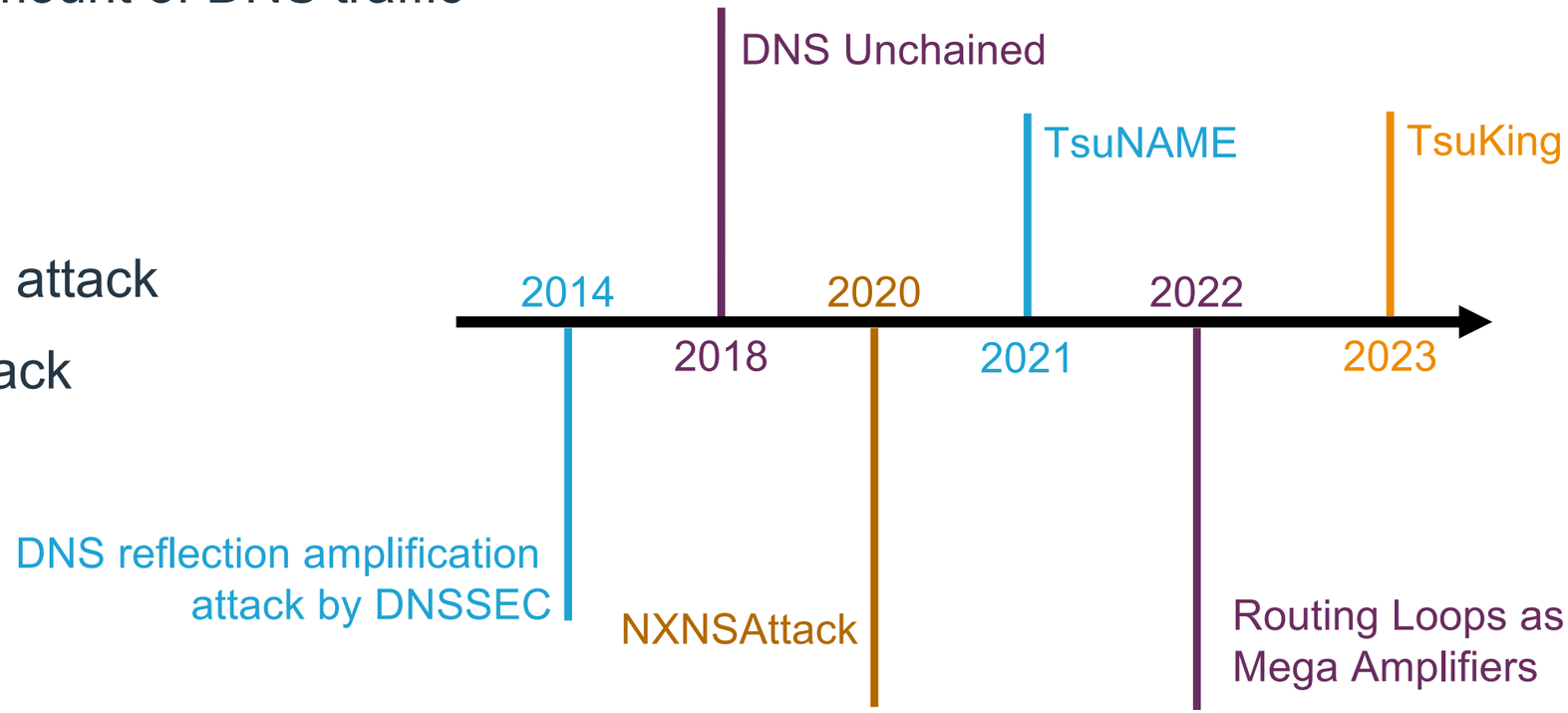
Question

What is the DNS Amplification Attack?

Attackers exploit open DNS resolvers to flood a target with an overwhelming amount of DNS traffic.

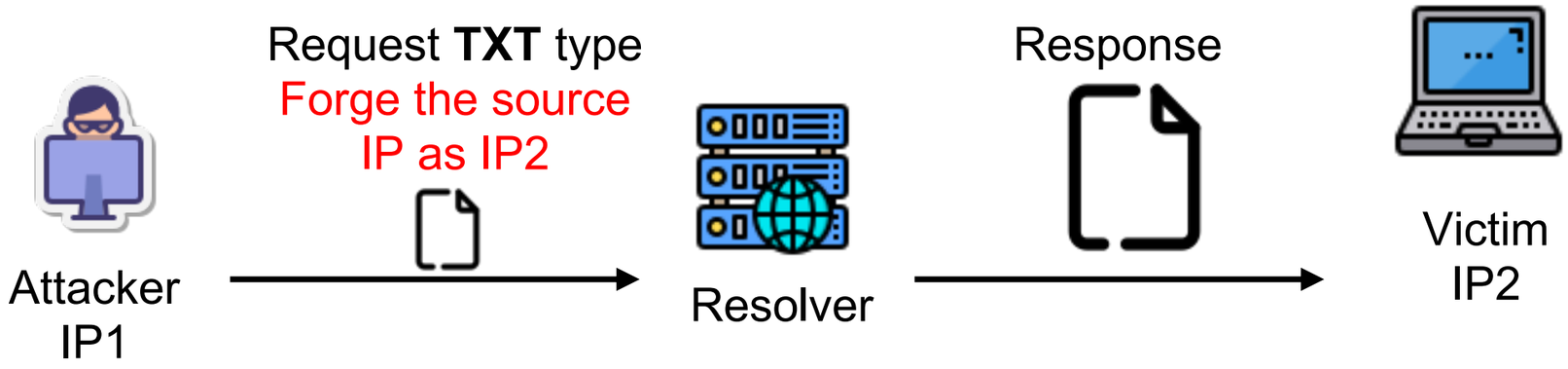
DNS Amplification Attacks

- **Target**
  - To flood a target with amount of DNS traffic
- **Taxonomy**
  - Bandwidth amplification attack
  - Packet amplification attack



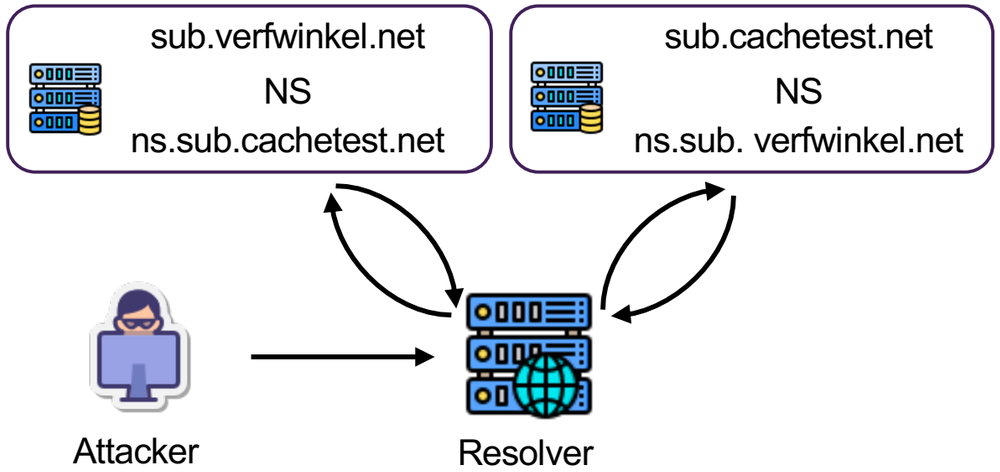
DNS Amplification Attacks

- **Bandwidth Amplification Attack**
  - **DNS reflection amplification attack**
  - Method: forging the source IP address.
  - Result: the victim receives large response packets.



DNS Amplification Attacks

- **Packet Amplification Attack**
  - **TsuNAME attack / NXNSAttack**
  - Method: utilizing NS or CNAME records to initiate multiple requests
  - Result: the victim receives multiple requests



TsuNAME Attack

Takeaway

The essence of a DNS amplification attack is to use small queries to make the victim receive a large amount of traffic.

In the past, the goal was to try to make **a single resolver** send as much traffic as possible.

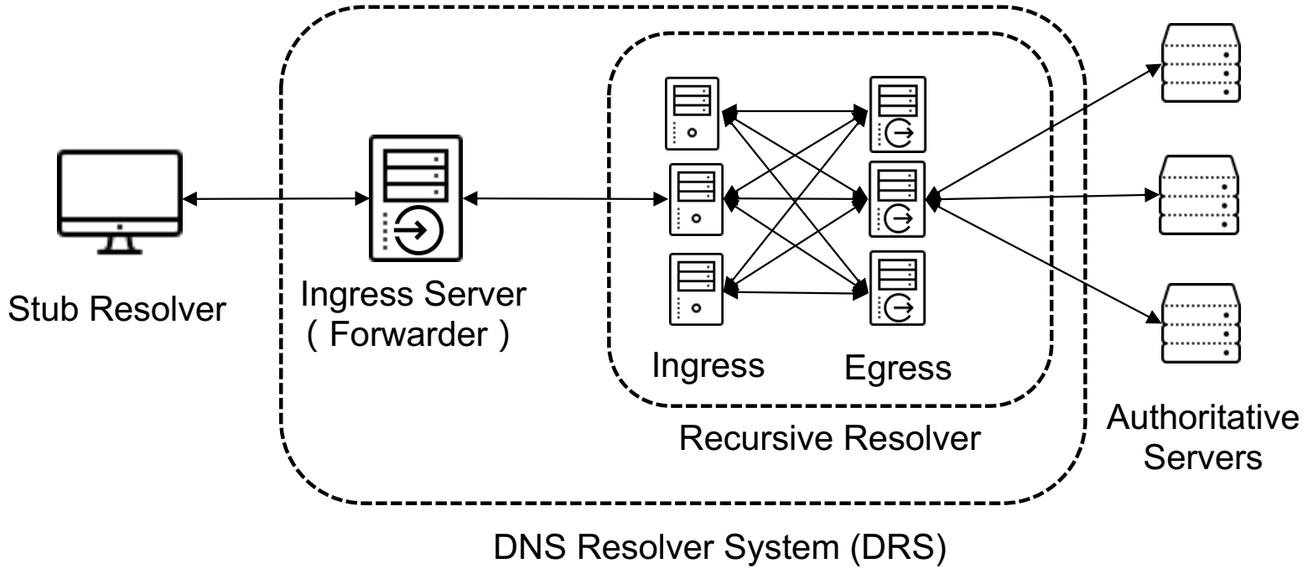
Question

What is the current DNS resolution process?

The emergence of DNS **forwarders** and **load balancers** has introduced more levels into the resolution process.

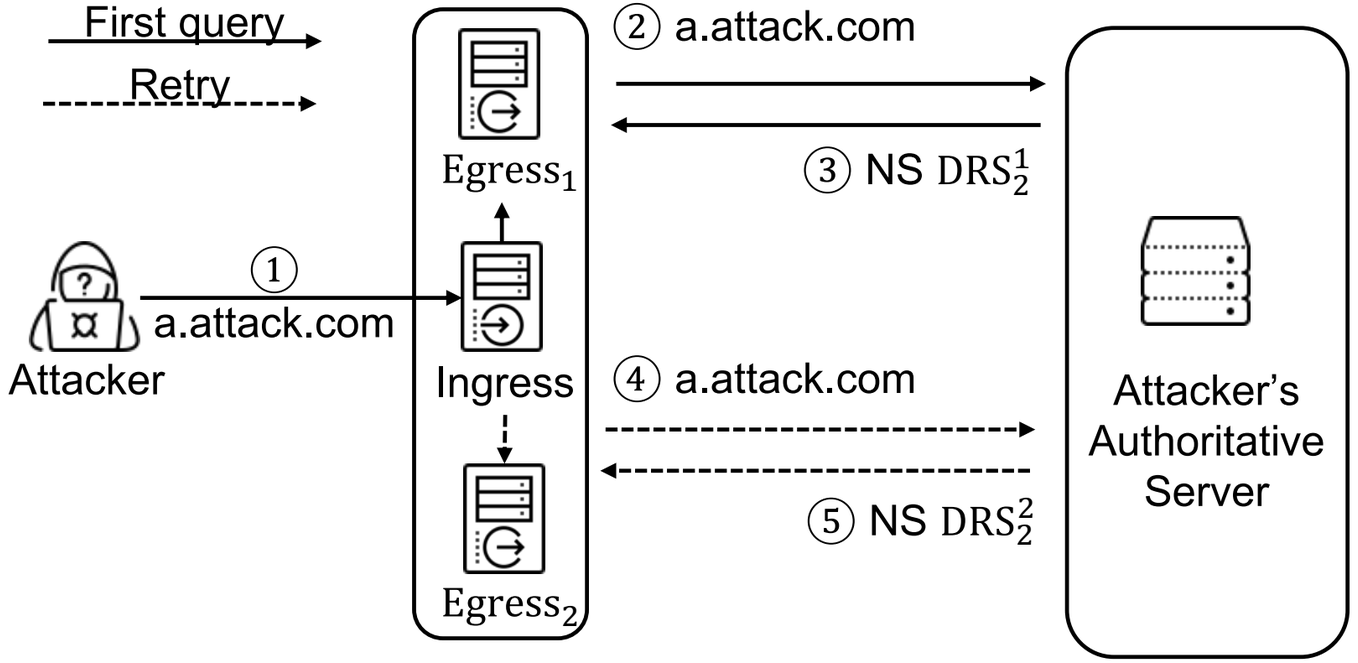
DRS (DNS Resolver System)

- **DNS forwarders** are responsible for forwarding incoming queries to their designated upstream servers.
- **Public recursive resolvers** like Google Public DNS have evolved into complex systems with load balancing, caching clusters, and direct communication with **authoritative servers.**
- We define a **DNS resolver system (DRS)** as an ingress server (such as an open DNS server) and all upstream servers and egress servers in **the resolution paths** until the authoritative.



DRS: Multiple Egresses and Caching

- **Multiple egresses** cache independently with each other
- **Retry** operations will invoke different egresses



Question

How to make multiple DRSes participate together to achieve traffic amplification?

TsuKing: combine multiple DRSes into **a traffic amplifier** and achieve traffic amplification internally.

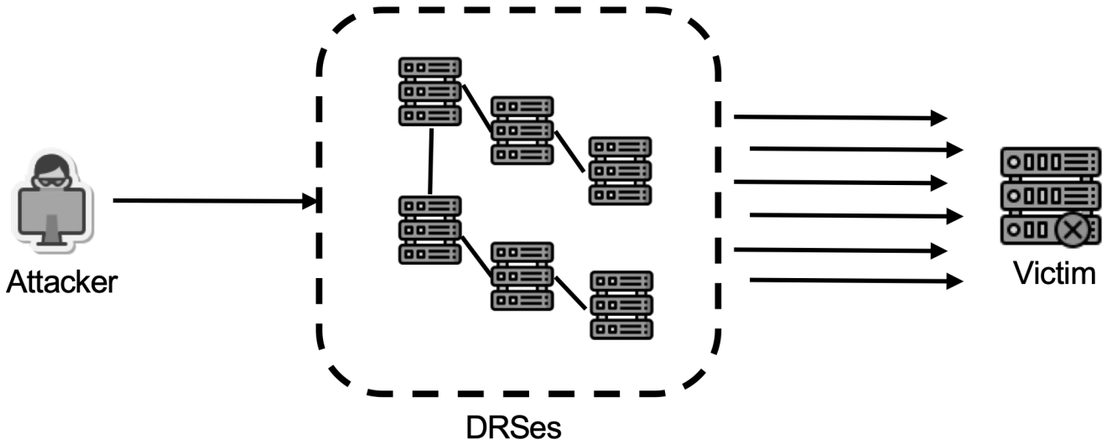
TsuKing Attack

➢ What is the TsuKing Attack

- ❑ Proposed by our **NISL** lab, Accepted by **CCS '23**
- ❑ A new **powerful** type of DNS traffic amplification attack
- ❑ The combination of **multiple DRS features** forms the final vulnerability

➢ Two Critical Steps

- ❑ Coordinating DRSes together
  - ○ Scheduling like the measurement tool **KING**
- ❑ Amplifying DNS packets
  - ○ **Tsunami**-like traffic amplification



[IMC '02] P. Krishna Gummadi et al. King: Estimating Latency between Arbitrary Internet End Hosts.

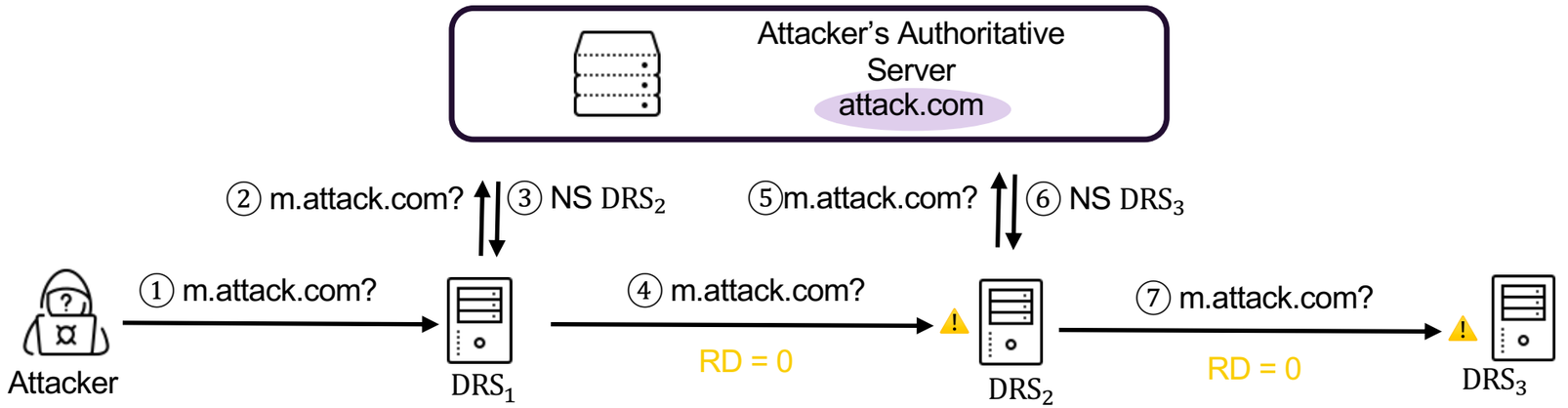
Question

How to combine and coordinate different DRSes?

The utilization of **dynamic NS records** and non-standard handling of **RD flag**.

Coordinating DRS Behavior Using Dynamically Generated NS Records

- **NS records** can control to which target the resolver sends requests.
- Attackers can use **dynamically generated malicious NS records** to continuously **forward requests between different DRS** (with RD handling deficiencies)



No Honor of RD Flag

- The **RD** (Recursion Desired) flag indicates whether clients wish the querying resolver to perform recursive processing.
- In the case of **RD=0**, the resolver should only perform **local resolution** and should not send any further requests externally, such as when requesting authoritative servers.
- However, based on measurement result[1], out of the 1,326,499 open DRSs in the real network, **361,621 (27.26%) do not comply with this specification**.

1: The measurement result is gained on January 2023.

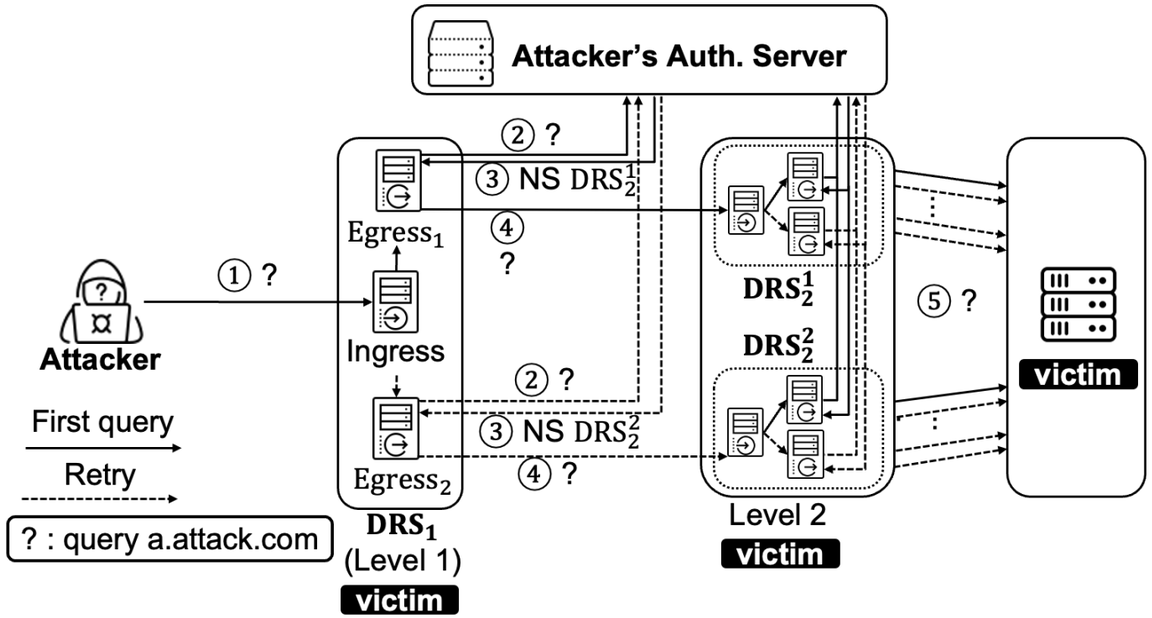
Question

How to amplify traffic across multiple DRSes?

Utilizing DRSes' **multiple egresses and retry** features.

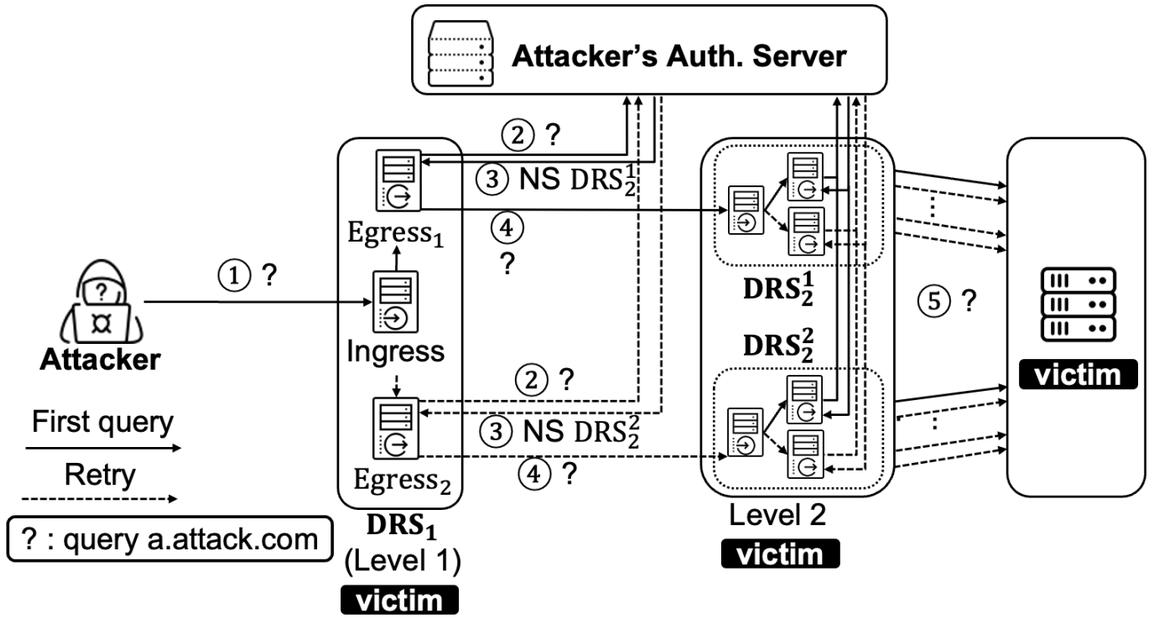
Amplifying Traffic through Multiple Egresses and Retries

- The attacker initiates a query to DRS_1.
- Backend Server $Egress_1$ **of** DRS_1 begins processing the query and receives a malicious NS response from the attacker's authoritative server.
- $Egress_1$ **sends a request to** DRS_2^1.
- Due to DRS_2^1 's non-standard RD handling, it also actively participates in the complete **domain resolution process**.



Amplifying Traffic through Multiple Egresses and Retries

- DRS_1 **fails** to resolve the query.
- $Egress_2$ **starts retrying.** It receives another malicious NS response, causing DRS_2^2 **to also participate** in the domain resolution process.
- DRS_2^1 **and** DRS_2^2 also initiate retries, resulting in the attacker's single query, forwarded by DRS_1, becoming four or more requests across two DRSes.



TsuKing

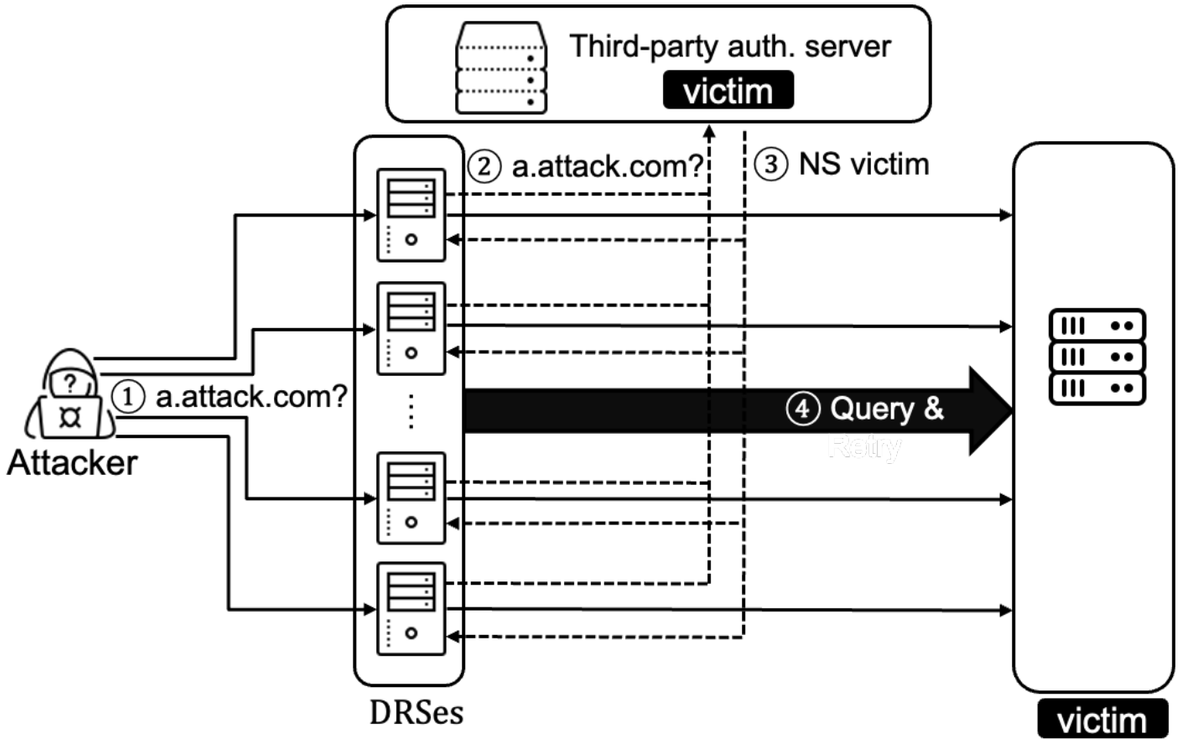
TsuKing Attacks

TsuKing has three attack variants.

TsuKing Attack (1/3): DNSRetry

➢ Exploiting Aggressive Retries to Amplify DNS Traffic

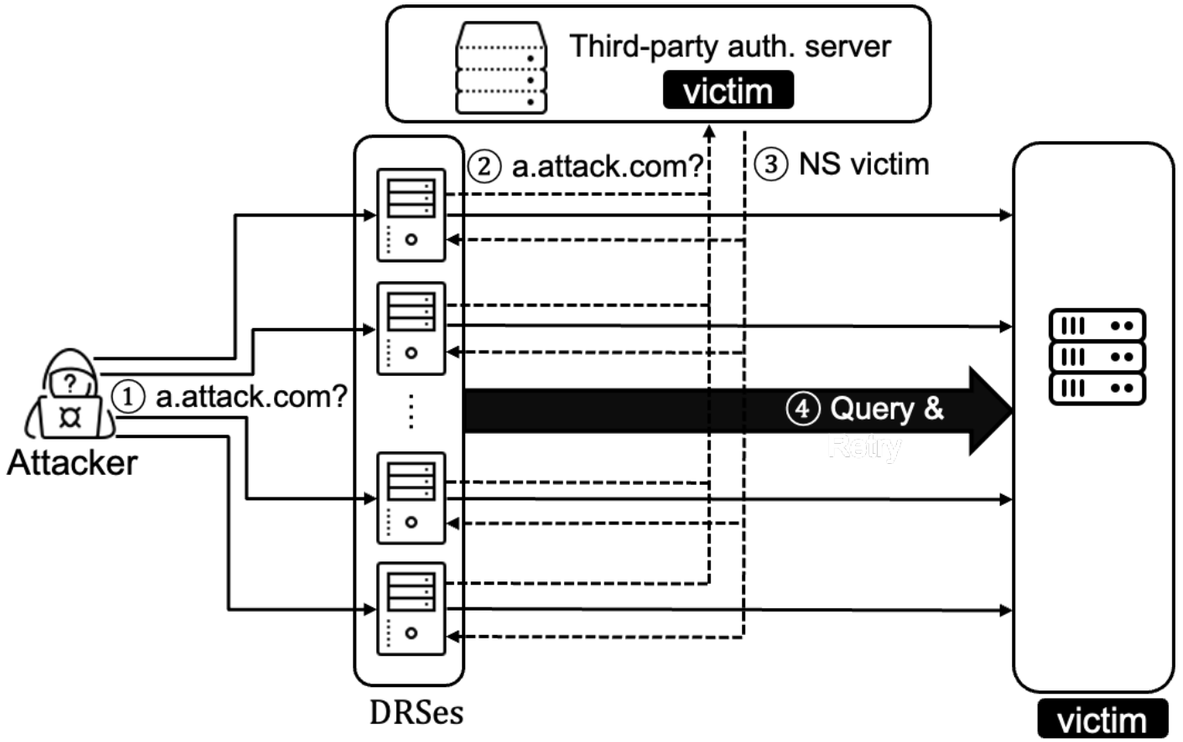
- ❑ Some DRSes exhibit extremely **aggressive retry behavior**, with the highest recorded retry count reaching **117,541 times**, according to measurement results
- ❑ By leveraging malicious NS records, attackers can cause these types of DRSes to **initiate many queries towards the victim**, resulting in traffic amplification.



TsuKing Attack (1/3): DNSRetry

➢ Exploiting Aggressive Retries to Amplify DNS Traffic

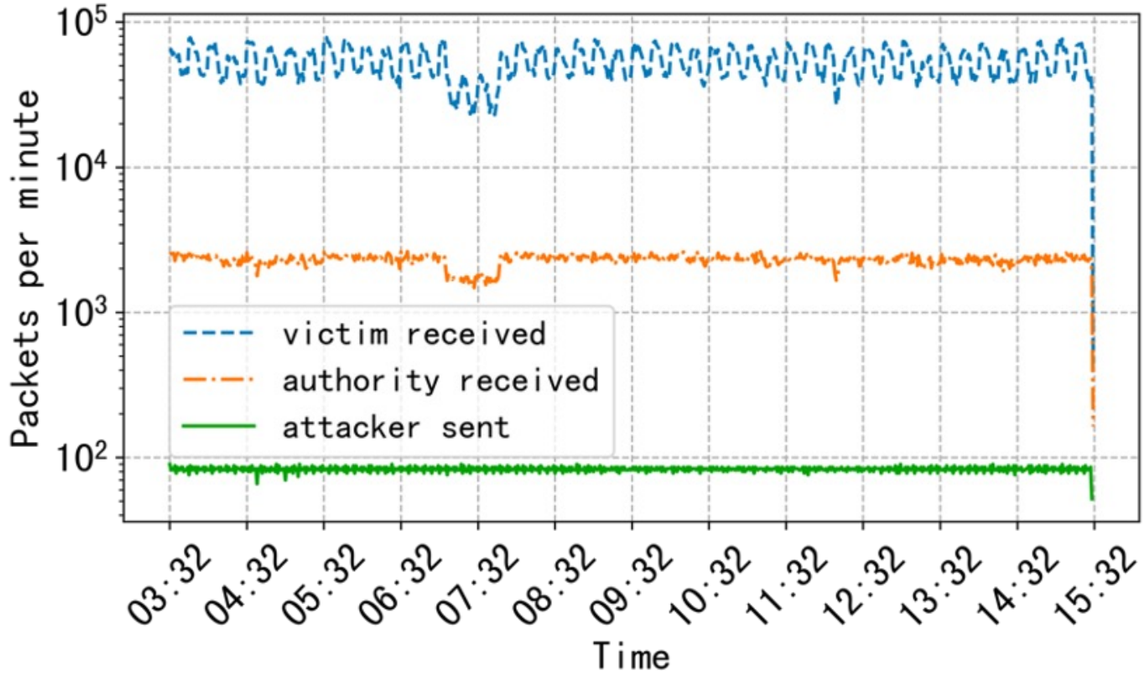
- ❑ The attacker deploys malicious NS records pointing to the victim on a **third-party authoritative server**.
- ❑ The attacker **periodically** sends query requests to the DRS exhibiting aggressive retry behavior.
- ❑ As a result, multiple DRSes will generate **a significant volume of requests towards the third-party authoritative server and the victim**, causing traffic amplification attacks on both entities.



TsuKing Attack (1/3): DNSRetry

➢ Experiment Results

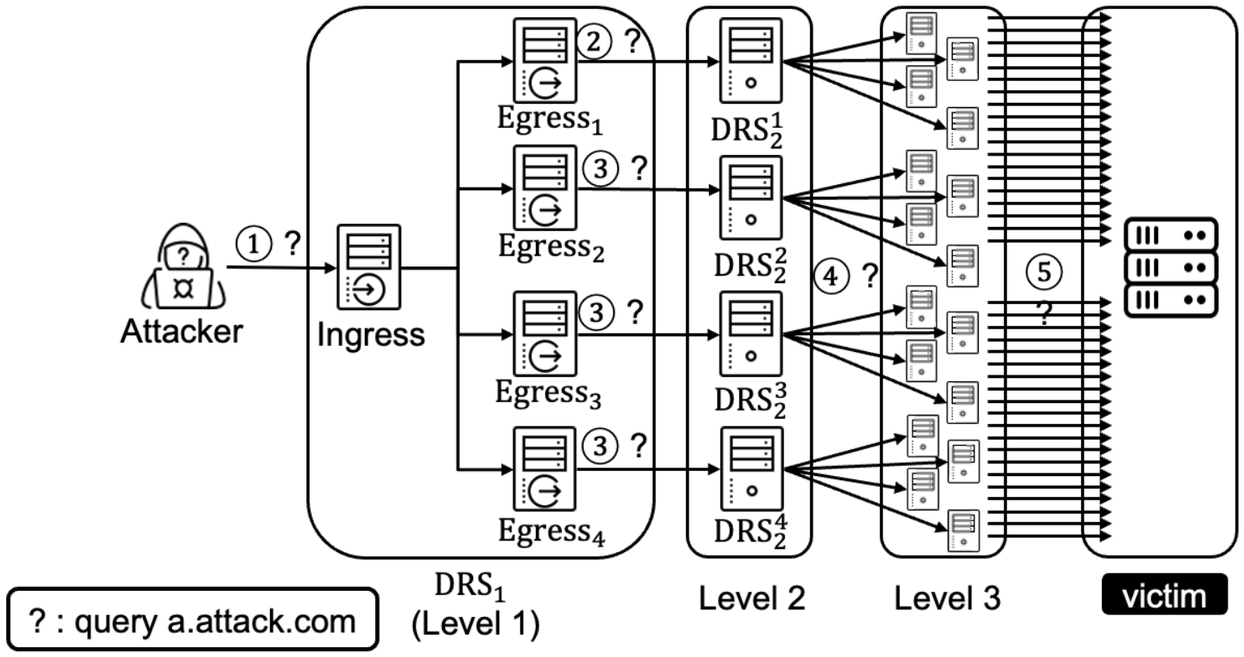
- ❑ In a real-world network experiment, 10 vulnerable DRS servers, each with retries exceeding **1,000** attempts, were organized to launch a 12-hour attack.
- ❑ The attacker's sending rate was **1.38** packets per second (p/s), while the victim received requests at an average rate of **882.6 p/s**. This resulted in a packet amplification factor of **638** times.



TsuKing Attack (2/3): DNSChain

➢ Coordinating DRSes into a Resolution Chain

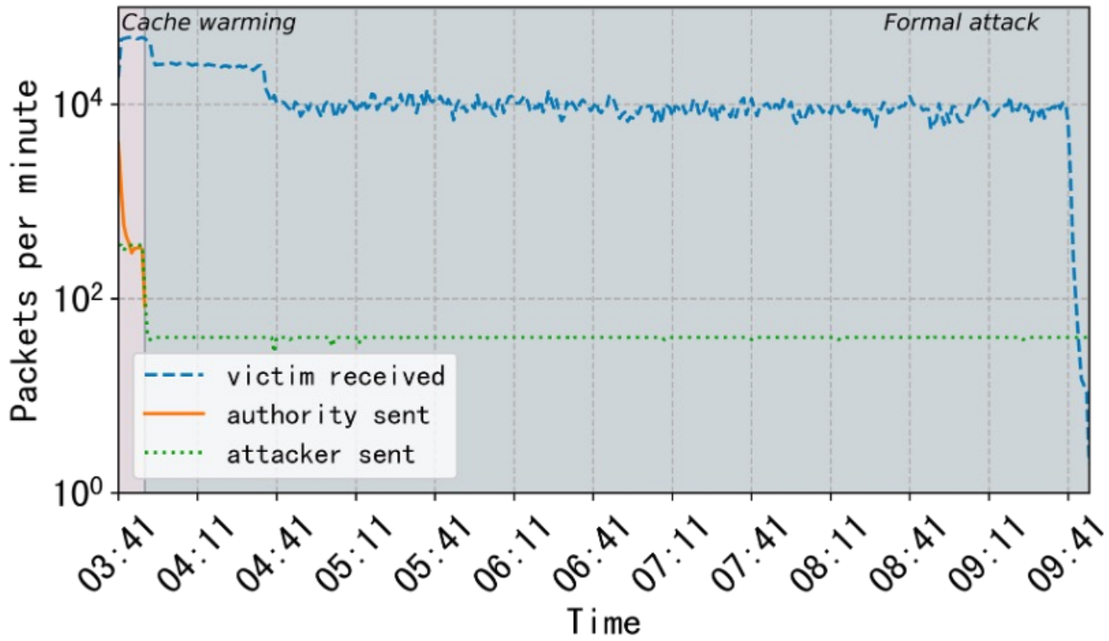
- ❑ By utilizing the core of TsuKing's combined scheduling for amplification, the **forwarding layers can be increased** to a certain extent.
- ❑ With a sufficient number of layers and DRSes, **all the malicious NS records** received by the outermost layer will point **towards the victim**.
- ❑ This forms a multi-layered **forwarding chain** with all DRSes, creating a powerful amplifier.



TsuKing Attack (2/3): DNSChain

➢ Experiment Results

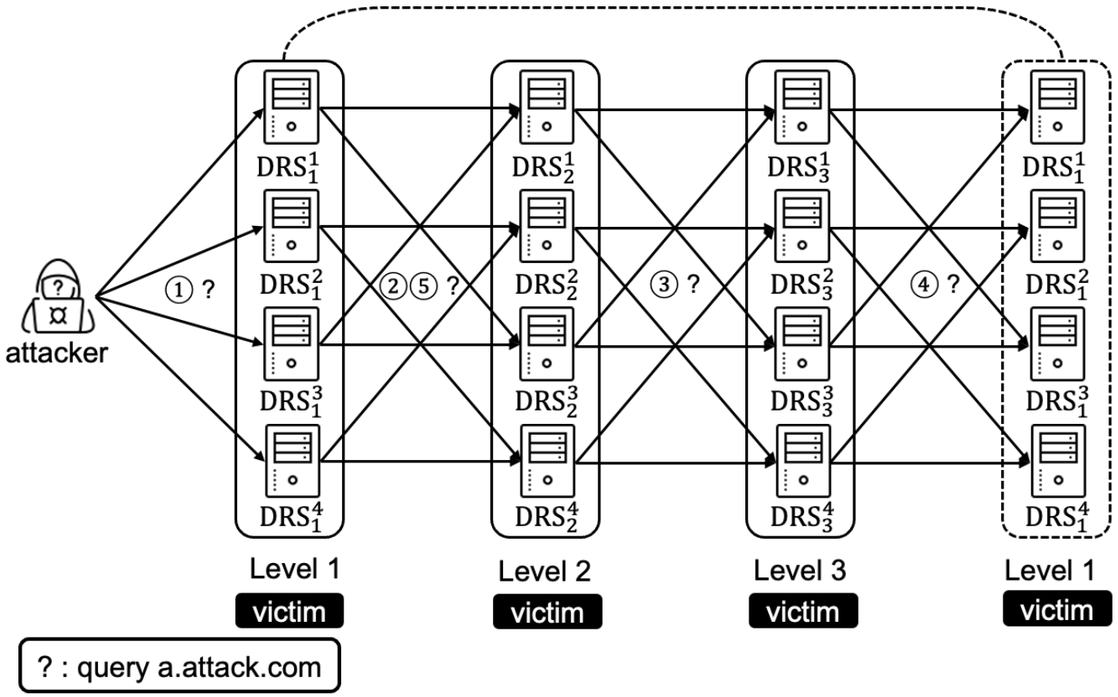
- ❑ A chain amplifier consisting of **253 vulnerable DRSes** was organized, reaching an amplification factor of **3,702** times across **7 levels**.
- ❑ In a specific small-scale experiment lasting **6 hours**, using a chain amplifier with 61 vulnerable targets across 5 levels, the attacker sent a total of **17,864** packets (at a rate of **0.8 p/s**), while the victim received **4,557,336** requests (at a rate of **206.4 p/s**), increased by **258 times**.



TsuKing Attack (3/3): DNSLoop

➢ Coordinating DRSes into a Resolution Loop

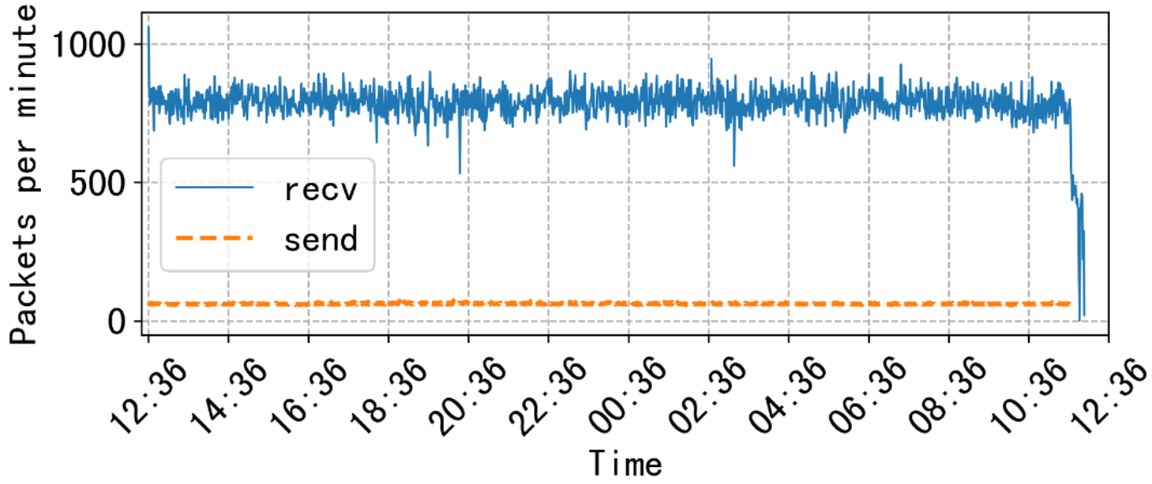
- By leveraging DNSChain as a foundation and **connecting the head and tail DRS**, the forwarding chain can be formed into a loop, creating a **DNSLoop** attack model.
- In the DNSLoop attack model, **any query** sent by the attacker within the loop will be perpetually **forwarded by the DRS servers within the loop**.
- As the attacker continuously **injects new queries** into the loop, the DRS servers within the loop **become increasingly burdened**, eventually leading to a denial-of-service (DoS) situation.



TsuKing Attack (3/3): DNSLoop

➢ Experiment Results

- ❑ In a real-world network experiment, a 7-level loop was constructed. The entire experiment **lasted for 24 hours** until it was manually stopped.
- ❑ Within the loop, our forwarders collectively sent **86,380** packets (at a rate of 1 p/s) and received **1,100,320** packets (at a rate of 12.7 p/s). This indicates that during the experiment, the requests were forwarded **43,190** times.



Mitigation

- **Honoring the RD Flag**
  - The key of TsuKing is not honoring the RD=0 flag, we recommend standardizing this implementation.
- **Implementing Negative Caching**
  - Negative caching can reduce the retry to a relatively small extent.
- **Avoiding Aggressive Retry**
  - Aggressive retry contributes to the part of TsuKing.
- **Optimizing Egress Schedule**
  - Non-interacting between different egress increases the amplifying impact.

Wrap-up

Thanks for listening!
Any questions?

Wei Xu, Xiang Li, Tsinghua University

xu-w21@mails.tsinghua.edu.cn
x-l19@mails.tsinghua.edu.cn