清華大學
Tsinghua University

Your Shield is My Sword: A Persistent Denial-of-Service Attack via the Reuse of Unvalidated Caches in DNSSEC Validation

Shuhan Zhang, Shuai Wang, Li Chen, Dan Li, Baojun Liu

Tsinghua University, Zhongguancun Laboratory

August 14, 2025

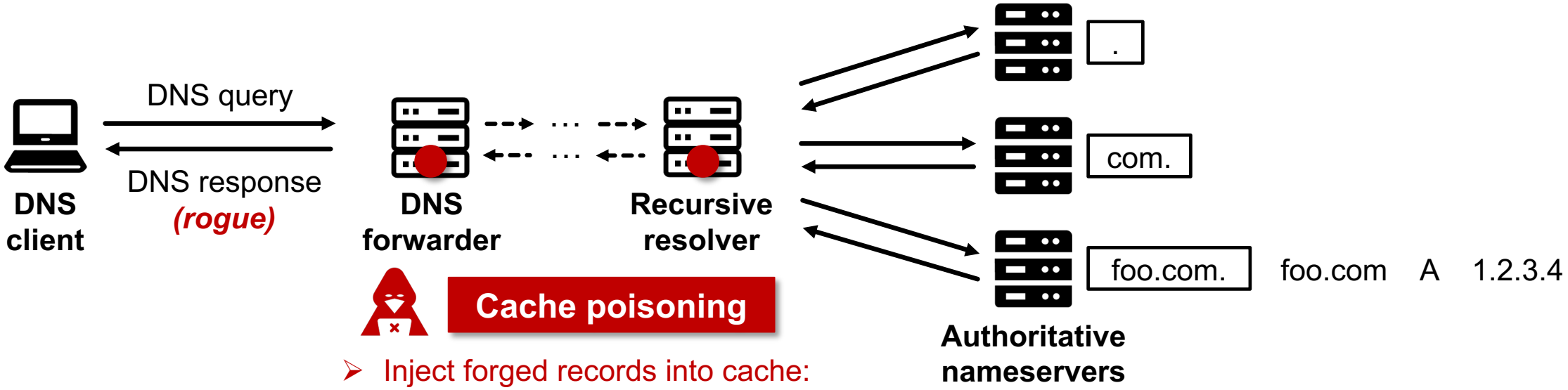
Our attack, RUC, turns DNSSEC from a *shield* to a *sword.*

RUC could persistently break the resolution of any domains under a DNSSEC-signed zone!
e.g., .com, .net, .org

DNS and DNSSEC

DNS was not initially designed with security considerations.

When a client requests foo.com…

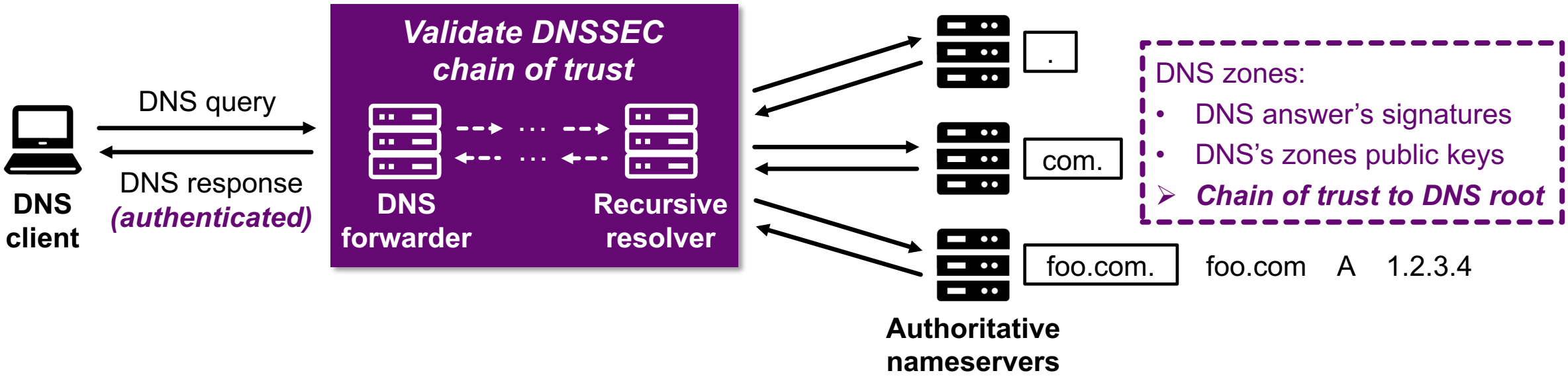


- Inject forged records into cache:
  foo.com A 6.6.6.6

DNS and DNSSEC

DNSSEC: ensure the authenticity and integrity of DNS data

When a client requests foo.com…

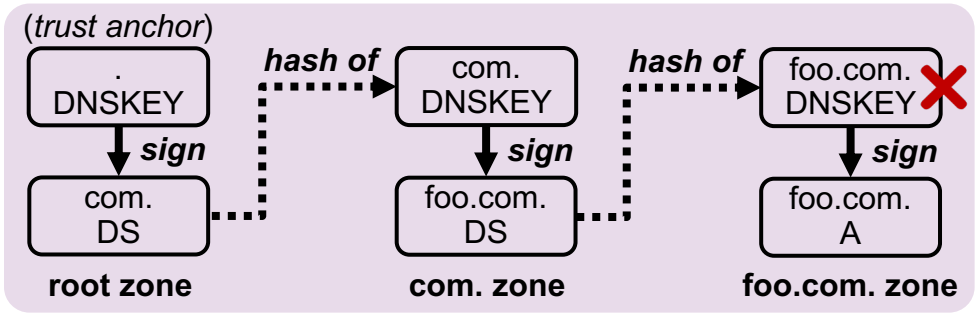


DNSSEC is a ***shield*** to protect DNS data against cache poisoning.

- 92.94% TLDs have been DNSSEC-signed (e.g., .com, .net, .org)
- Widely-used public DNS services enable validation by default (e.g., 8.8.8.8)

DNSSEC Troubleshooting

DNS resolvers allow the public to troubleshoot resource records, *during which DNSSEC validation is not enforced.*



Configuration complexity of the chain of trust

- Frequent misconfigurations lead to outages

Troubleshooting: set CD (Checking Disabled) in DNS query header

Ordinary query: CD=0 (default)

Resolver: validation fails

```
DNS response: SERVFAIL
(no answer)
```

🤔 Where is the problem??

Troubleshooting query: CD=1

Resolver: no validation

```
DNS response: NOERROR
foo.com  DNSKEY tx8EZ…
foo.com  RRSIG  DNSKEY (expired)
```

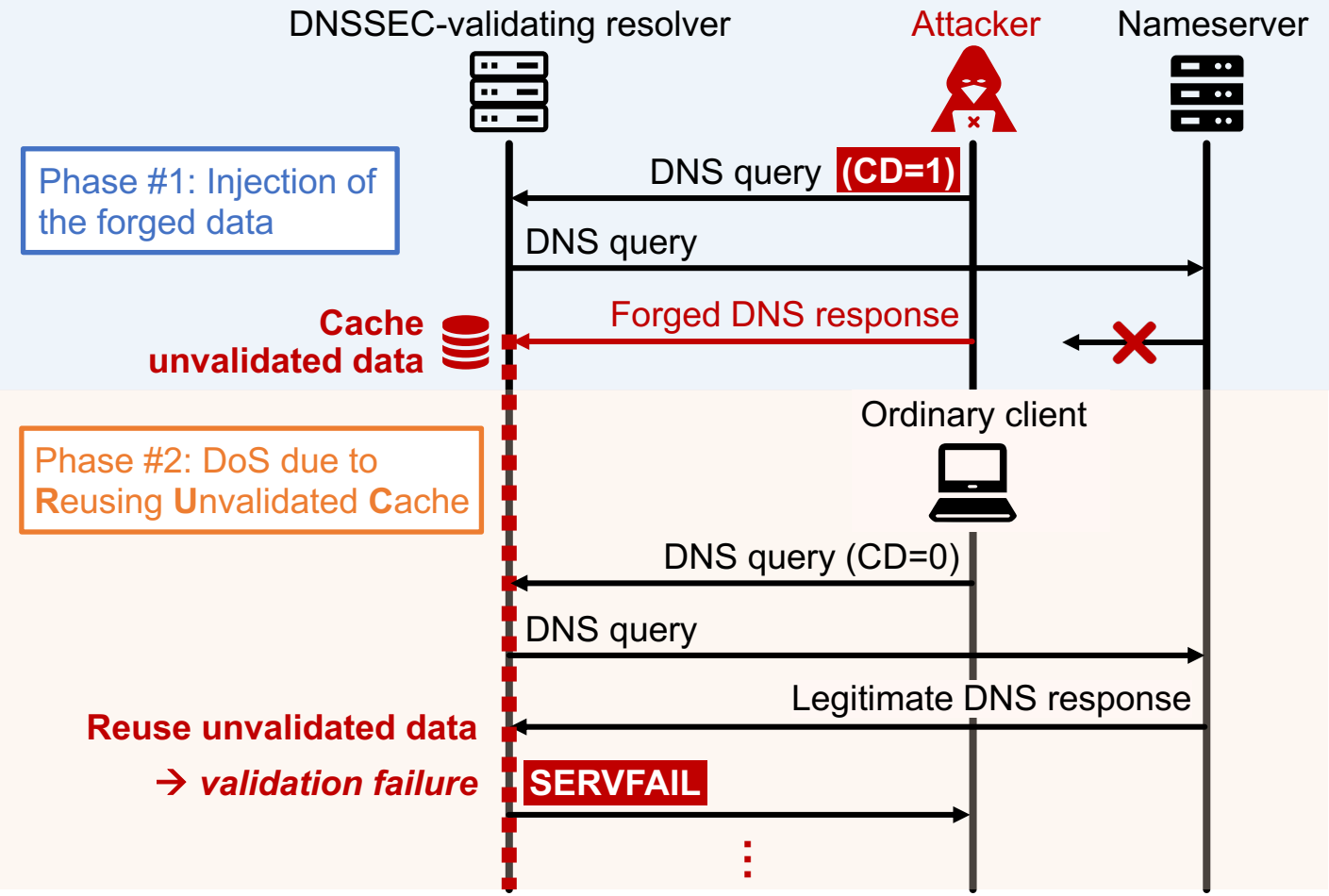
😉 I know! The RRSIG of foo.com's DNSKEY has expired.

Unfortunately, without clear specifications, DNS resolvers mix the caching and reusing of troubleshooting data with those in routine operations.

- DNSSEC can become a ***sword*** that thwarts domain resolution.

RUC: Reuse Unvalidated Caches

Exploitation: DNS resolvers cache data that have not passed DNSSEC validation, and reuse these cached, unvalidated data in subsequent resolutions.



A single cache injection via the troubleshooting query can trigger a persistent resolution DoS lasting for the caching TTL!

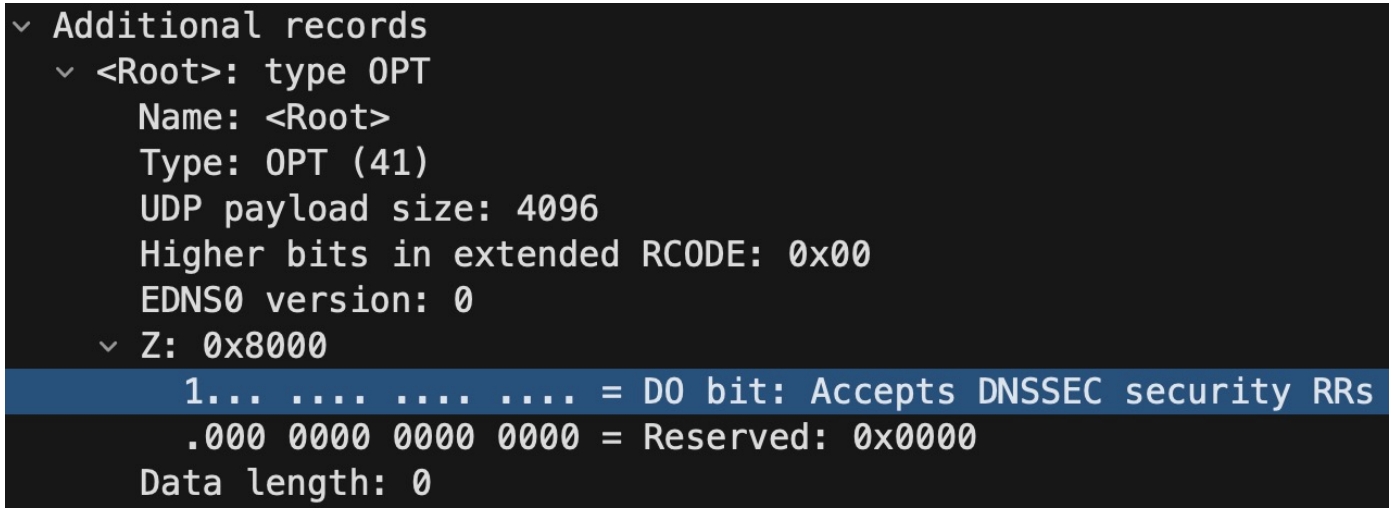
■ ■ ■ Duration of unvalidated cache

RUC Attack Variants

How RUC breaks resolver's DNSSEC validation:

- ✓ By forging records along domain's DNSSEC chain of trust
- ✓ By forging records of domain's nameserver
- ✓ By tricking the resolver into believing that domain's nameserver is not EDNS0*-capable

*EDNS0: indicated by the OPT record in the additional section, signaling DNSSEC capability via the DO (DNSSEC OK) bit.

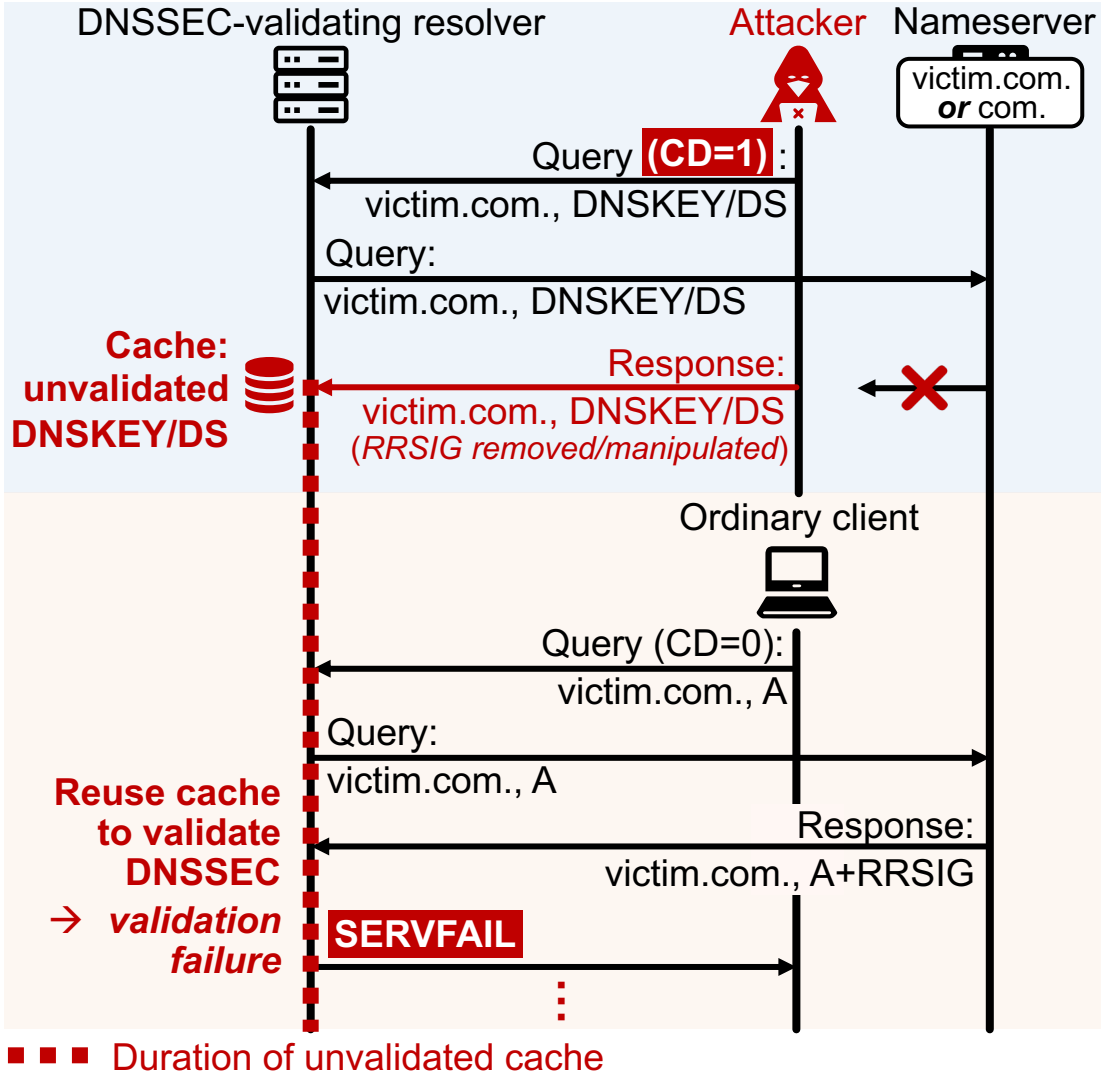


RUC Attack Variants

Variant #1: RUC_{SEC}

- ✓ **Exploited unvalidated cache:**
  DNSKEY or DS record

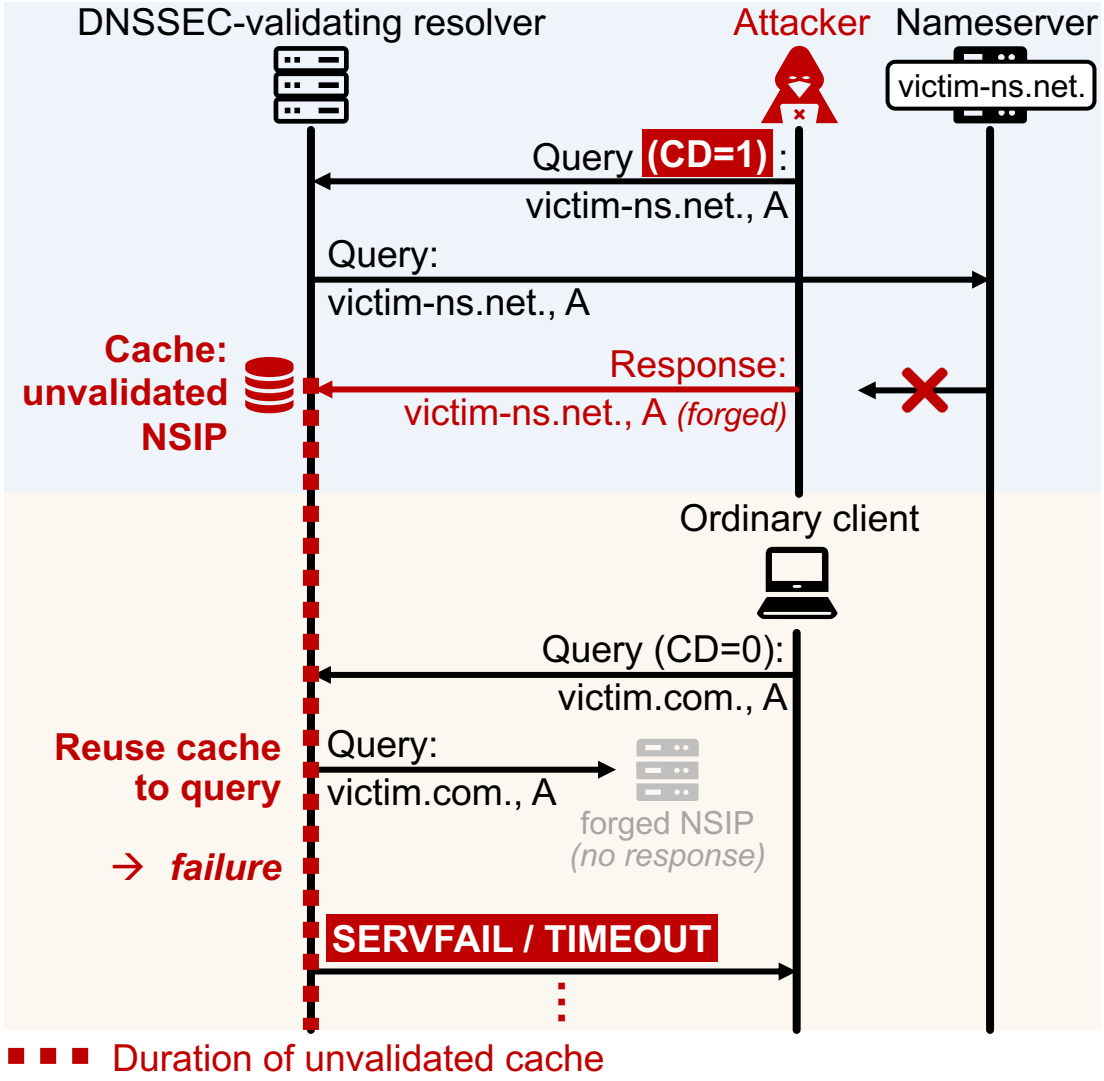
- ✓ **Victim zone:**
  Owner of the DNSKEY/DS record



RUC Attack Variants

Variant #2: RUC_{NSIP}

- ✓ **Exploited unvalidated cache:**
  Nameserver's authoritative A/AAAA record
- ✓ **Victim zone:**
  Zones delegating to the nameserver

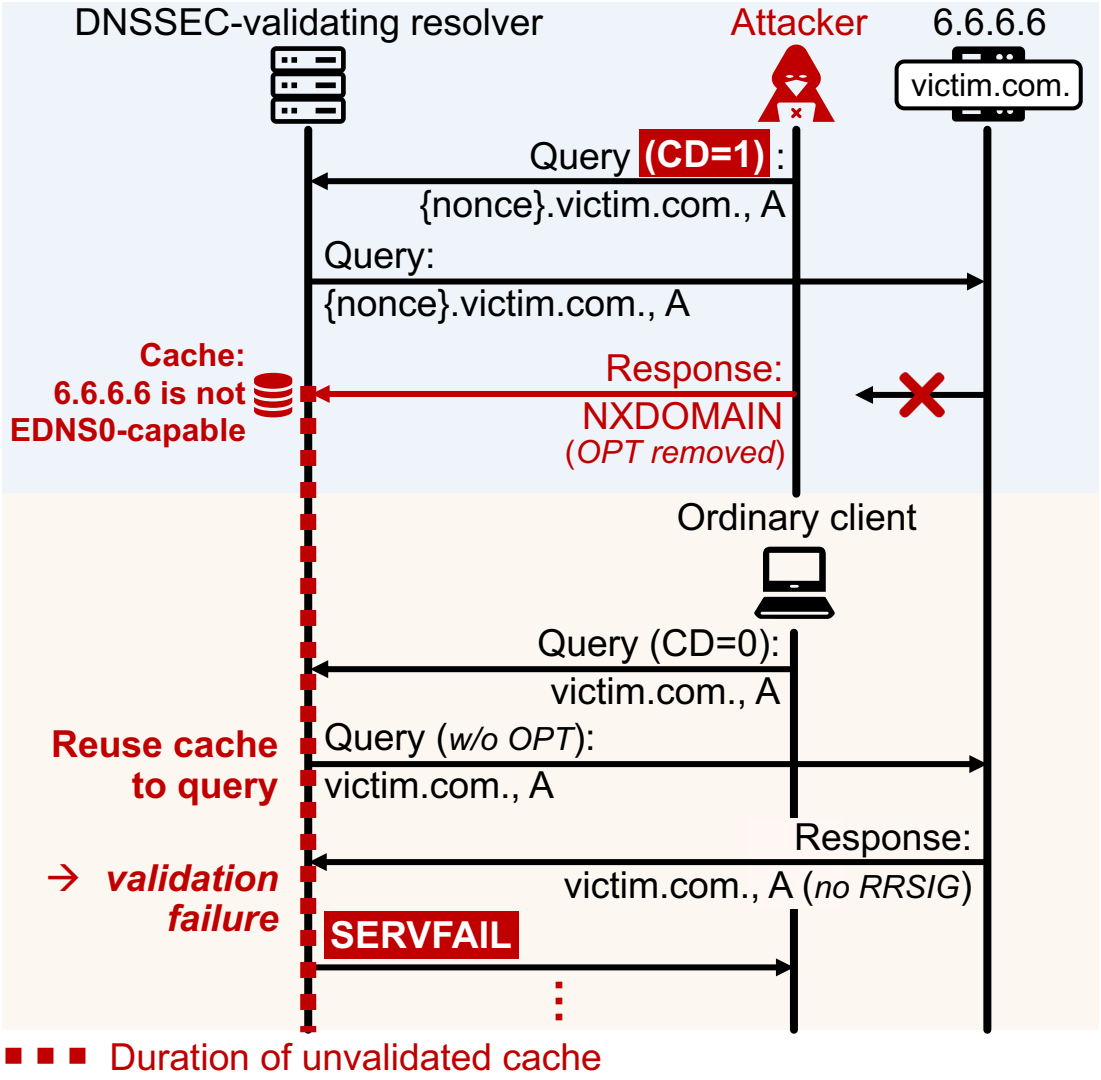


RUC Attack Variants

Variant #3: RUC_{EDNS0}

- ✓ **Exploited unvalidated cache:**
  EDNS0 capability of a nameserver host

- ✓ **Victim zone:**
  Zones delegating to the nameserver host



End-to-End RUC Attack

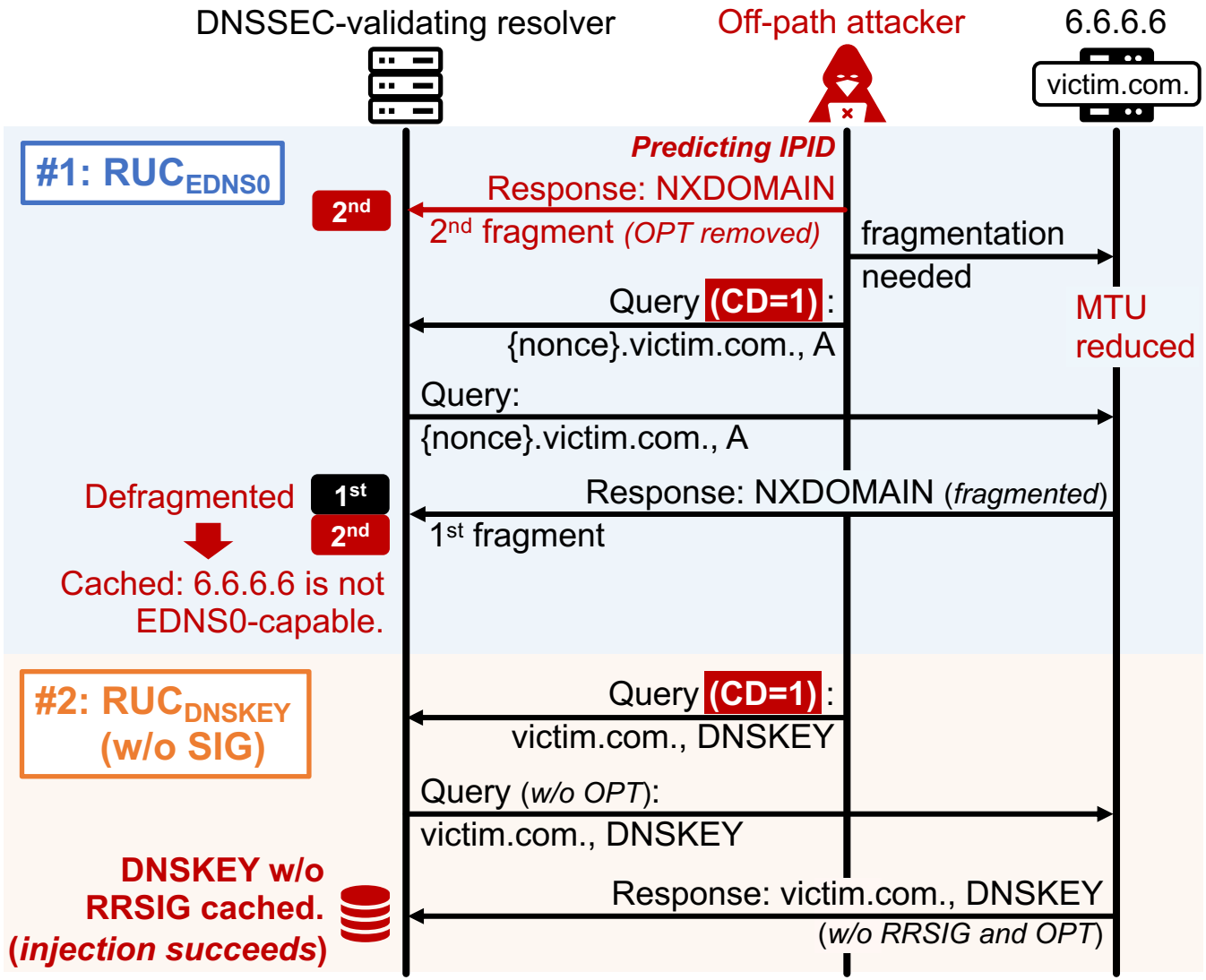
Off-path: via *response fragmentation*

- ✓ Fragment large DNSKEY responses[1]
- ✓ Fragment large NXDOMAIN responses[2]

On-path

- ✓ Nameserver takeover (even *partially,* i.e., 1 of multiple nameservers)
- ✓ BGP route hijacking (even *as short as a few seconds*)

[1] 60.8% TLDs and 69.9% SLDs at risk
[2] 87.5% TLDs and 83.0% SLDs at risk, even under the Ethernet MTU (1,500 bytes)



An example of off-path RUC attack via response fragmentation

RUC Vulnerability Evaluation

5 vulnerable mainstream DNS software: BIND, PowerDNS, etc.

DNS software	Tested version	Bypass DNSSEC validation?	Max TTL of validated records	Max TTL of unvalidated records	RUC_{SEC}				RUC_{NSIP}	RUC_{EDNS0}
					RUC_{DNSKEY}		RUC_{DS}			
					w/o SIG	w/ SIG	w/o SIG	w/ SIG		
BIND	9.20.3	✓	604,800	604,800	✓	✗	✓	✗	✓	✓
PowerDNS	5.1.3	✓	86,400	3,600	✓	✓	✓	✓	✓	✗
Microsoft DNS	2022	✗	86,400	180	✓	✓	✓	✓	✓	✗
Unbound	1.22.0	✗	86,400	60	✓	✓	✓	✓	✓	✗
Knot Resolver	5.7.4	✓	518,400	86,400	✗	✗	✗	✗	✓	✗
Technitium	13.1	✗	604,800	0	✗	✗	✗	✗	✗	✗

All tests were conducted under the default DNSSEC settings of each software. ✓: Yes. ✗: No. ✓: Vulnerable. ✗: Not vulnerable.

The most widely-used software, BIND, does not restrict the TTL for the unvalidated cache, making it possible to cache and reuse the unvalidated data for up to 7 days!

RUC Vulnerability Evaluation

28 vulnerable public DNS services: Cloudflare, Quad9, OpenDNS, etc.

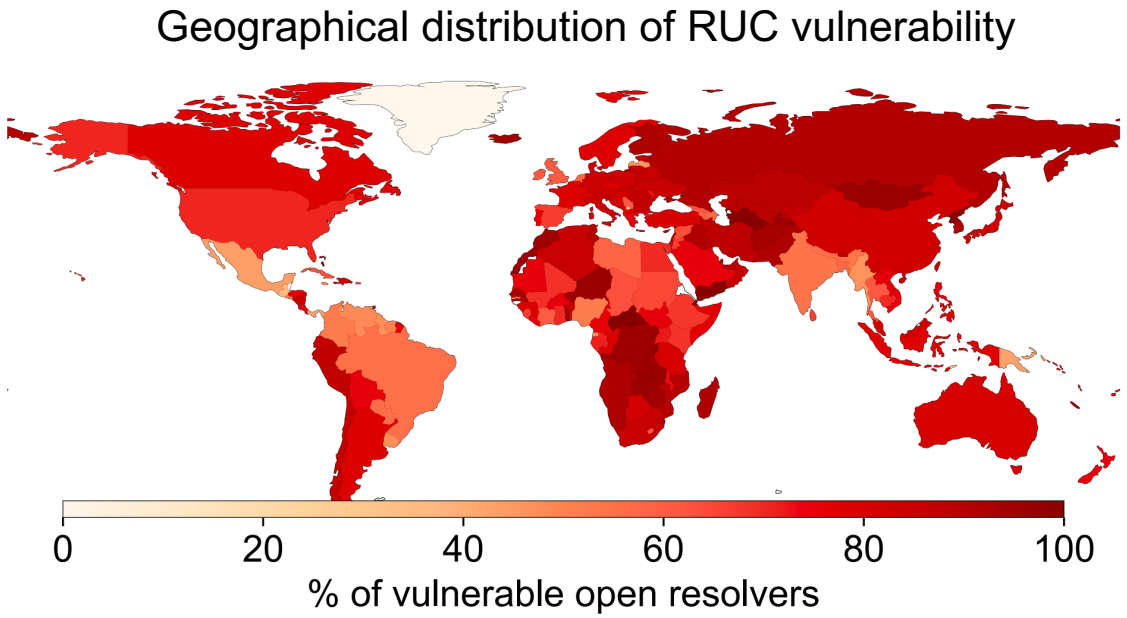
Public DNS service	Multiple cache?	Retain unvalidated records?	TTL of replied unvalidated records	RUC_{SEC}				RUC_{NSIP}	RUC_{EDNS0}
				RUC_{DNSKEY}		RUC_{DS}			
				w/o SIG	w/ SIG	w/o SIG	w/ SIG		
Google	✓	✓	21,600	✗	✗	✗	✗	4,565	✗
Cloudflare	✓	✓	86,400*	✗	✗	✗	✗	22,336	✗
Quad9	✓	✓	43,200	3,450	3,618	3,505	3,282	23,294	✗
OpenDNS	✓	✓	86,400*	626	608	604	612	86,521	✗
OpenNIC	✗	✗	86,400*	86,423	✗	86,470	✗	1,872	1,838
Quad101	✗	✗	14,400	14,468	✗	14,513	✗	1,813	1,804

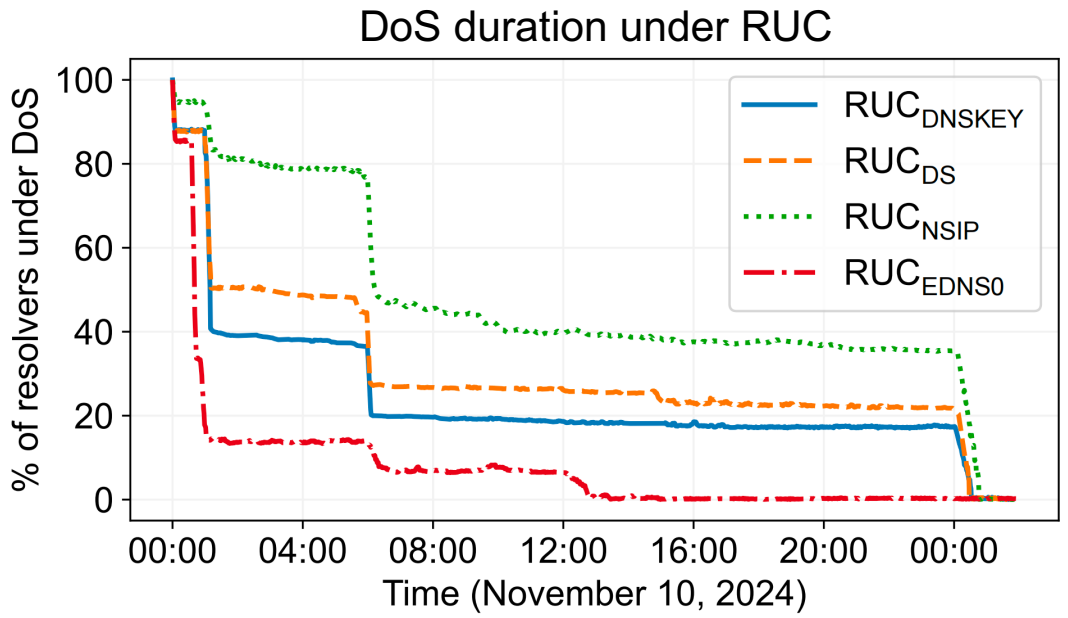
✓: Yes. ✗: No. # seconds: DoS duration in seconds. # seconds : DoS duration in seconds, which exceeds 5 minutes. ✗: Not vulnerable.

22 widely-used public DNS services could encounter DoS under RUC for >5 minutes.

RUC Vulnerability Evaluation

65.9% vulnerable DNSSEC-compliant open resolvers in the wild

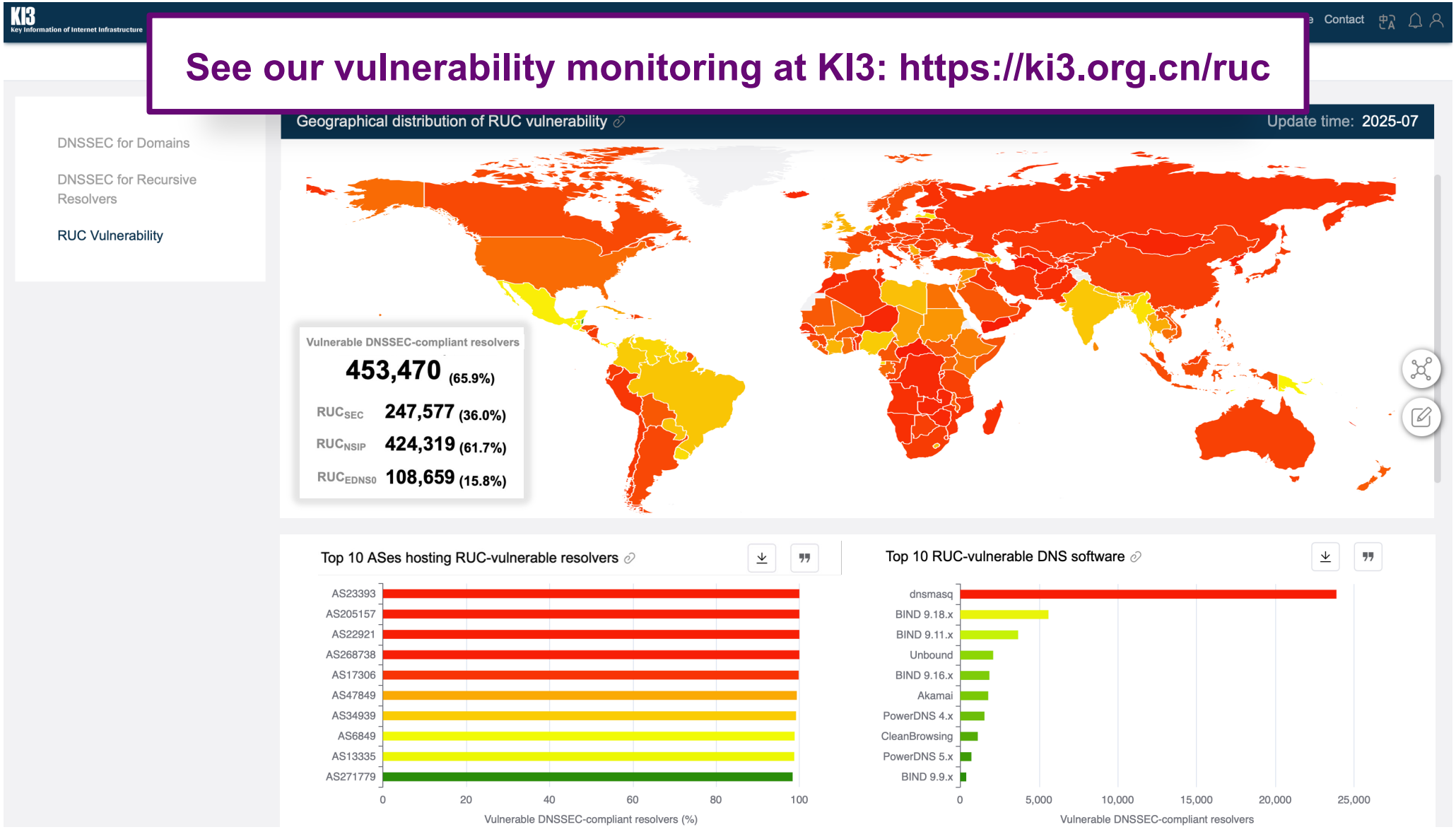




- **151 / 217 countries have more than 70% detected resolvers that are vulnerable to RUC.**
- **About 150k resolvers could encounter a DoS over 24 hours under RUC.**

RUC Vulnerability Monitoring

See our vulnerability monitoring at KI3: https://ki3.org.cn/ruc



Mitigation Suggestions

Caching unvalidated data

- ✓ Unvalidated records received during DNSSEC troubleshooting should be explicitly restricted for caching, e.g., setting a small TTL of no more than 5 minutes.

Reusing unvalidated data

- ✓ Unvalidated caches should only be reused when directly hit by queries that do not demand validation. Particularly, when the cached unvalidated records are indirectly involved in routine resolution, the resolver should strive to use their validated version.

Verifying nameserver's EDNS0 capability

- ✓ Resolvers should constantly verify the EDNS0 capability of nameserver hosts, and never remove the EDNS0 OPT record when processing DNSSEC-related queries.

Responsible Disclosure

Up to now, 3 renowned DNS vendors have acknowledged and patched their vulnerabilities:

- ✓ **BIND (version after April 2025):** prohibit the reuse of unvalidated records and the caching of nameserver EDNS0 capability.
- ✓ **Cloudflare (latest version):** develop heuristics to identify unvalidated, rogue cache entries.
- ✓ **OpenDNS (latest version):** reduce the caching TTL of unvalidated data.

Our IETF draft has been under discussion within the DNSOP working group!

- ✓ **Handling Unvalidated Data during DNSSEC Troubleshooting**

  https://datatracker.ietf.org/doc/draft-zhang-dnsop-dnssec-unvalidated-data/

Conclusion

Paper summary

- ✓ RUC: a new attack surface introduced via DNSSEC troubleshooting mechanism
- ✓ Comprehensive vulnerability evaluation: RUC affects 5 mainstream DNS software, 28 public DNS services and 65.9% DNSSEC-compliant open resolvers
- ✓ Responsible disclosure: 3 renowned DNS vendors have patched based on our suggestions

Key takeaways

- ✓ Troubleshooting mechanisms: indispensable, but lack formal guidelines
- ✓ Troubleshooting data could have severe but overlooked impact on production systems
- ✓ Troubleshooting data should be clearly segregated from data in routine operations

清華大學
Tsinghua University

THANKS

- **Website:** https://ki3.org.cn/ruc
- **Code:** https://zenodo.org/records/15543846
- **IETF draft:** https://datatracker.ietf.org/doc/draft-zhang-dnsop-dnssec-unvalidated-data/
- **Contact:** zhangsh22@mails.tsinghua.edu.cn