How the Internet Dodged a Bullet: The KeyTrap Denial-of-Service Attacks against DNSSEC

Speaker(s):
Elias Heftrig, Niklas Vogel

Contributors:

Haya Schulmann, Michael Waidner

Refresher: DNS and DNSSEC

Why is it always DNS?

DDoS attack that disrupted internet was largest of its kind in history, experts say

Dyn, the victim of last week's denial of service attack, said it was orchestrated using a weapon called the Mirai botnet as the 'primary source of malicious attack'

Understanding how Facebook disappeared from the Internet

2021-10-04

DNS poisoning slams web traffic from millions in China into the wrong hole

ISP blames unspecified attack for morning outage

Home > News > Security > Akamai DNS global outage takes down major websites, online services

Akamai DNS global outage takes down major websites, online services

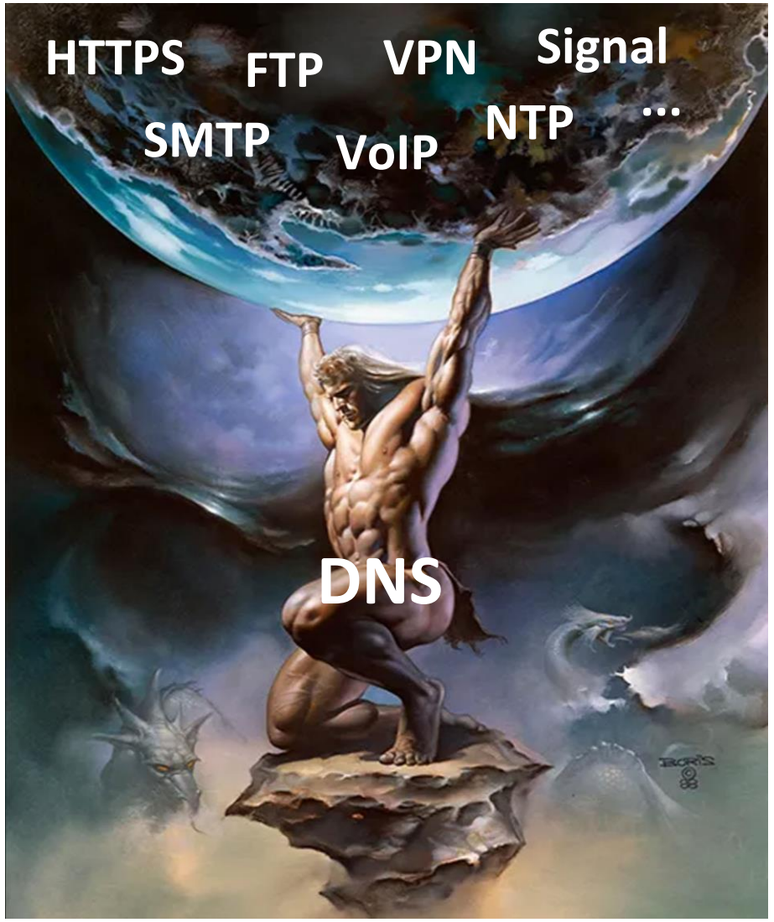
By **Sergiu Gatlan** July 22, 2021 12:39 PM 2

Salesforce cloud services go down worldwide

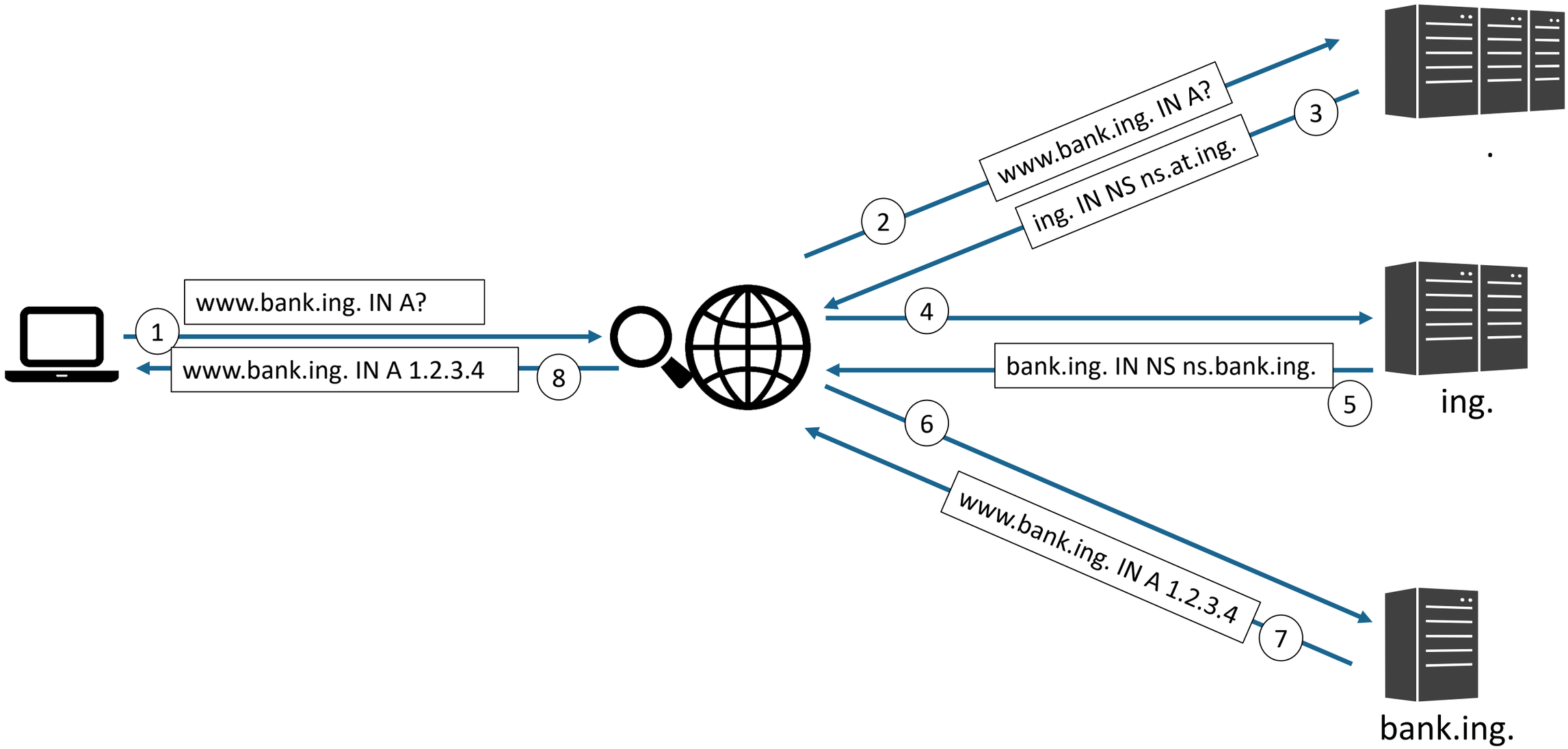
By Juha Saarinen
May 12 2021 9:06AM

Caused by DNS issue.

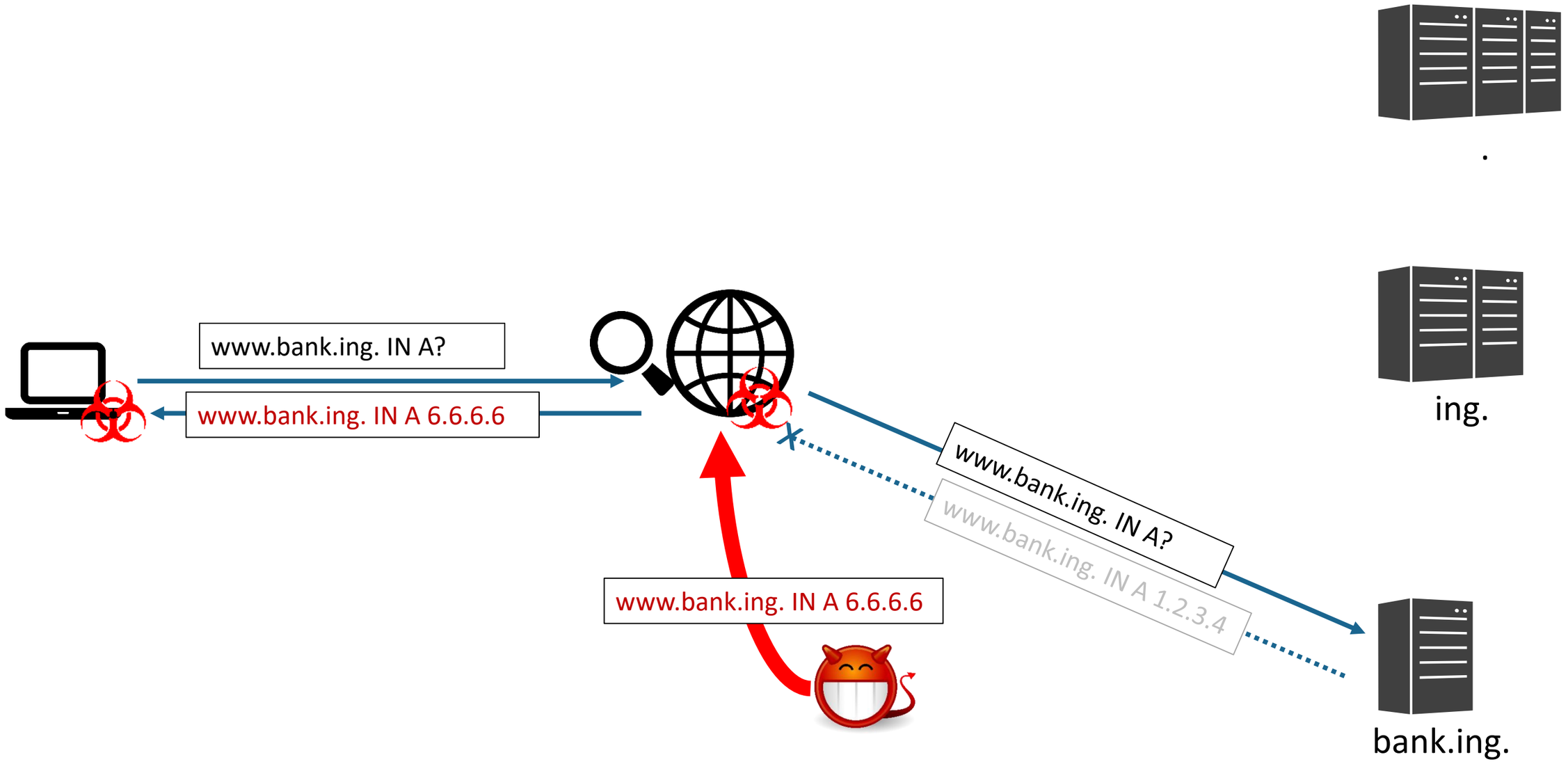
Why is it always DNS?



DNS Resolution

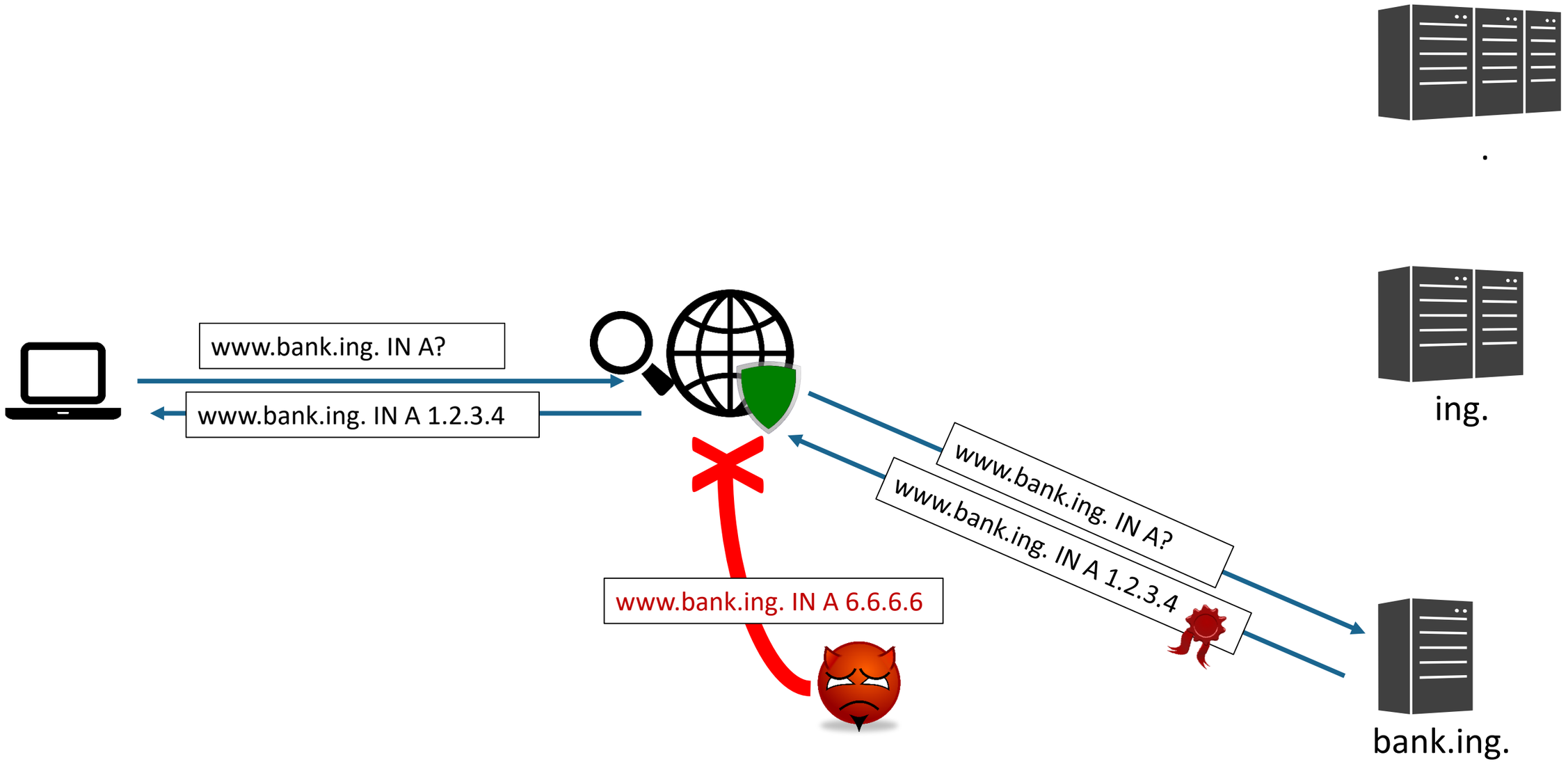


DNS Poisoning



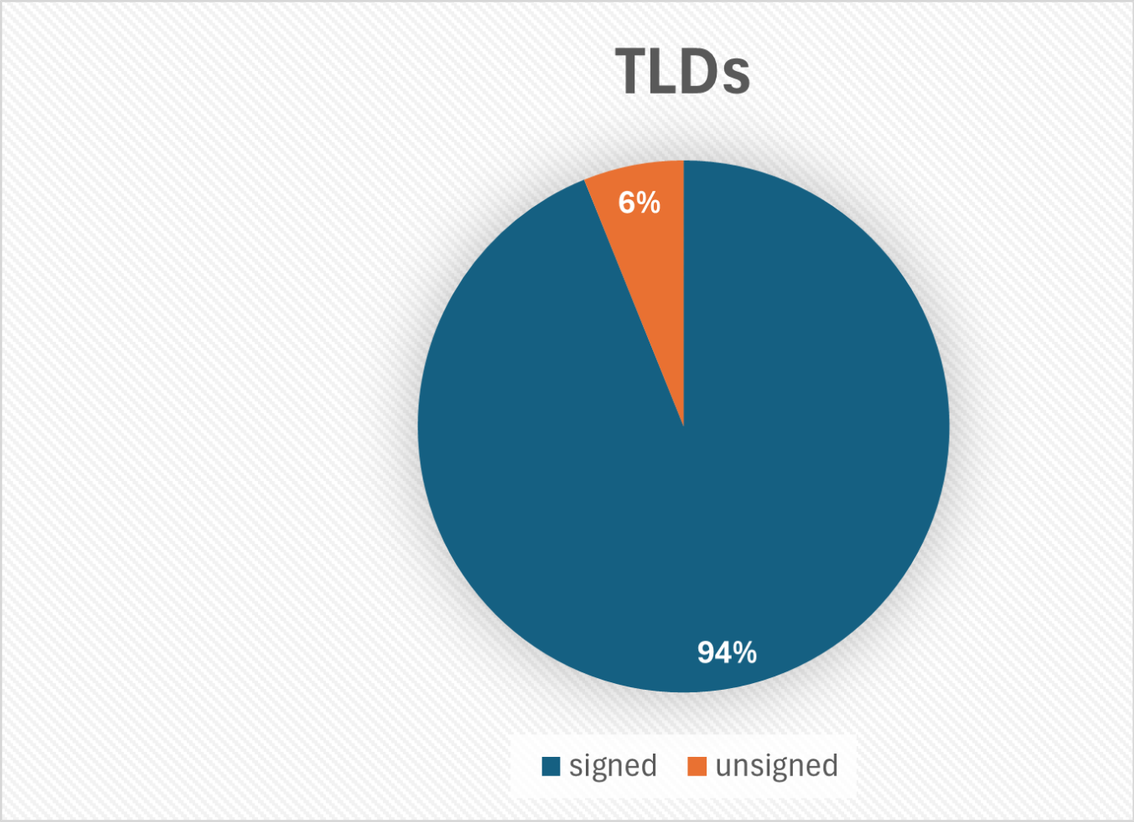
- Attack on DNS Record Authenticity

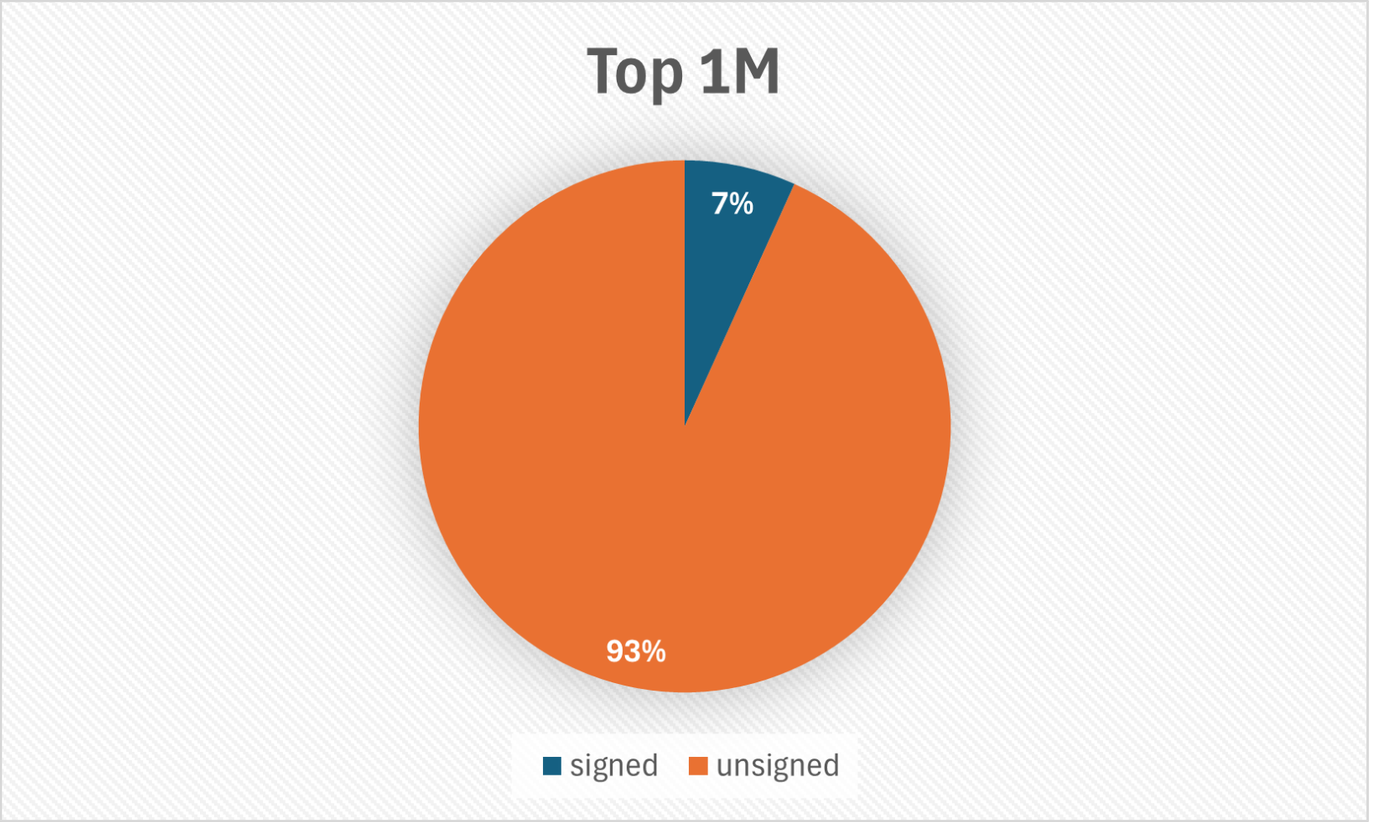
DNSSEC to the Rescue!



- Attack prevented

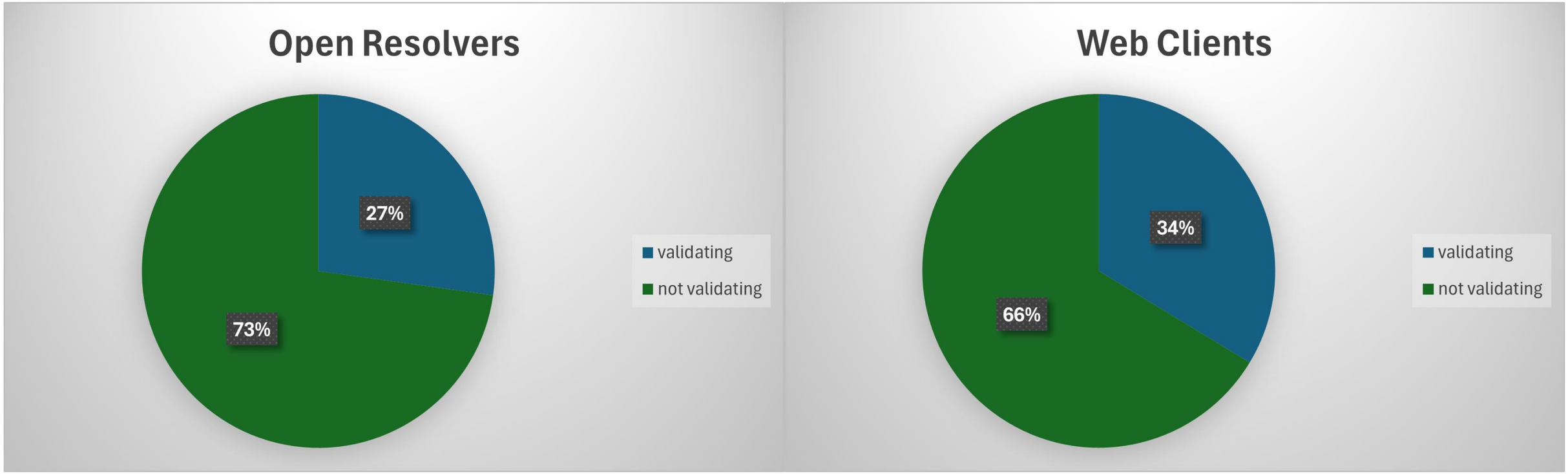
DNSSEC Adoption on the Internet





Adoption in domains is dragging

DNSSEC Adoption on the Internet



Better adoption in Resolvers

How DNSSEC Validation Works

```
~$ dig www.ietf.org -t A +dnssec
```

What is the IP Address of www.ietf.org?

How DNSSEC Validation Works

```
~$ dig www.ietf.org -t A +dnssec
```

What is the IP Address of www.ietf.org?

```
;; ANSWER SECTION:
www.ietf.org.           300     IN      A       104.16.45.99
www.ietf.org.           300     IN      A       104.16.44.99
www.ietf.org.           300     IN      RRSIG A 13 3 300 20241211 20241209
      ↳ 34505 ietf.org. eSTHK9ql[…]uvSgBA==
```

How DNSSEC Validation Works

```
~$ dig www.ietf.org -t A +dnssec
```

What is the IP Address of www.ietf.org?

How to validate??

```
;; ANSWER SECTION:
www.ietf.org.          300    IN    A       104.16.45.99
www.ietf.org.          300    IN    A       104.16.44.99
www.ietf.org.          300    IN    RRSIG A 13 3 300 20241211 20241209
      ↳ 34505 ietf.org. eSTHK9ql[...]uvSgBA==
```

Obtaining Public Keys

```
~$ dig ietf.org -t DNSKEY +dnssec
```

What are the keys for ietf.org?

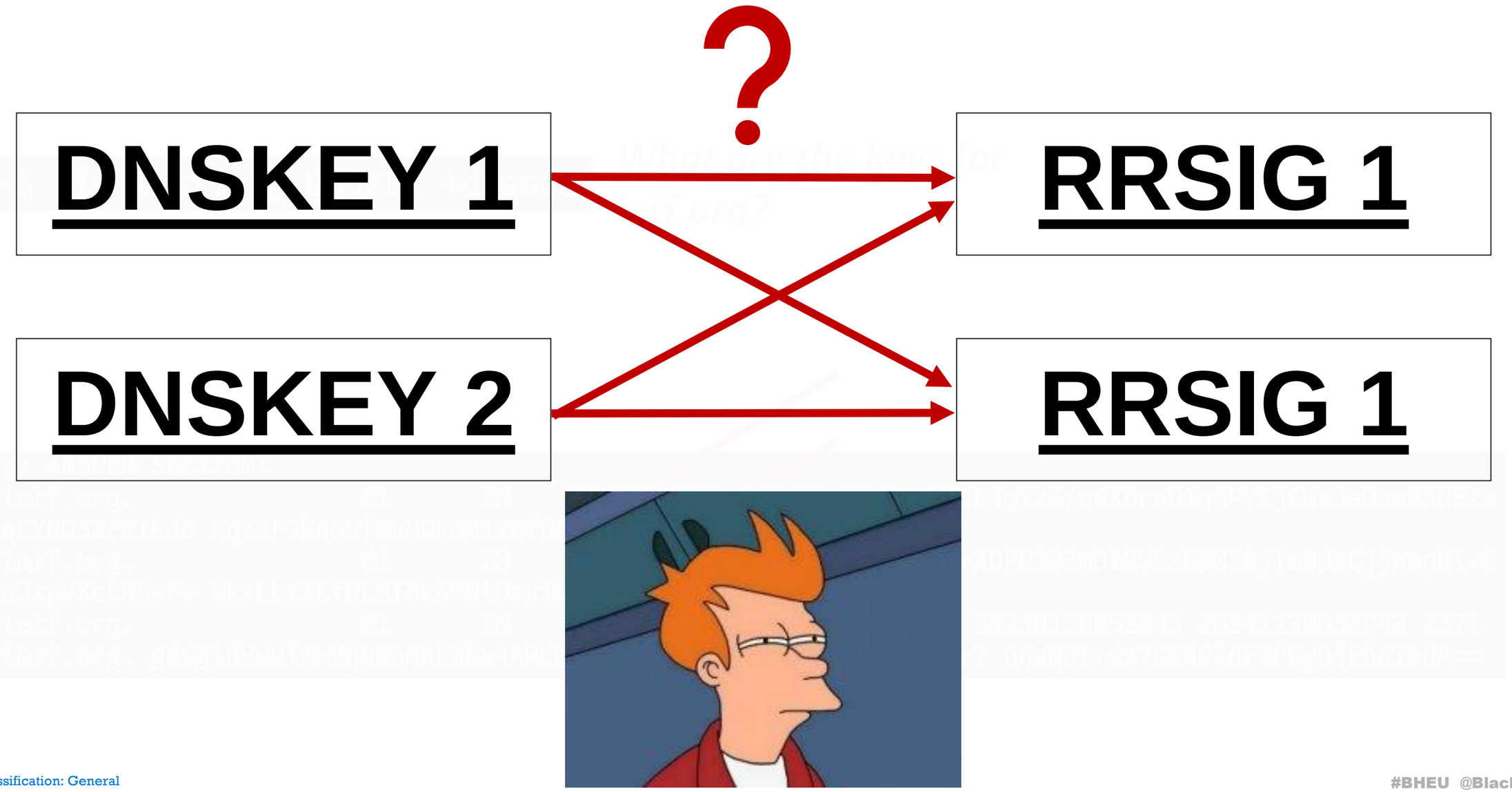
Obtaining Public Keys

```
~$ dig ietf.org -t DNSKEY +dnssec
```

What are the keys for ietf.org?

```
;; ANSWER SECTION:
ietf.org.           3600    IN      DNSKEY  257 3 13 mdsswUyr[…]53eKGQ==
ietf.org.           3600    IN      DNSKEY  256 3 13 oJMRESz5[…]a2XhSA==
ietf.org.           3600    IN      RRSIG   DNSKEY 13 2 3600 20250130 20241130
      ↳ 2371 ietf.org. gdCgidVw[…]heIodA==
```

Associating Signatures with Keys



Associating Signatures with Keys

```
;; ANSWER SECTION:
ietf.org.               3600    IN      DNSKEY  257 3 13 mdsswUyr[…]53eKGQ==
ietf.org.               3600    IN      DNSKEY  256 3 13 oJMRESz5[…]a2XhSA==
```

```
;; ANSWER SECTION:
www.ietf.org.           300     IN      RRSIG A 13 3 300 20241211 20241209
      ↳ 34505 ietf.org. eSTHK9qI[…]uvSgBA==
```

Associating Signatures with Keys

```
;; ANSWER SECTION:
ietf.org.                [redacted]  [redacted]  DNSKEY  [redacted]  13  [redacted]
ietf.org.                [redacted]  [redacted]  DNSKEY  [redacted]  13  [redacted]
```

```
;; ANSWER SECTION:
www.ietf.org.            [redacted]  [redacted]  RRSIG  [redacted]  13  [redacted]  [redacted]  [redacted]
        ↳ 34505 ietf.org. [redacted]
```

Associating Signatures with Keys

DNSKEY 1: ietf.org | Algo 13 | [KqXX...]

DNSKEY 2: ietf.org | Algo 13 | [KkxL...]

Signature: ietf.org | Algo 13 | Key Tag 34505

```
;; ANSWER SECTION:
ietf.org.                [redacted]  DNSKEY  [redacted]  13  [redacted]
ietf.org.                [redacted]  DNSKEY  [redacted]  13  [redacted]

;; ANSWER SECTION:
www.ietf.org.            [redacted]  RRSIG   [redacted]  13  [redacted]
        ↳ 34505 ietf.org.  [redacted]
```

Associating Signatures with Keys

DNSKEY 1: ietf.org | Algo 13 | f(KqXX...) = 2371

DNSKEY 2: ietf.org | Algo 13 | f(KkxL...) = 34505

Signature: ietf.org | Algo 13 | Key Tag 34505

```
;; ANSWER SECTION:
ietf.org.                DNSKEY    13
ietf.org.                DNSKEY    13
```

```
;; ANSWER SECTION:
www.ietf.org.            RRSIG  13
      ↳ 34505 ietf.org.
```

Associating Signatures with Keys

DNSKEY 1: ietf.org | Algo 13 | f(KqXX...) = 2371

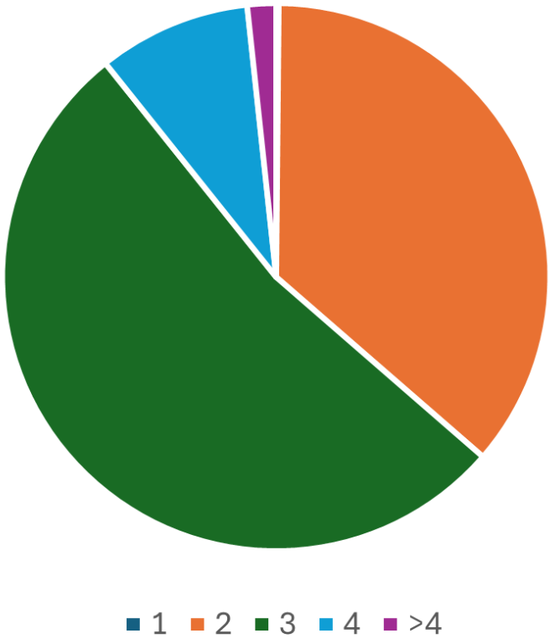
DNSKEY 2: ietf.org | Algo 13 | f(KkxL...) = **34505**

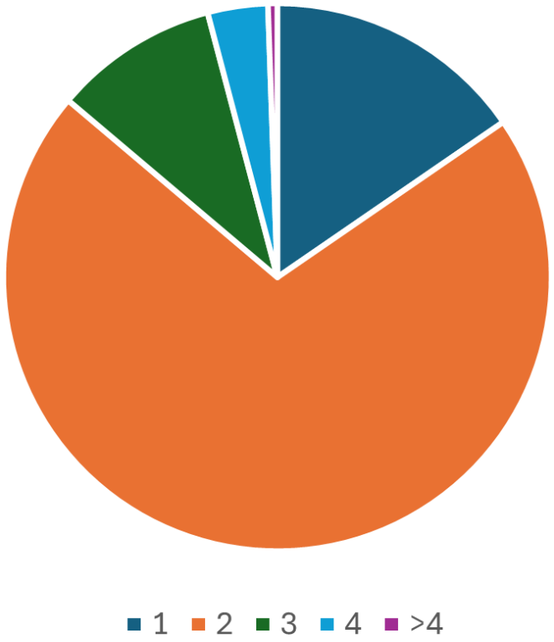
Signature: ietf.org | Algo 13 | Key Tag **34505**

```
;; ANSWER SECTION:
ietf.org.                    DNSKEY    13
ietf.org.                    DNSKEY    13
```

```
;; ANSWER SECTION:
www.ietf.org.                RRSIG     13
    ↳ 34505 ietf.org.
```

Frequencies of Keys in Domains





Association of Keys and Signatures is efficient under normal conditions.

KeyTrap Attacks

Associating Signatures with Keys

DNSKEY 1: ietf.org | Algo 13 | f(KqXX...) = 2371

DNSKEY 2: ietf.org | Algo 13 | f(KkxL...) = 34505

Signature: ietf.org | Algo 13 | Key Tag **34505**

```
;; ANSWER SECTION:
ietf.org.                DNSKEY  13
ietf.org.                DNSKEY  13
```

```
;; ANSWER SECTION:
www.ietf.org.            RRSIG   13
      ↳ 34505 ietf.org.
```

Colliding Key Tags

DNSKEY 1: ietf.org | Algo 13 | f(KqXX...) = **2371**

DNSKEY 2: ietf.org | Algo 13 | f(KkxL...) = 34505

Signature: ietf.org | Algo 13 | Key Tag **34505**

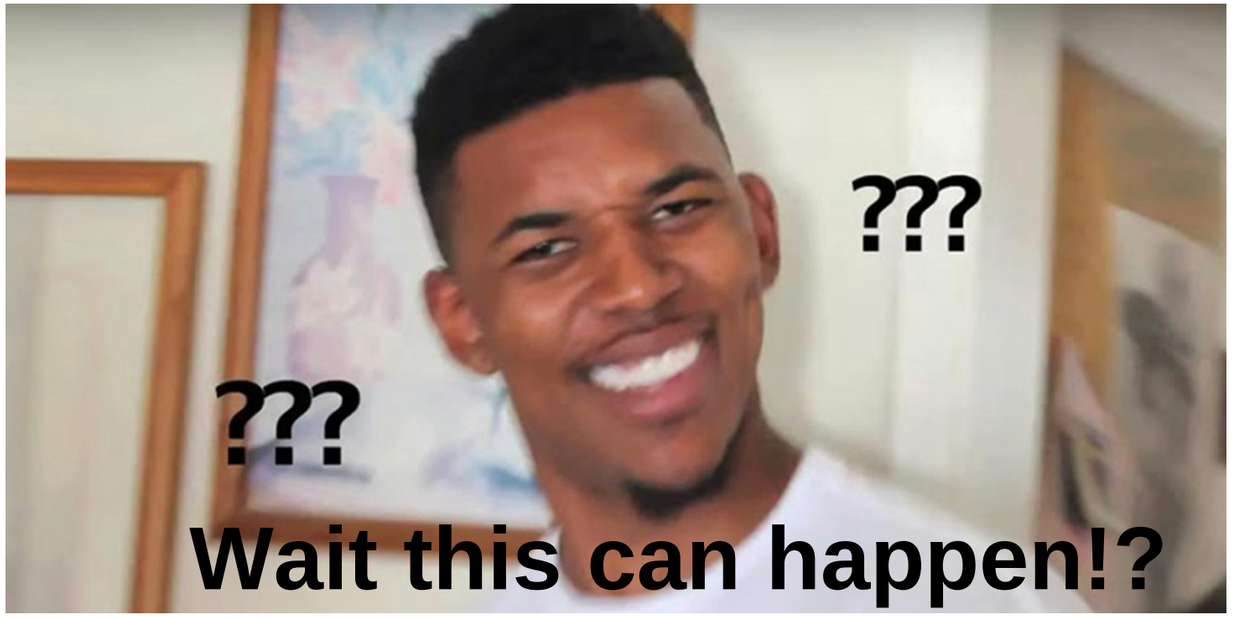
```
;; ANSWER SECTION:
ietf.org.                    DNSKEY    13
ietf.org.                    DNSKEY    13

;; ANSWER SECTION:
www.ietf.org.                RRSIG     13
    ↳ 34505 ietf.org.
```

Colliding Key Tags

DNSKEY 1: ietf.org | Algo 13 | f(KqXX...) = **34505**

DNSKEY 2: ietf.org | Algo 13 | f(KkxL...) = **34505**

Signature: ietf.org | Algo 13 | Key Tag **34505**



A Closer Look at Signature Validation

However, it is essential to note that the key tag is not a unique identifier. It is theoretically possible for two distinct DNSKEY RRs to have the same owner name, the same algorithm, and the same key tag. The key tag is used to limit the possible candidate keys, but it does not uniquely identify a DNSKEY record. Implementations MUST NOT assume that the key tag uniquely identifies a DNSKEY RR.

RFC4034, Appendix B "Key Tag Calculation"

A Closer Look at Signature Validation

However, it is essential to note that the key tag is not a unique identifier. It is theoretically possible for two distinct DNSKEY RRs to have the same owner name, the same algorithm, and the same key tag. The key tag is used to limit the possible candidate keys, but it does not uniquely identify a DNSKEY record. Implementations MUST NOT assume that the key tag uniquely identifies a DNSKEY RR.

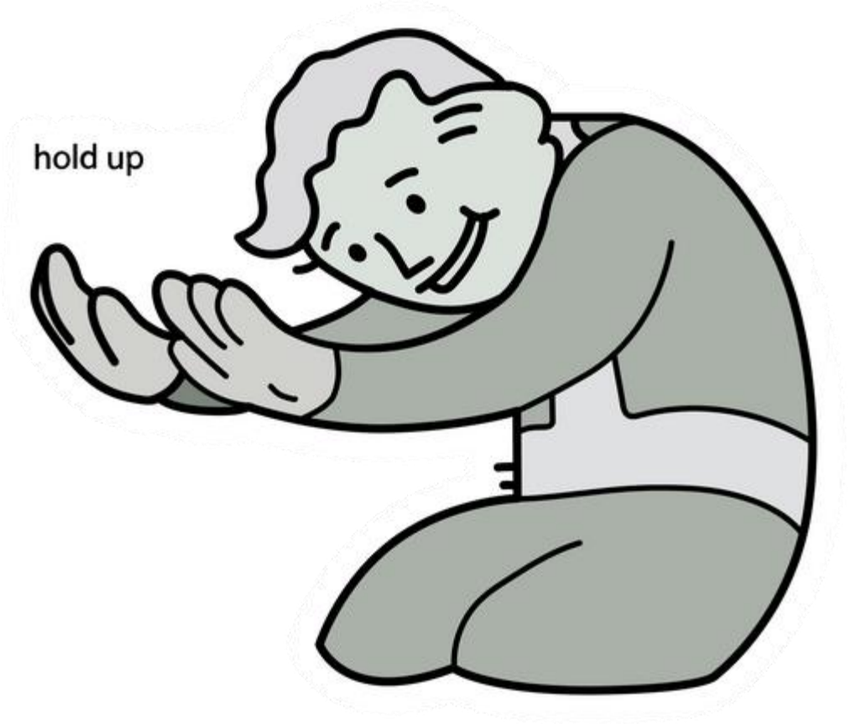
RFC4034, Appendix B "Key Tag Calculation"

It is possible for more than one DNSKEY RR to match the conditions above. In this case, the validator cannot predetermine which DNSKEY RR to use to authenticate the signature, and it MUST try each matching DNSKEY RR until either the signature is validated or the validator has run out of matching public keys to try.

RFC4035, Section 5.3.1. "Checking the RRSIG RR Validity"

A Closer Look at Signature Validation

A Closer Look at Signature Validation



- Linear time algorithm
- Resource-intensive public key crypto operations
- Resolver MUST try all keys?

That sounds like a bad idea...

A Closer Look at Signature Validation

- Linear time algorithm
- Resouce intensive public key crypto operations
- Resolver MUST try all keys?

That sounds like a bad idea...

Wait, wasn't there more like this?

We can make it even worse

This document specifies that a resolver SHOULD accept any valid RRSIG as sufficient, and only determine that an RRset is Bogus if all RRSIGs fail validation.
If a resolver adopts a more restrictive policy, there's a danger that properly signed data might unnecessarily fail validation due to cache timing issues. Furthermore, certain zone management techniques, like the Double Signature Zone Signing Key Rollover method described in Section 4.2.1.2 of [RFC6781], will not work reliably. Such a resolver is also vulnerable to malicious insertion of gibberish signatures.

RFC6840 Section 5.4. "Caution about Local Policy and Multiple RRSIGs"

It’s quadratic!

This document specifies that a resolver SHOULD accept any valid RRSIG as sufficient, and only determine that an RRset is Bogus if all RRSIGs fail validation.
If a resolver adopts a more restrictive policy, there’s a danger that properly signed data might unnecessarily fail validation due to cache timing issues. Furthermore, certain zone management techniques, like the Double Signature Zone Signing Key Rollover method described in Section 4.2.1.2 of [RFC6781], will not work reliably. Such a resolver is also vulnerable to malicious insertion of gibberish signatures.

RFC6840 Section 5.4. "Caution about Local Policy and Multiple RRSIGs"

Open Source is doing this...

```
// Get amount of signatures on the rrset
num = rrset_get_sigcount(rrset);

// Iterate all Signatures
for (i = 0; i < num; i++)
{
    sec = dnskeyset_verify_rrset_sig(env, ve, *env->now, rrset,
                                     dnskey, i, &sortree, reason, reason_bogus,
                                     section, qstate);
```

Signature Loop

https://github.com/NLnetLabs/unbound/blob/master/validator/val_sigcrypt.c (line 704)

```
// Get amount of keys in keyset
size_t i, num = rrset_get_count(dnskey);

// Iterate over all keys for the set
for (i = 0; i < num; i++)
{
    /* see if key matches keytag and algo */
    if (algo != dnskey_get_algo(dnskey, i) ||
        tag != dnskey_calc_keytag(dnskey, i))
        continue;

    sec = dnskey_verify_rrset_sig(env->scratch,
                                  env->scratch_buffer, ve, now, rrset, dnskey, i,
                                  sig_idx, sortree, &buf_canon, reason, reason_bogus,
                                  section, qstate);

    if (sec == sec_status_secure)
        return sec;
    else if (sec == sec_status_indeterminate)
        numindeterminate++;
}
```

Key Loop

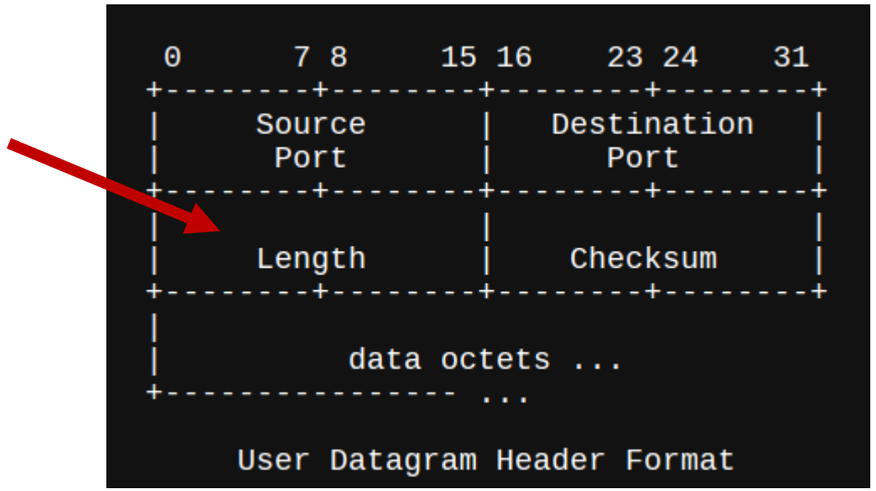
https://github.com/NLnetLabs/unbound/blob/master/validator/val_sigcrypt.c (line 641)

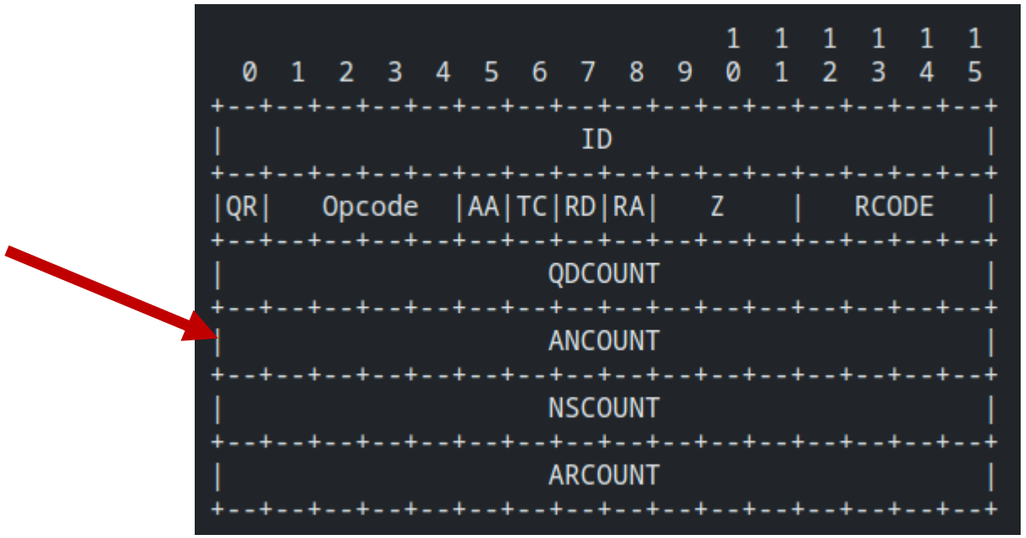
Nested Loops!

- Limited buffers?
- Other Mitigations in place?

→ there is only one way to find out

How much can “all” be?





Not many restrictions in the DNS protocol

Quick'n'Dirty Maths

Size DNS msg. 65 kB

Keys: 80 Byte => 65.000 // 80 ~ 820 p. msg.

Sigs: 110 Byte => 65.000 // 110 ~ 600 p. msg.

Validations: Keys x Sigs => 820 x 600 ~ 490.000

Time per Validation: 150 µs [1]

Stall from 1 req: 490.000 x 150 µs ~ 74 s

[1] https://blog.cloudflare.com/go-crypto-bridging-the-performance-gap/

Let's construct an attack...

How to obtain lots of DNSKEYs with the same tag?

➔ Generate many and keep only those conforming to a specific one

Zone File Contents: DNSKEYs

```
keytrap.org 60 IN DNSKEY 257 3 13 YpsCtFxj[...]O/G81g== ;{id=26539 (ksk)}

keytrap.org 60 IN DNSKEY 256 3 13 wwZzOFux[...]7s+jGg== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 AHl8SDyd[...]vgk3gQ== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 8GP1H4eS[...]3X6dCA== ;{id=1337 (zsk)}
                                  ...
keytrap.org 60 IN DNSKEY 256 3 13 brtDSnWm[...]nzlKOw== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 WfKVRBtM[...]eC3Alw== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 6+Gbtx4h[...]RacDLw== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 6KNR3F+Q[...]RcK2kQ== ;{id=1337 (zsk)}
keytrap.org 60 IN DNSKEY 256 3 13 ijhRFa4f[...]L5cBtQ== ;{id=1337 (zsk)}

keytrap.org 60 IN RRSIG DNSKEY 13 4 60 20250117 20240119 26539 keytrap.org gTq+83Bx[...]hcLjQ==
```

Zone File Contents: RRSIGs

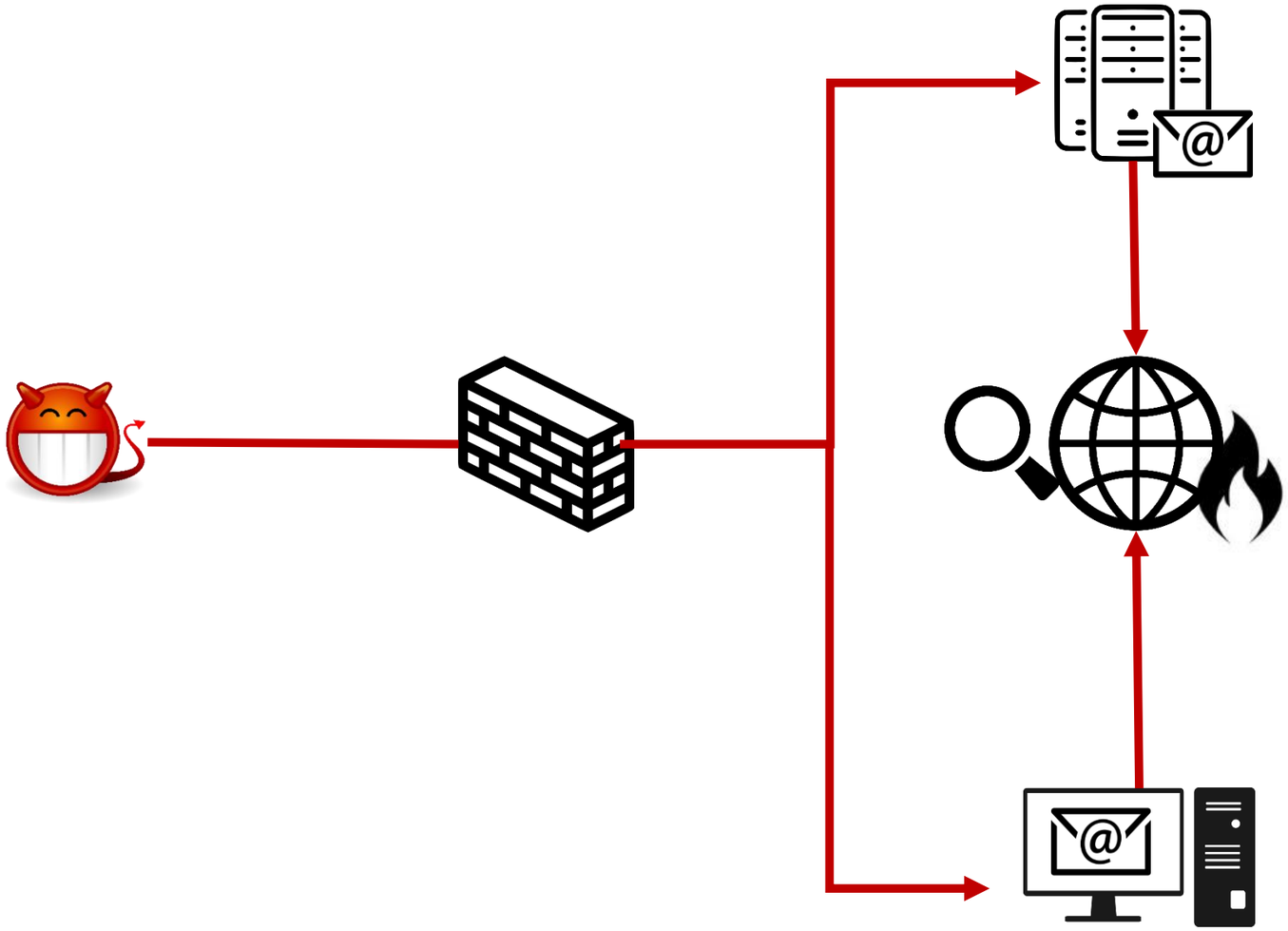
```
a0001.keytrap.org 0 IN A 10.2.3.4

a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
…
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
a0001.keytrap.org 0 IN RRSIG A 13 5 0 20250117 20240119 1337 keytrap.org GVHrz5G0+[...]0VObdw==
```

Attracting a Victim Resolver

Attack Vectors

- Querying directly

Attracting a Victim Resolver



Attack Vectors

- Querying directly
- SMTP Bounce
- HTML E-mail

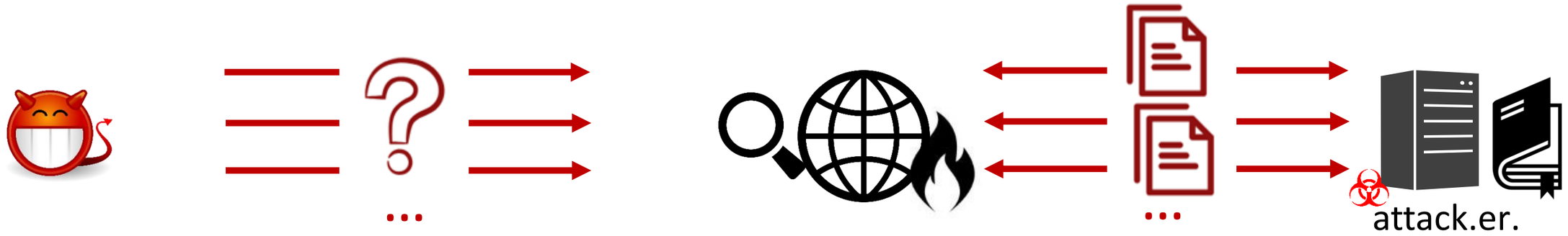
Scaling up the victim count

How to attract many resolvers at the same time?

- Online Ads

- Internet Measurement Networks

Attack Procedure



- Flooding the victim resolver with malicious queries

Demo Time

(Demo)

KeyTrap Attacks

SigJam
Jamming the validator with signatures only

LockCram
Collisions-only attack, using a single RRSIG per response

KeySigTrap
Combined attack

HashTrap
KeySigTrap with DS records

Impact and Evaluations

Which algorithm to use?

Cipher	Keys	Signatures	Validations
ED448	907	454	411 778
ED25519	1 391	696	968 136
ECDSAP384SHA384	589	519	305 691
ECDSAP256SHA256	828	696	576 288
RSA-512	788	696	548 448
RSA-1024	444	413	183 372
RSA-2048	237	228	54 036
RSA-4096	122	119	14 518

Max validations

Min validations

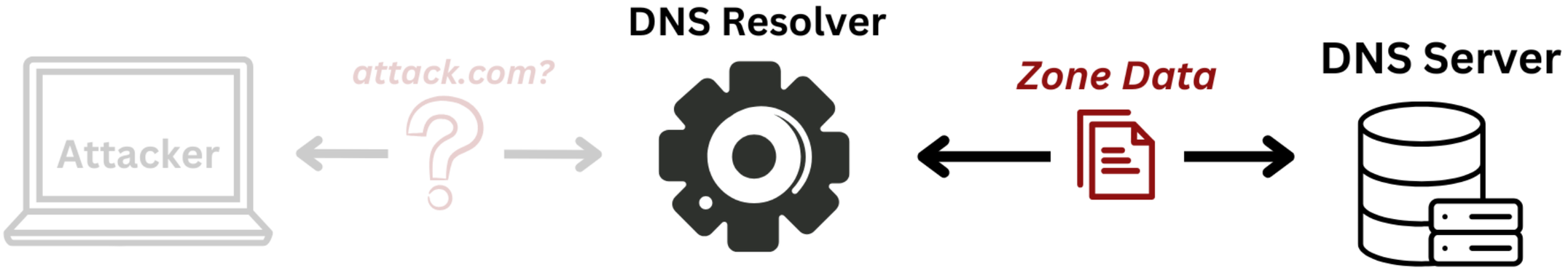
Up to ~1M signature validations per DNS Request

Evaluations

Test Setup with Unbound.

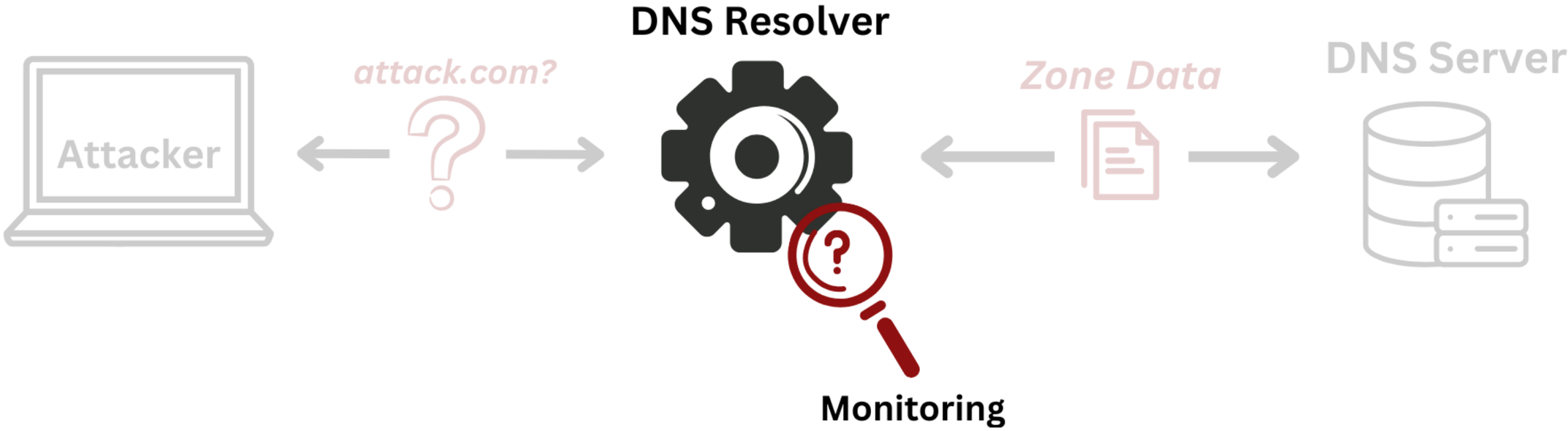
Evaluations

Test Setup with Unbound.



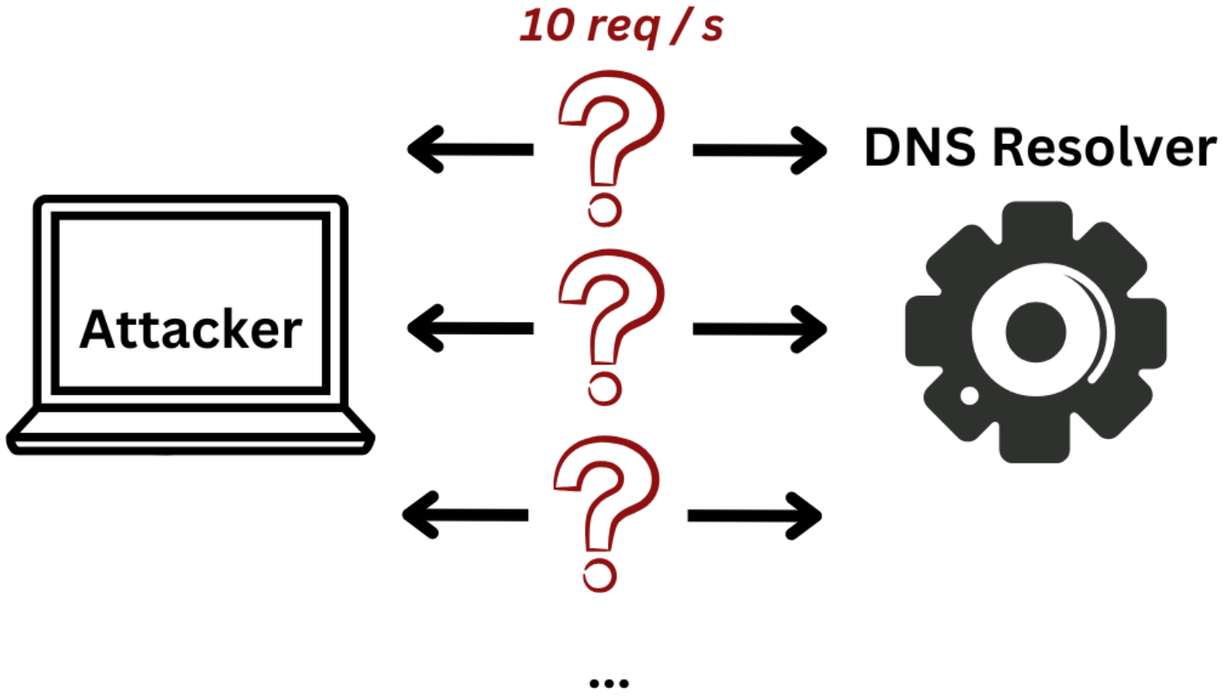
Evaluations

Test Setup with Unbound.



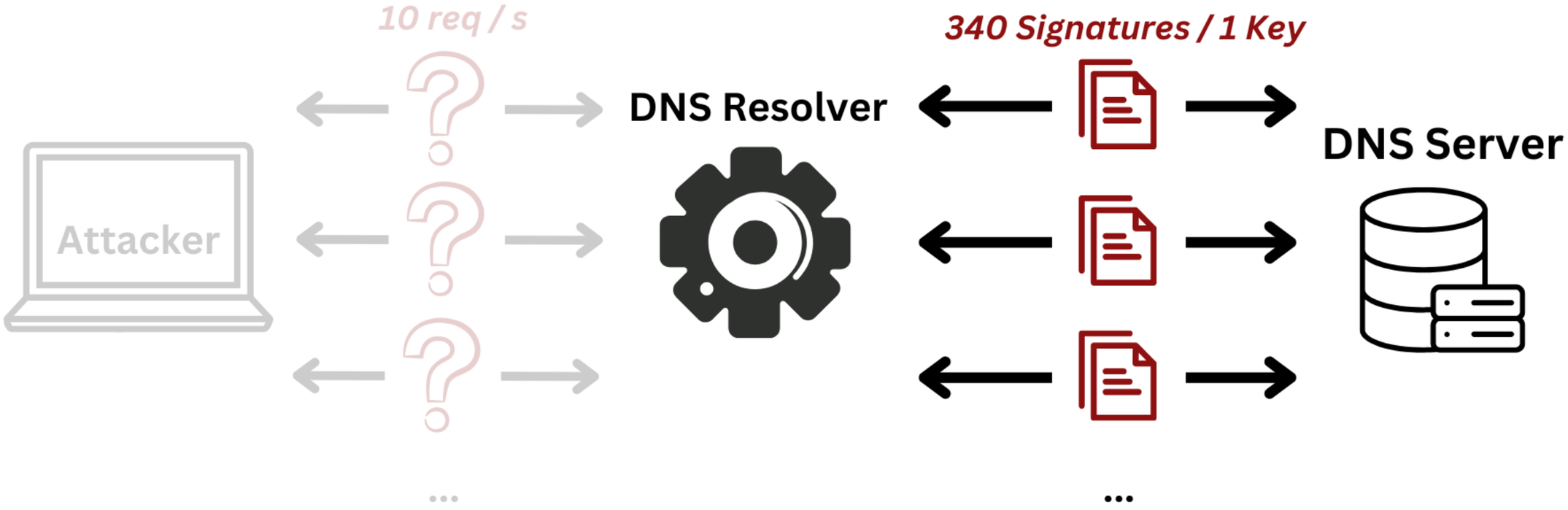
Evaluations

Test 1: SigJam (Many Signatures / 1 Key)



Evaluations

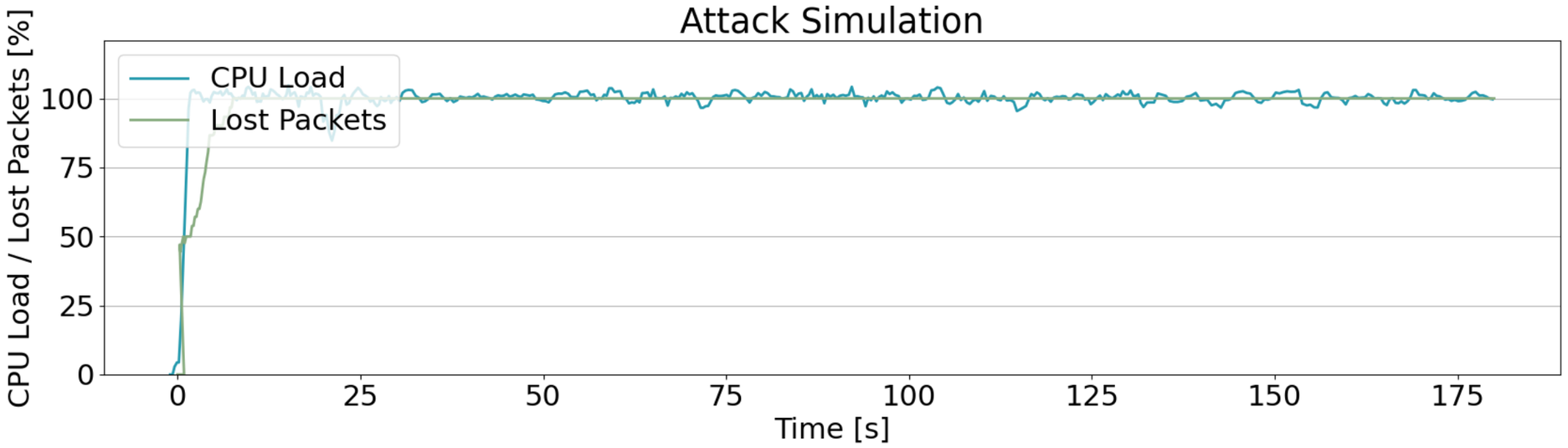
Test 1: SigJam (Many Signatures / 1 Key)



Evaluations

340 Signature Validations per Request

10 req/s

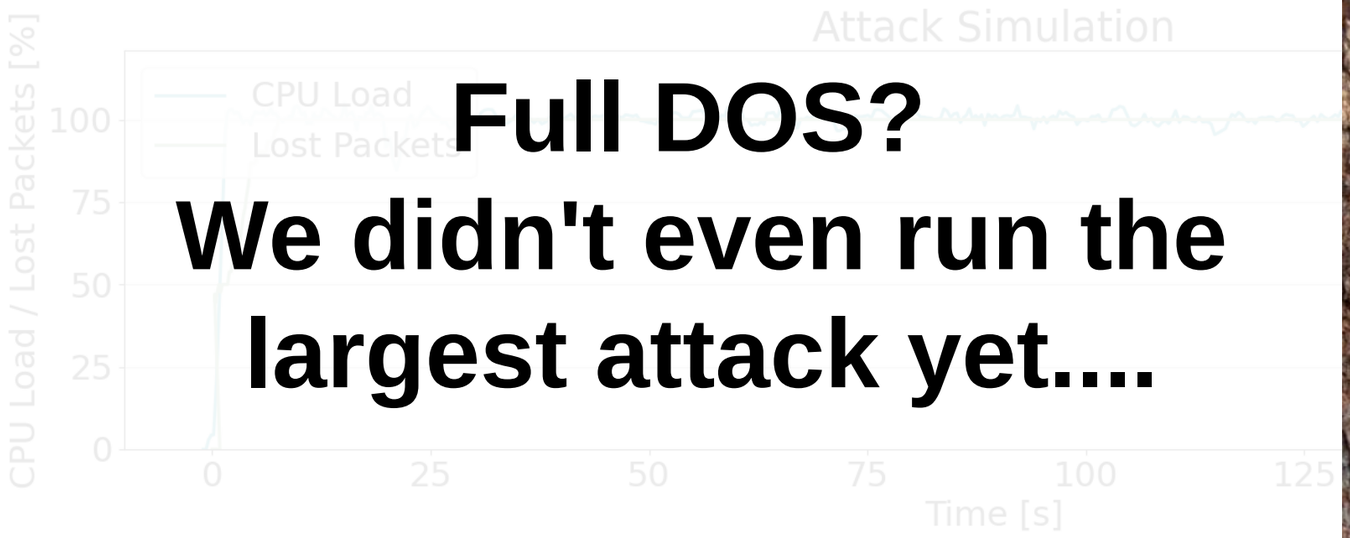


Measured on Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz with Unbound

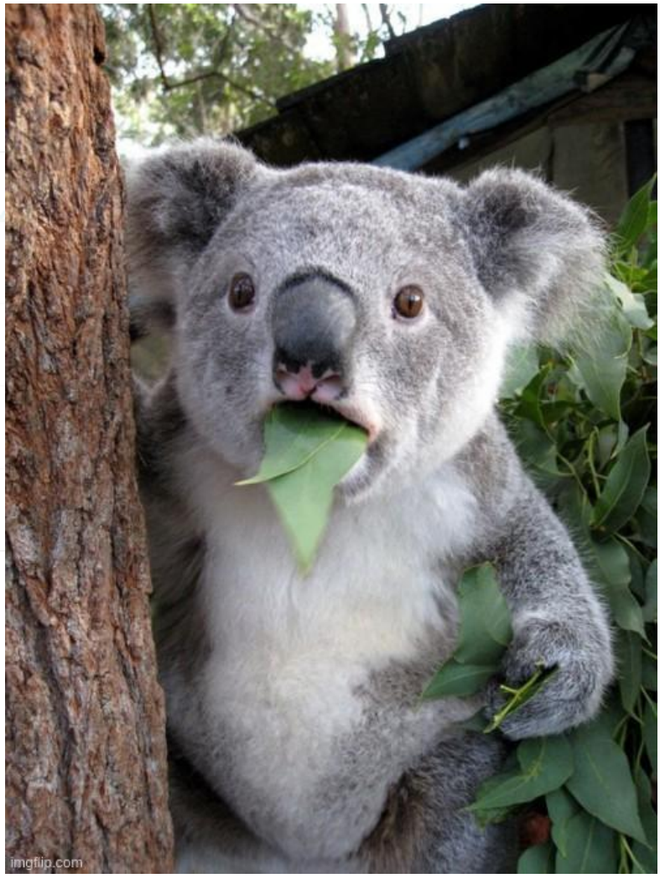
Evaluations

340 Signature Validations per Request

10 req/s



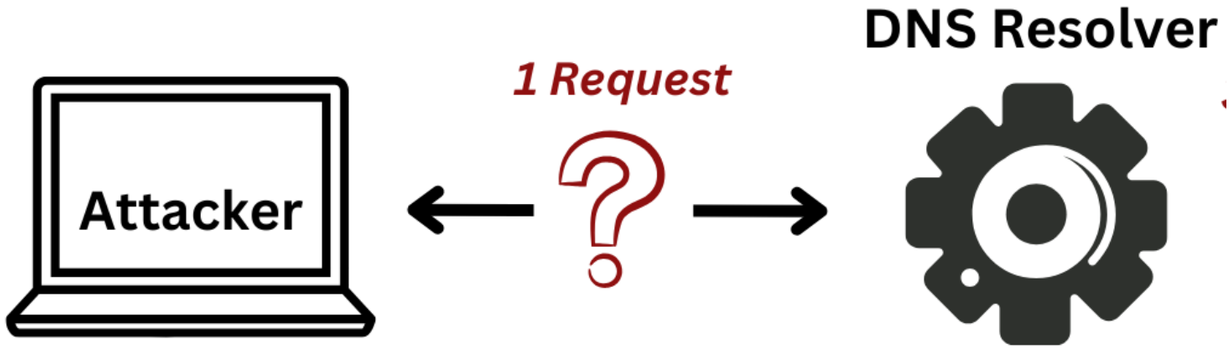
Full DOS?
We didn't even run the largest attack yet....



Measured on Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz

Evaluations

Test 2: KeySigTrap (Many Signatures / Many Keys)



Evaluations

Test 2: KeySigTrap (Many Signatures / Many Keys)

Evaluations

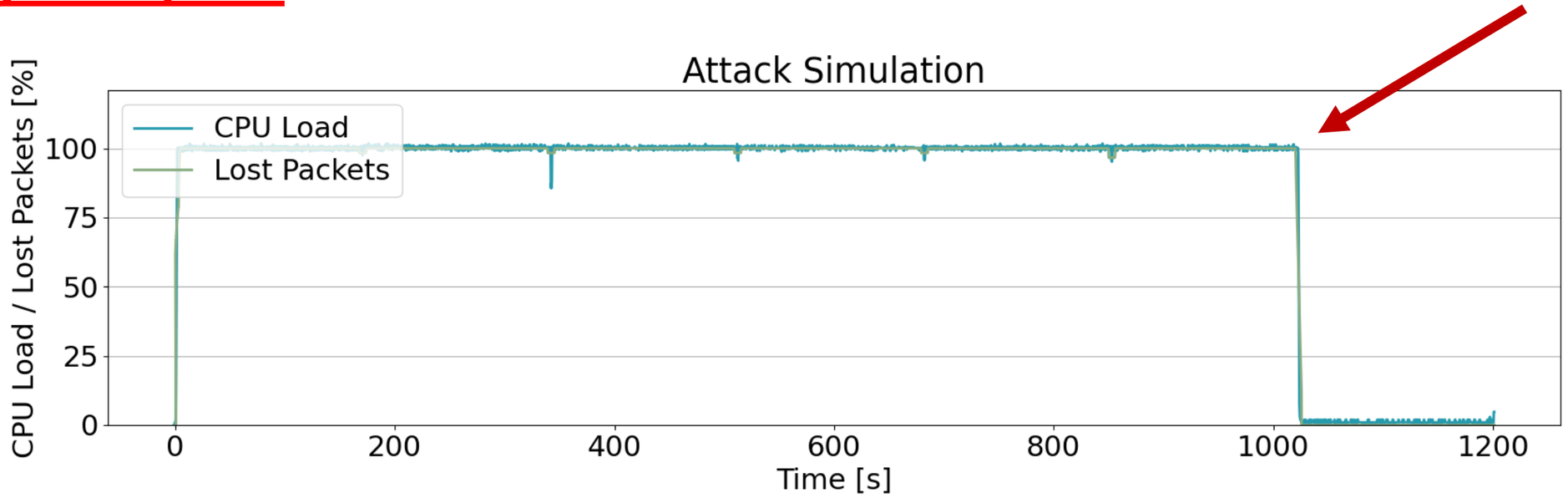
340 Signatures x 582 Keys = 221160 Validations

Single Request

Evaluations

340 Signatures x 582 Keys = 221160 Validations

Single Request

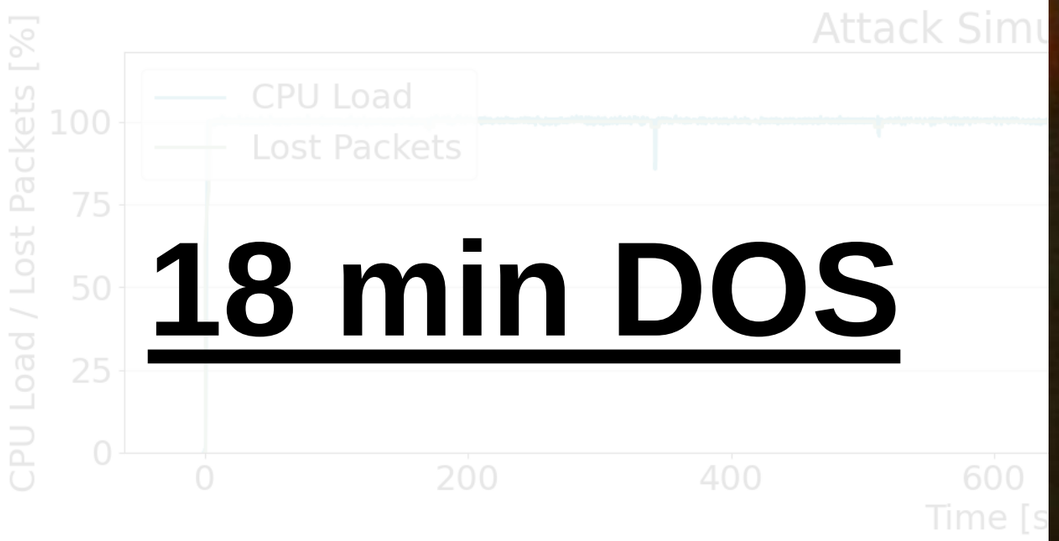


Measured on Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz

Evaluations

340 Signatures x 582 Keys = 221160 Validations

Single Request



18 min DOS

Measured on Intel(R) Xeon(R) Gold 6242 CPU @ 2.80GHz

Evaluations

Resolver	Stall Duration (Single Request)
Unbound	1014s
Bind9	58632s
Knot	51s
Akamai	186s
PowerDNS	170s
Windows Server	132s
Stubby	184s
Cloudflare 1.1.1.1	Vulnerable
Amazon Route 53	Vulnerable
Google DNS	Vulnerable

All tested DNSSEC implementing resolvers vulnerable!

Evaluations

Resolver	Stall Duration (Single Request)
Unbound	1014s
Bind9	58632s
Knot	51s
Akamai	186s
PowerDNS	170s
Windows Server	132s
Stubby	184s
Cloudflare 1.1.1.1	Vulnerable
Amazon Route 53	Vulnerable
Google DNS	Vulnerable

All tested DNSSEC implementing resolvers vulnerable!

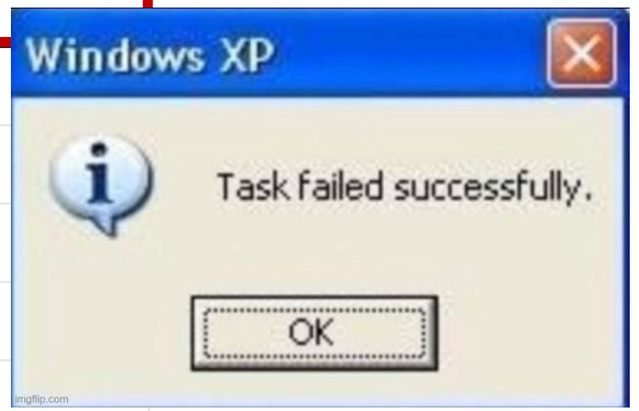
Evaluations

Resolver	Stall Duration (Single Request)
Unbound	1014s
Bind9	58632s
Knot	51s
Akamai	186s
PowerDNS	170s
Windows Server	132s
Stubby	184s
Cloudflare 1.1.1.1	Vulnerable
Amazon Route 53	Vulnerable
Google DNS	Vulnerable

All tested DNSSEC implementing resolvers vulnerable!

Evaluations

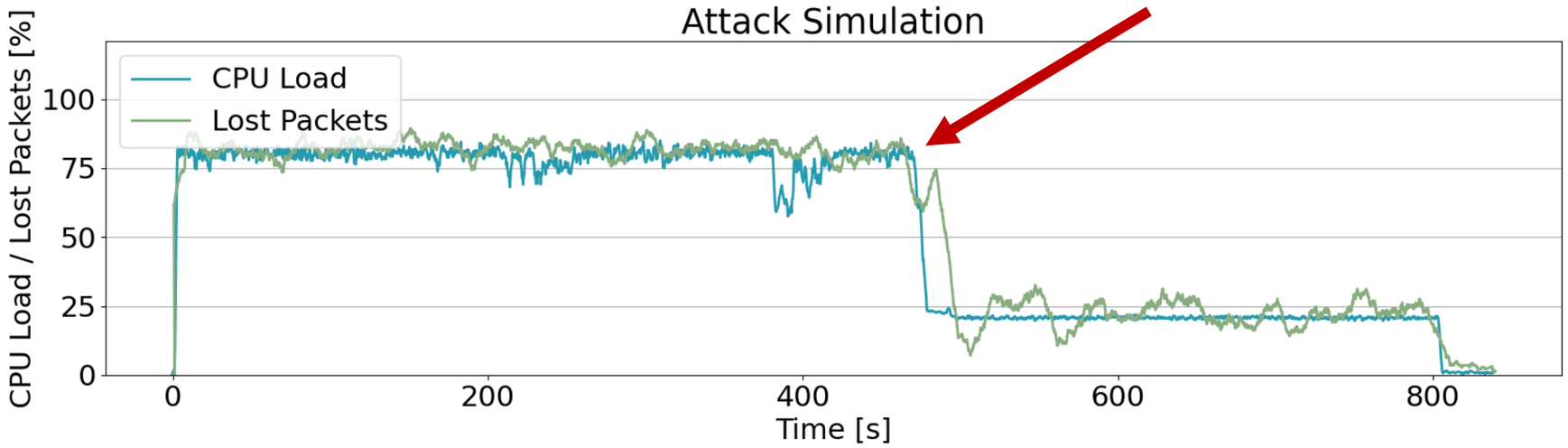
Resolver	Stall Duration (Single Request)
Unbound	1014s
Bind9	58632s
Knot	51s
Akamai	186s
PowerDNS	170s
Windows Server	132s
Stubby	184s
Cloudflare 1.1.1.1	Vulnerable
Amazon Route 53	Vulnerable
Google DNS	Vulnerable



All tested DNSSEC implementing resolvers vulnerable!

Evaluations

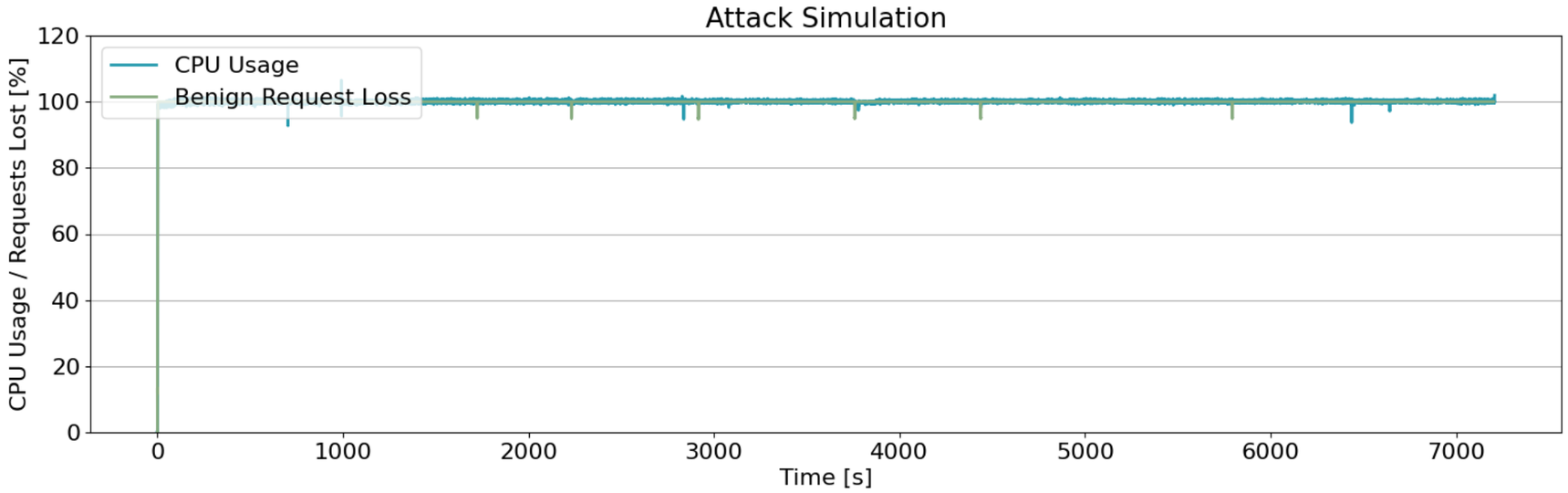
Does Multi-Threading help?



No, but attacker needs to send a few more requests....

Evaluations

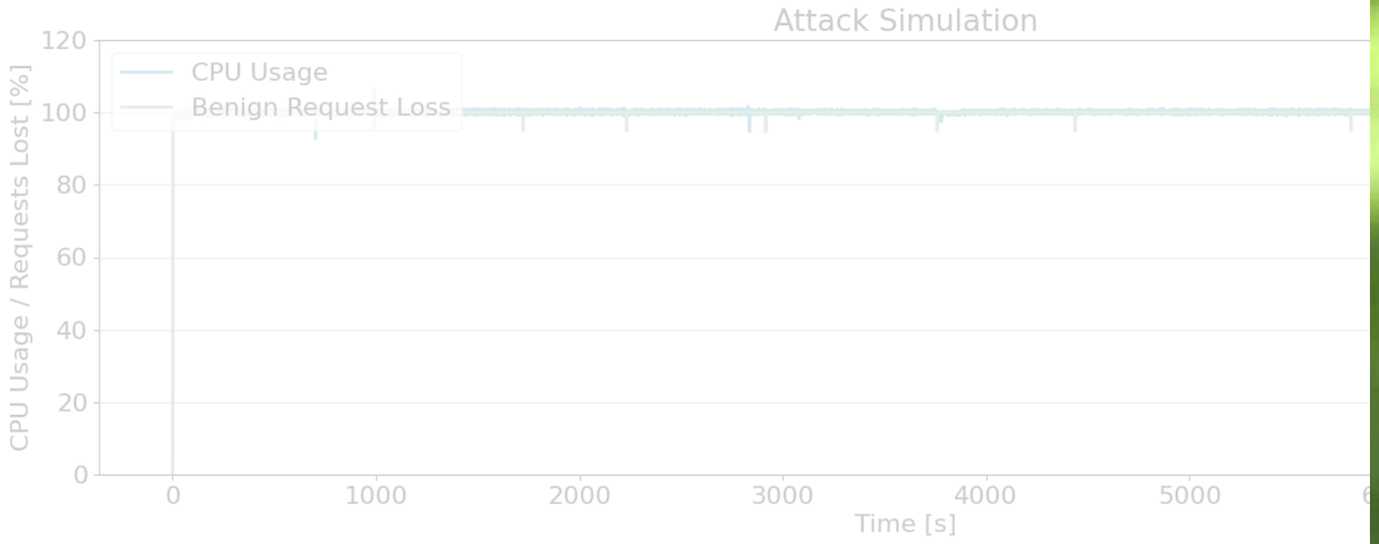
Viable attack: Comprehensive persistent DoS

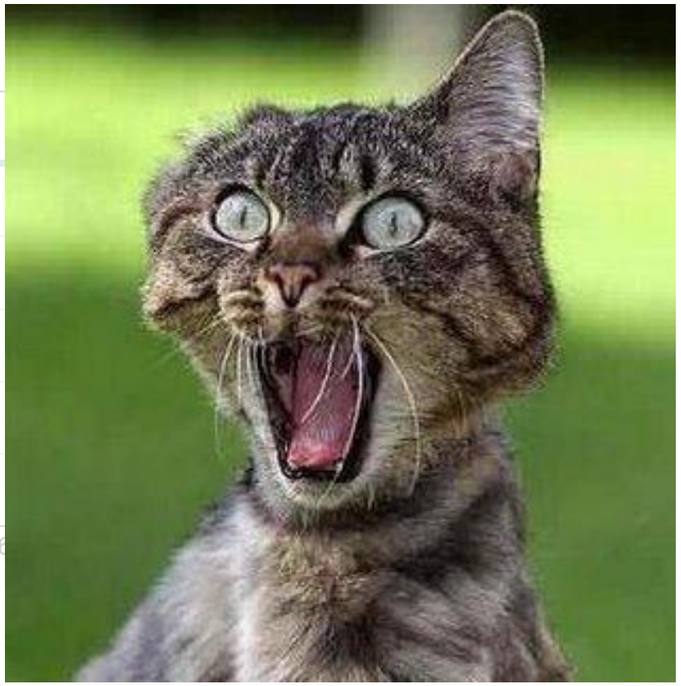


KeyTrap allows to disable DNSSEC resolver with max. a few packets / min

Evaluations

Viable attack: Comprehensive persistent DoS

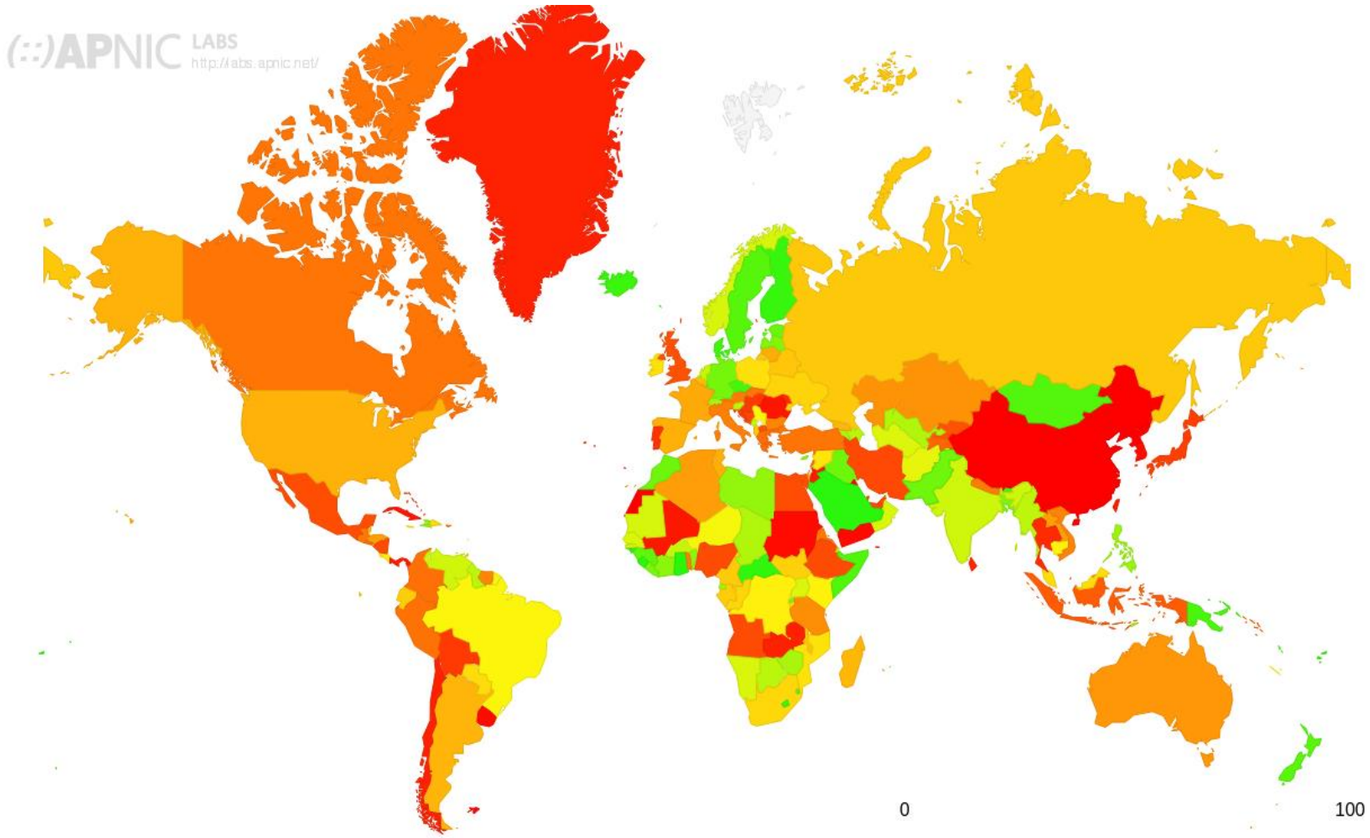




KeyTrap allows to disable DNSSEC resolver with max. a few packets / min

Impact

Global Impact

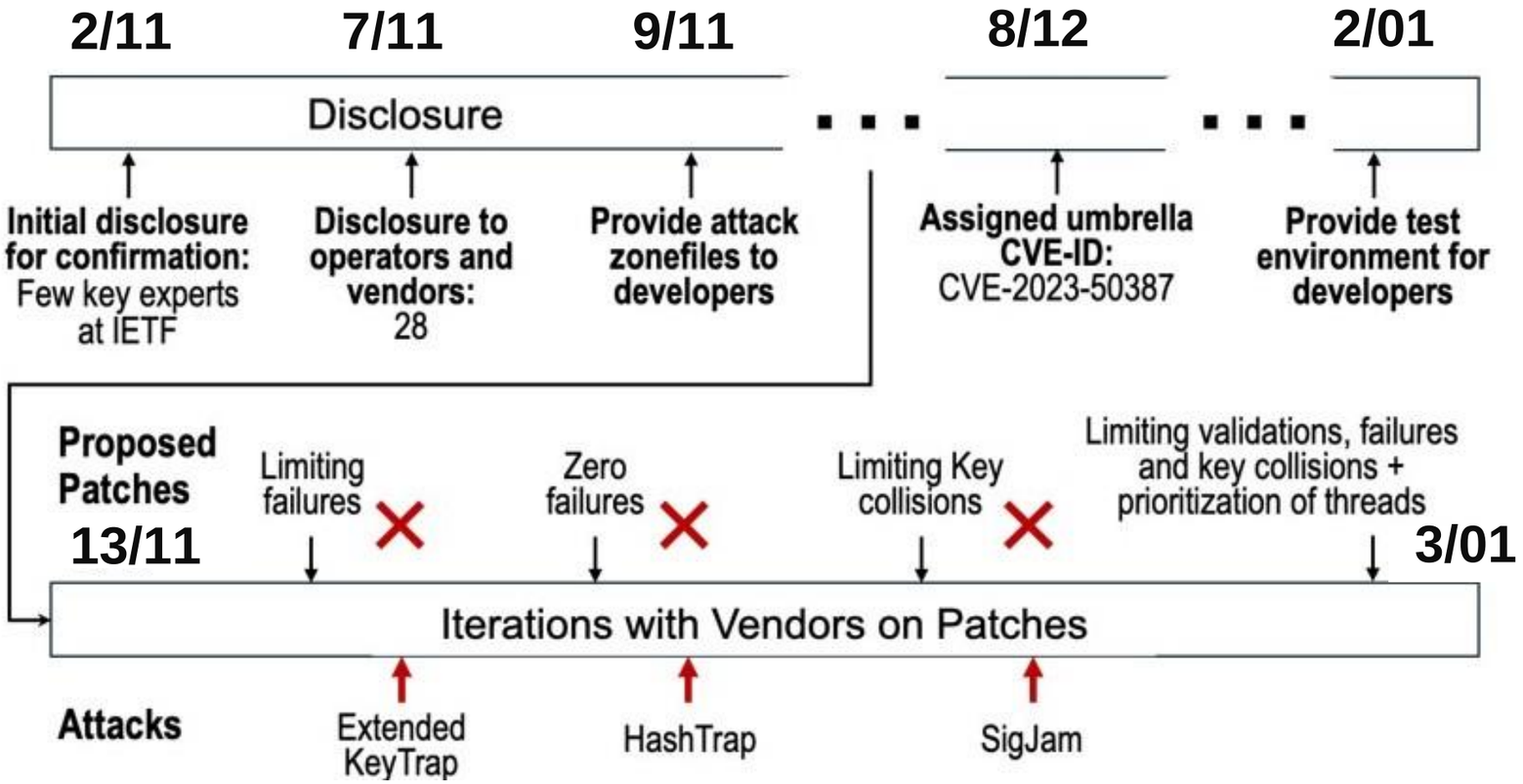


Code	Region	DNSSEC Validates	Partial Validates	Total Validates
XA	World	33.67%	8.02%	41.69%
XF	Oceania	44.95%	3.80%	48.75%
XE	Europe	42.92%	12.77%	55.69%
XB	Africa	37.67%	12.11%	49.78%
XC	Americas	34.82%	6.61%	41.43%
XD	Asia	30.07%	6.57%	36.64%
XG	Unclassified	28.84%	16.29%	45.13%

Measurements by APNIC Labs

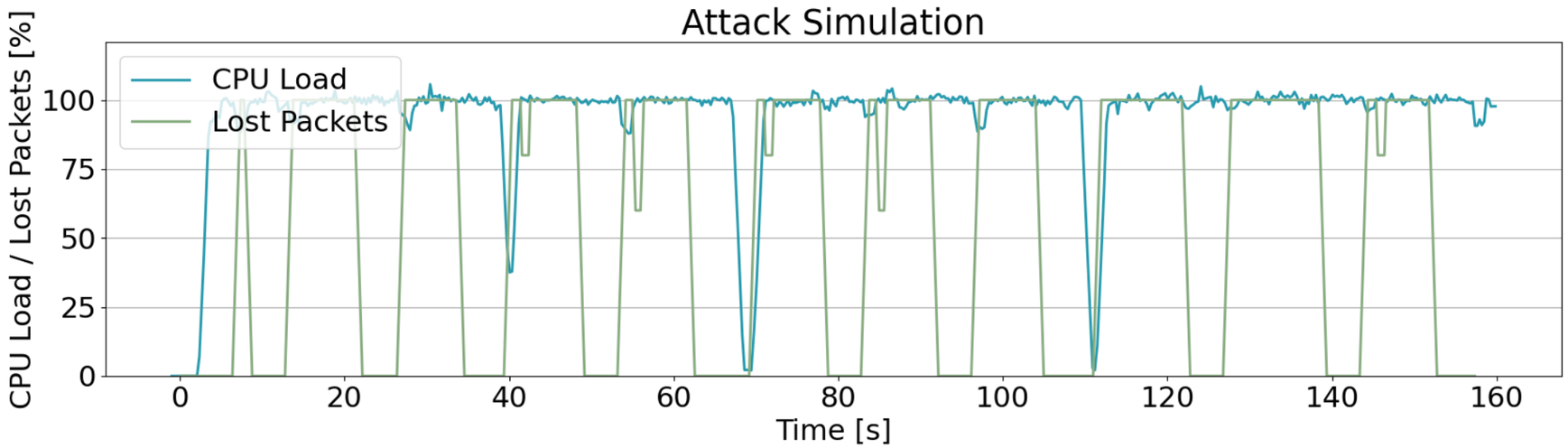
Disclosure: Patch-Break-Fix Process

Disclosure



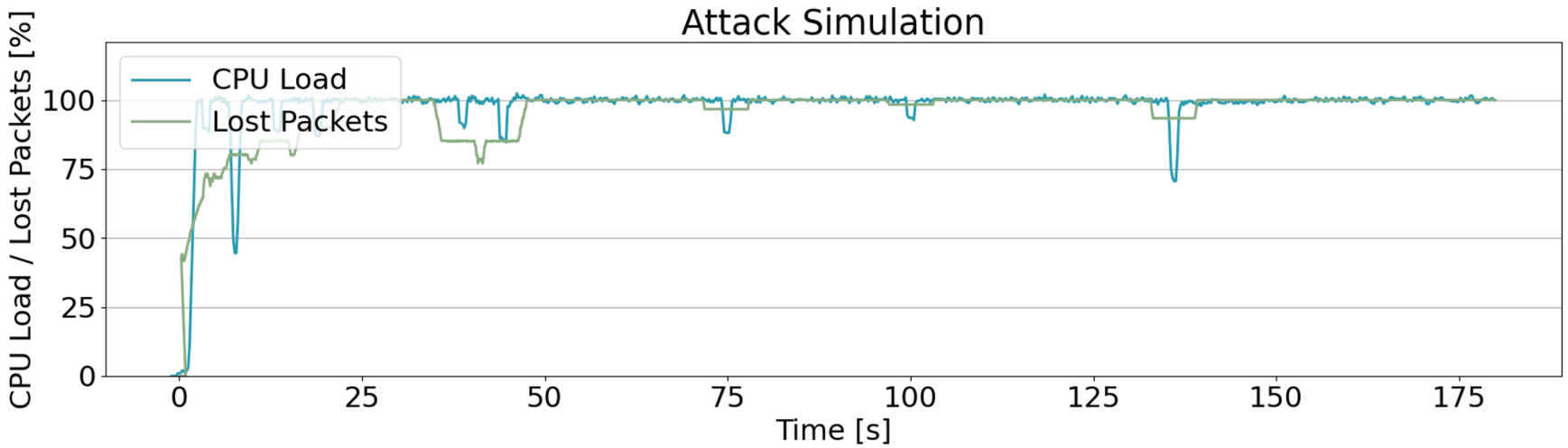
Fixing KeyTrap included > 30 people and took 3 months

(Patched) Resolver under continuous attack



Patches

(Patched) Resolver under Hash attack



Lessons Learned

BlackHat Soundbites

- The KeyTrap Attacks allow for comprehensive DoS on a plethora of DNSSEC-validating resolvers

- Vulnerability stems from "eager validation" in RFCs
  Desire to ensure availability compromised availability

- Specification needs to consider impact on resource consumption by implementations

- Read our paper here -
https://doi.org/10.1145/3658644.3670389