Recent DNS DDoS Attacks:

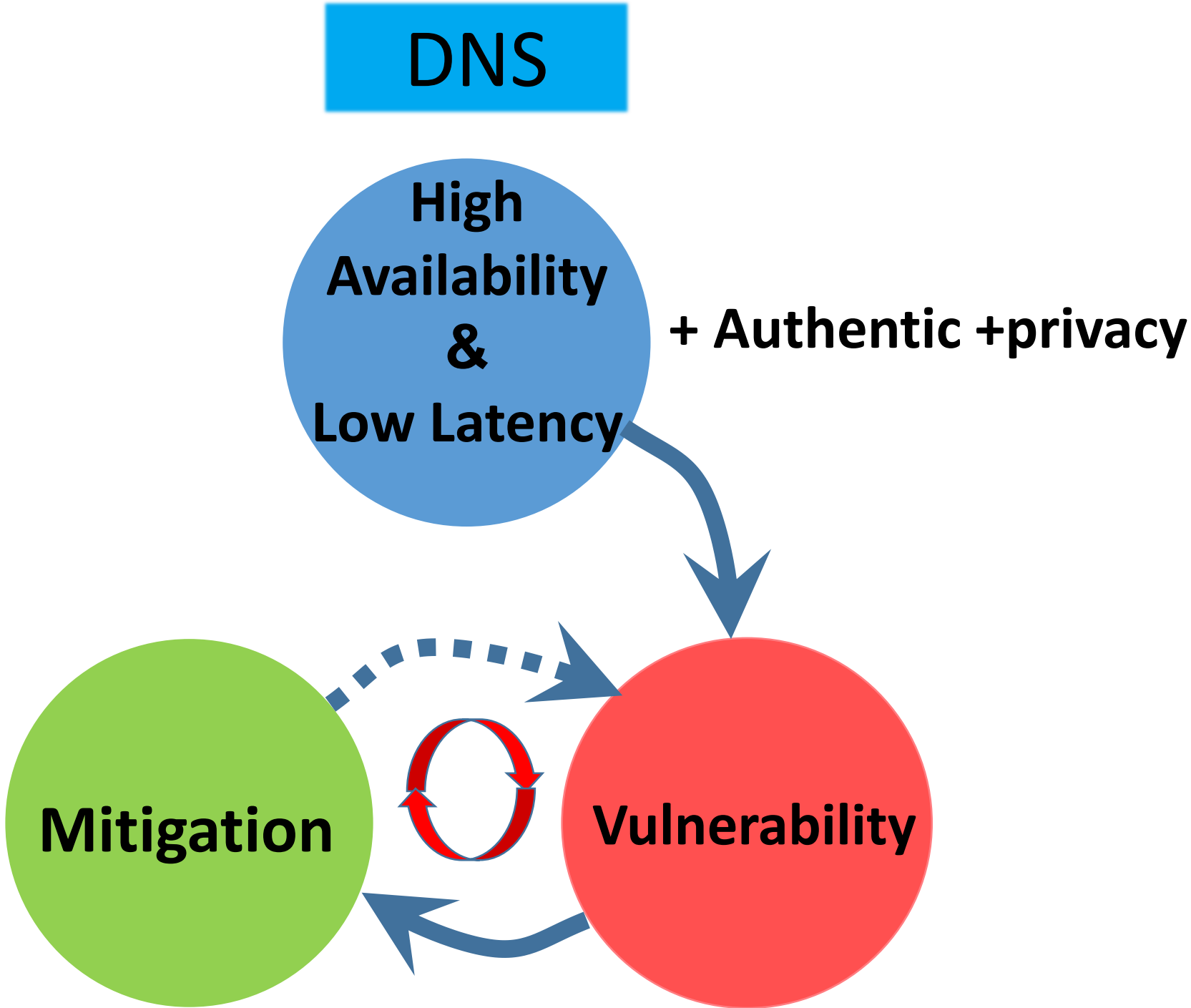
Academic Perspective

DNS OARC 43 -- 10/26/24

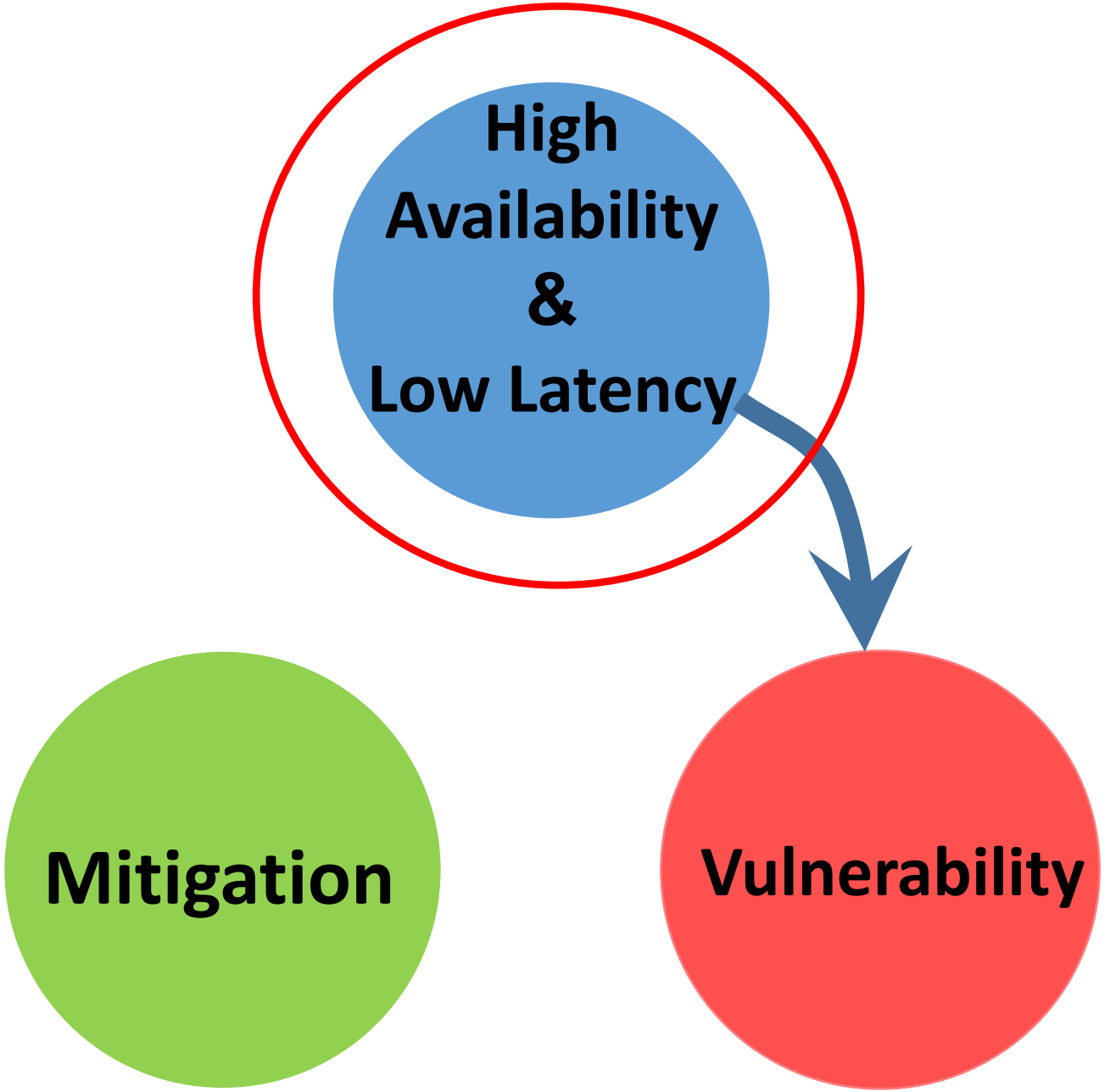
Yehuda Afek and Anat Bremler-Barr

Tel-Aviv University

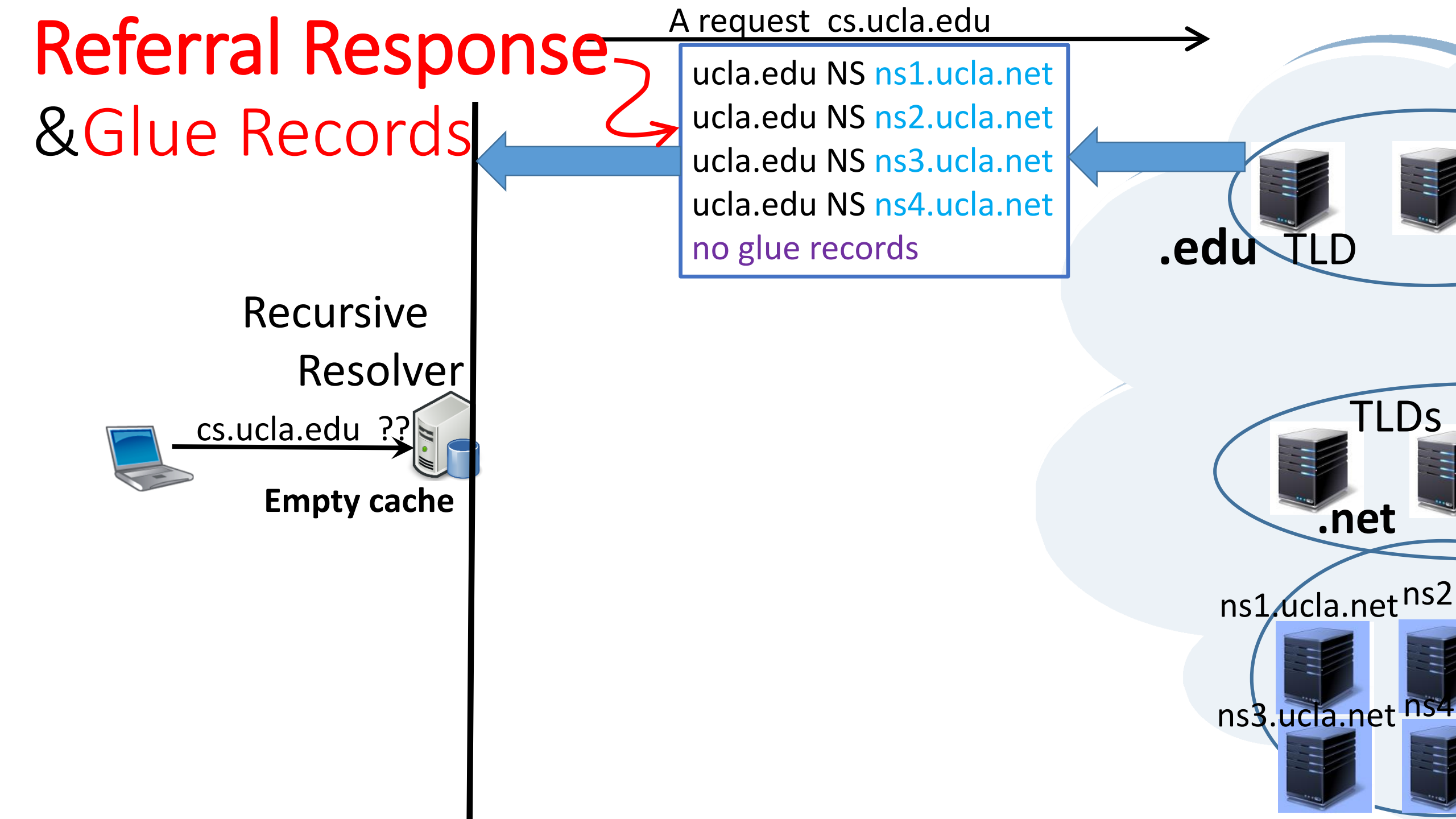
Outline


DNS
High Availability & Low Latency
+ Authentic +privacy
Mitigation
Vulnerability

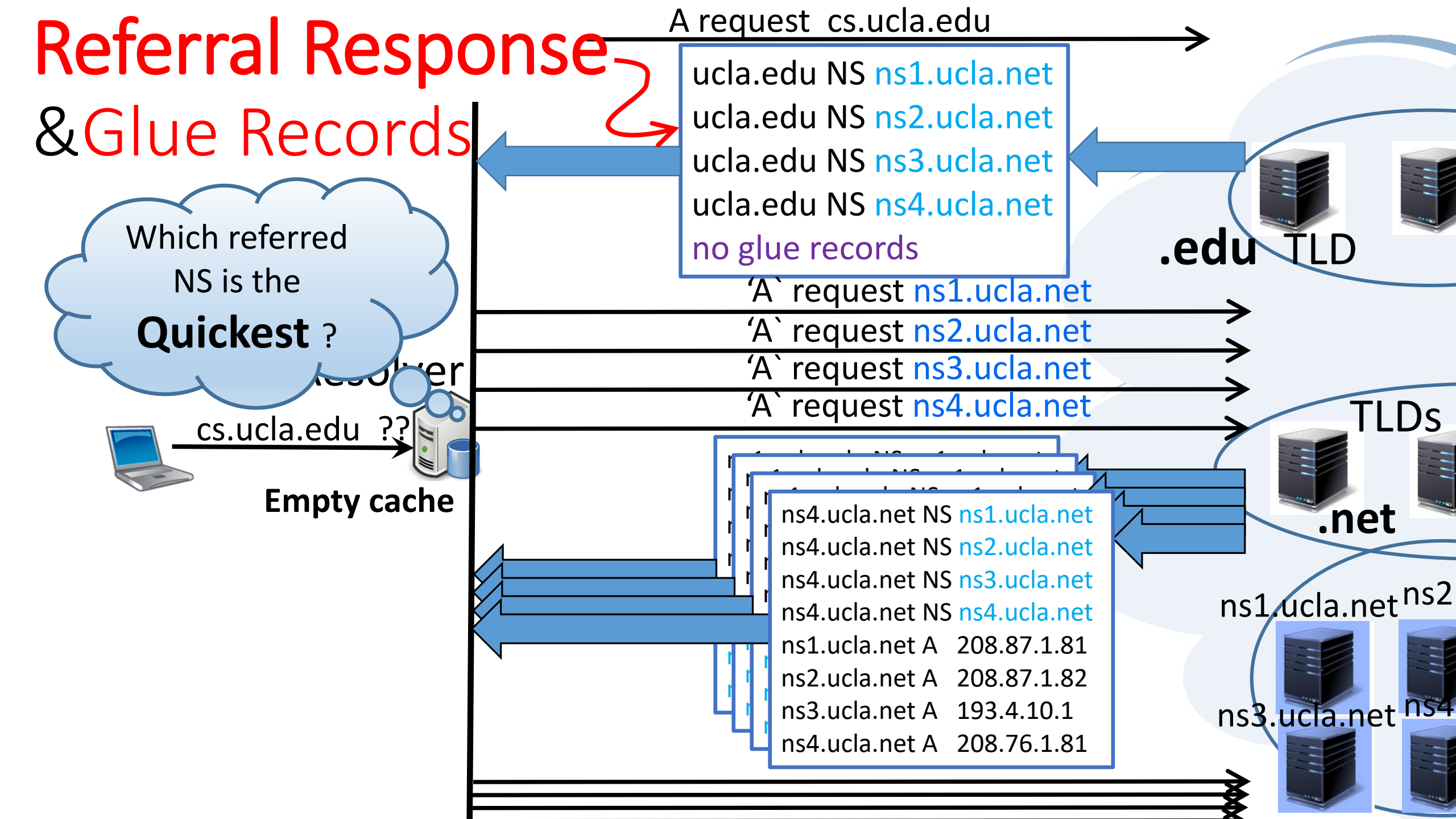
cycle 1 example



Referral Response &Glue Records



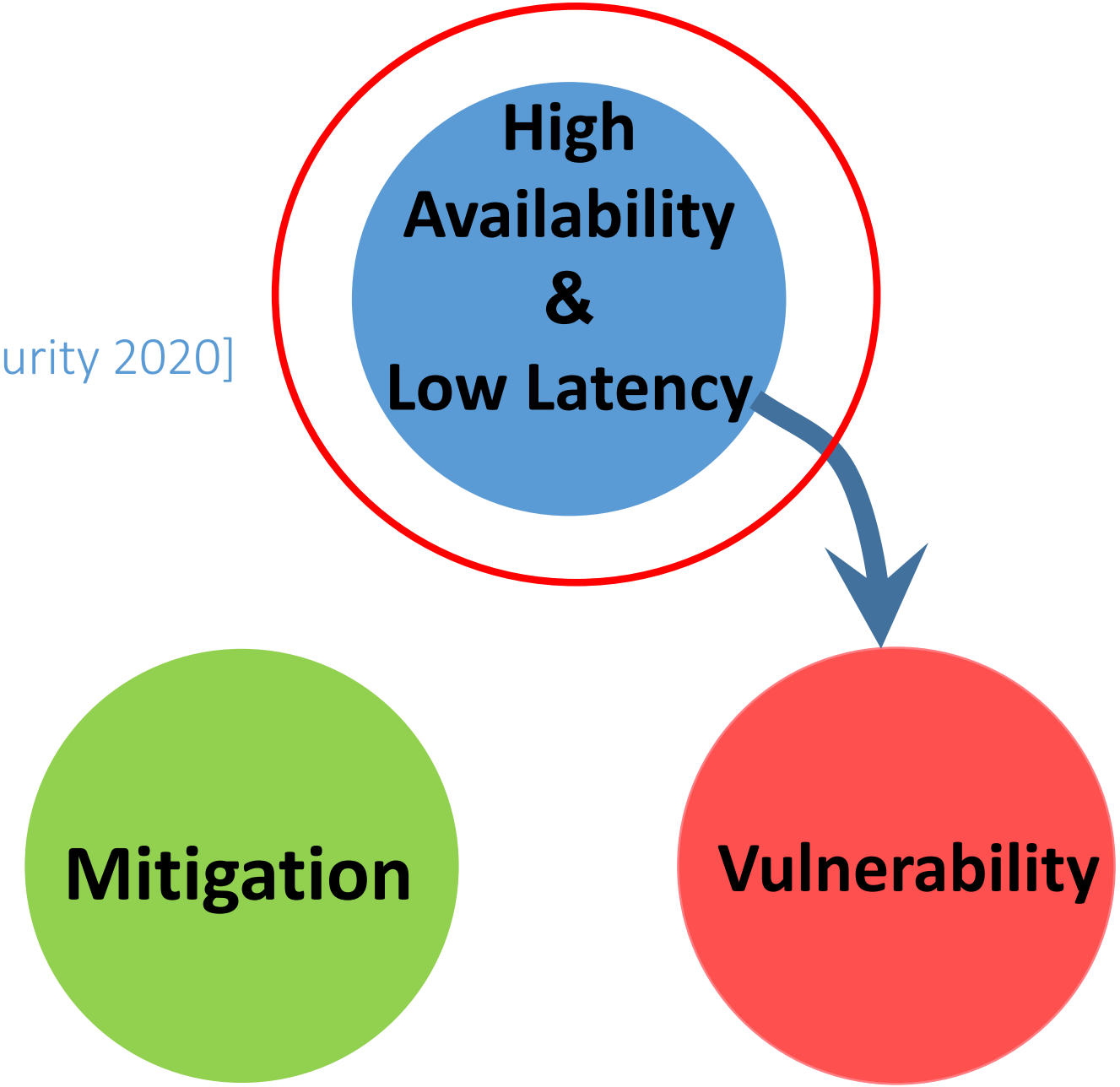
Referral Response & Glue Records



First cycle example

NXNS Attack

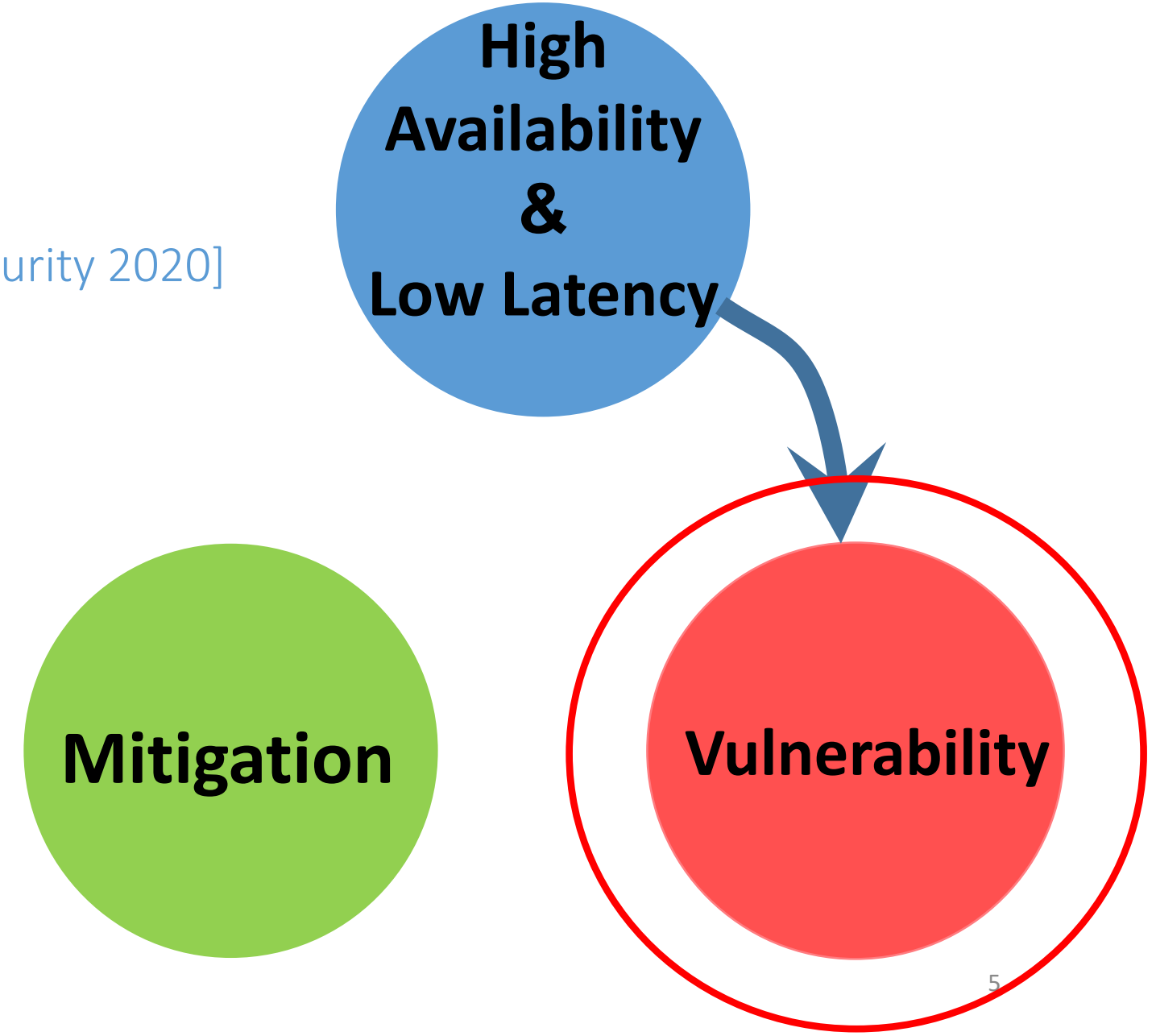
[A, Anat Bremler-Barr, Lior Shafir. Usenix Security 2020]



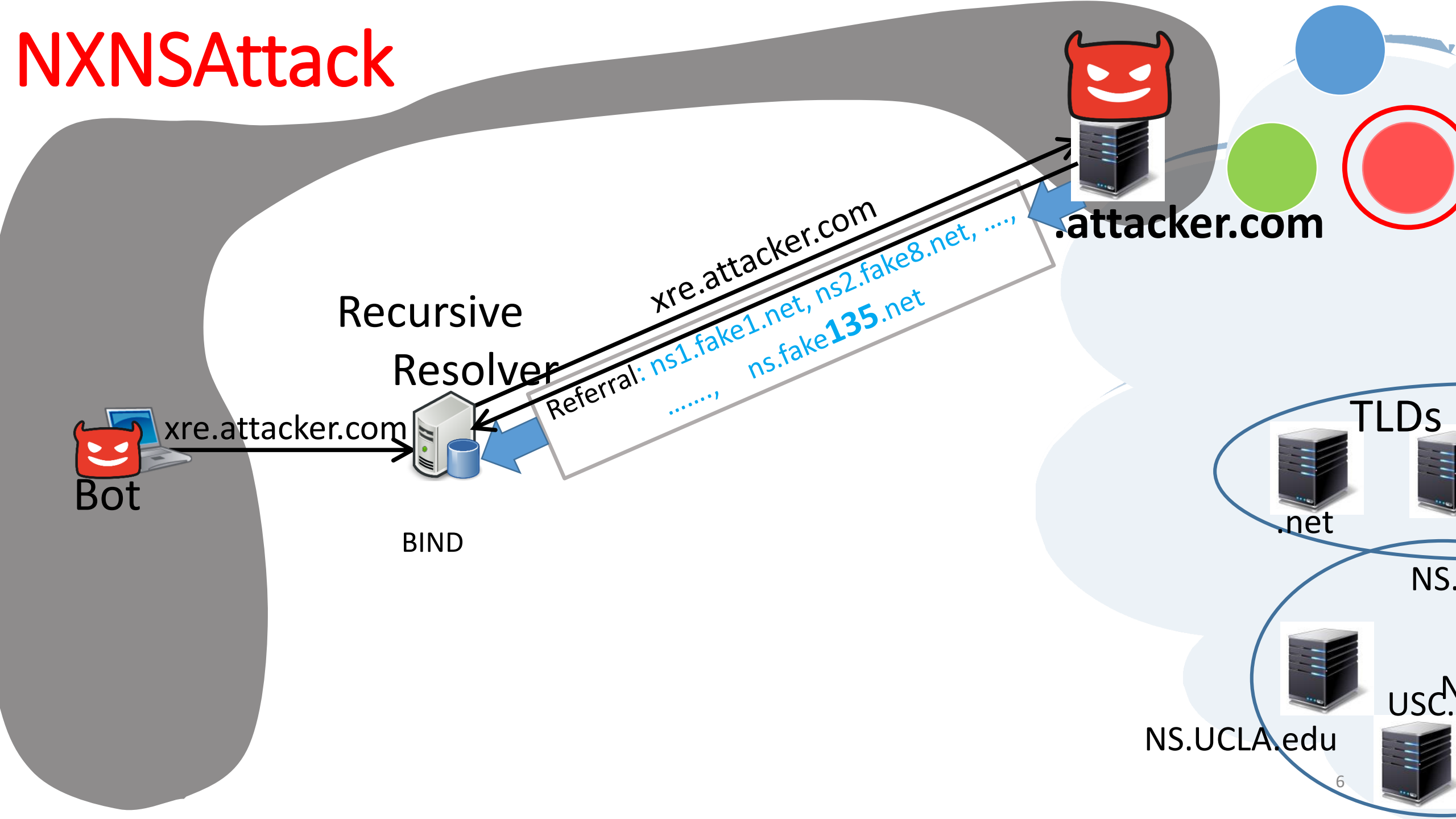
First cycle example

NXNS Attack

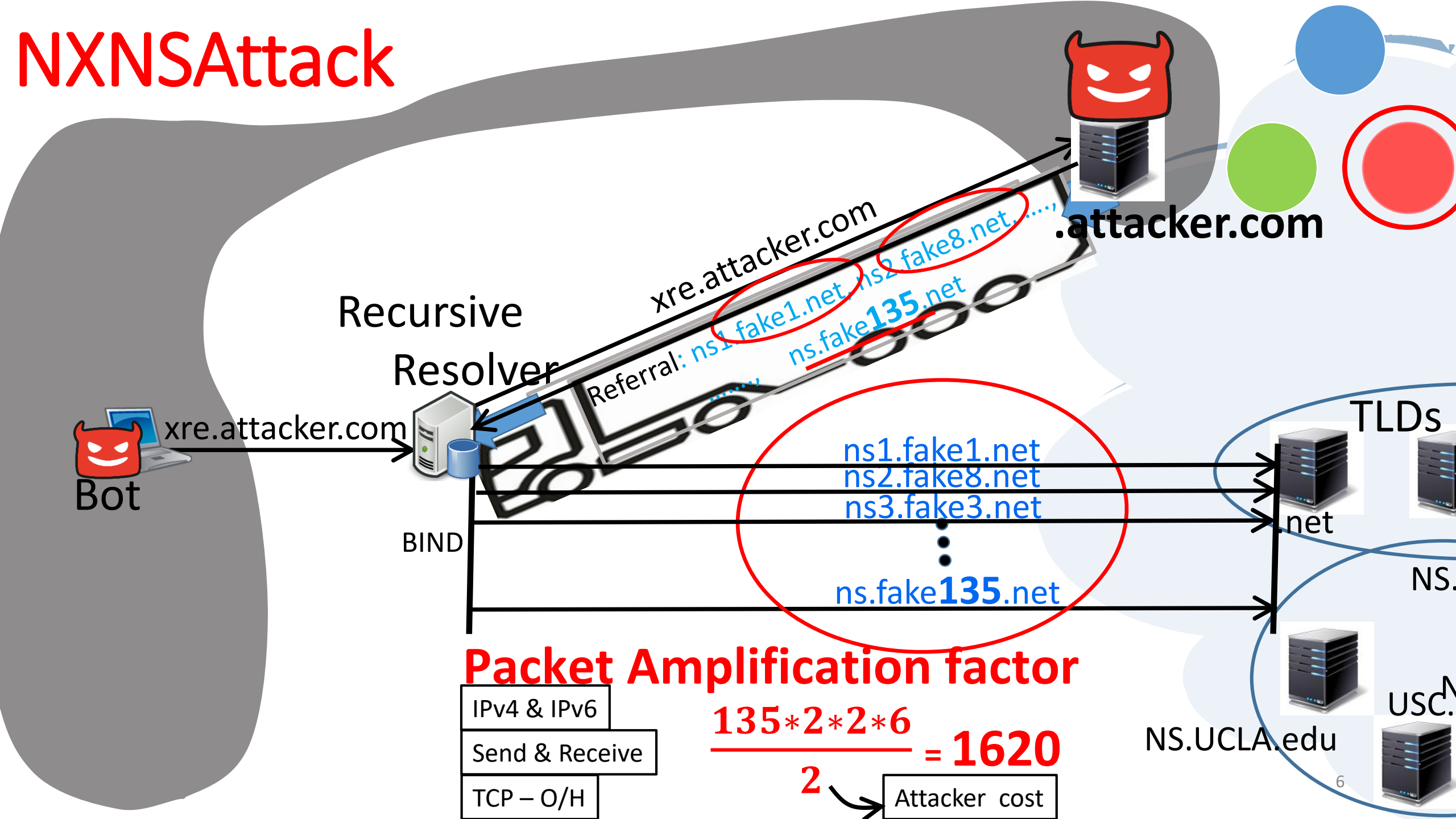
[A, Anat Bremler-Barr, Lior Shafir. Usenix Security 2020]



NXNSAttack


.attacker.com
Recursive
Resolver
xre.attacker.com
Referral: ns1.fake1.net, ns2.fake8.net, ….,
……, ns.fake135.net
xre.attacker.com
Bot
BIND
TLDs
.net
NS.
NS.UCLA.edu
USC.

NXNSAttack



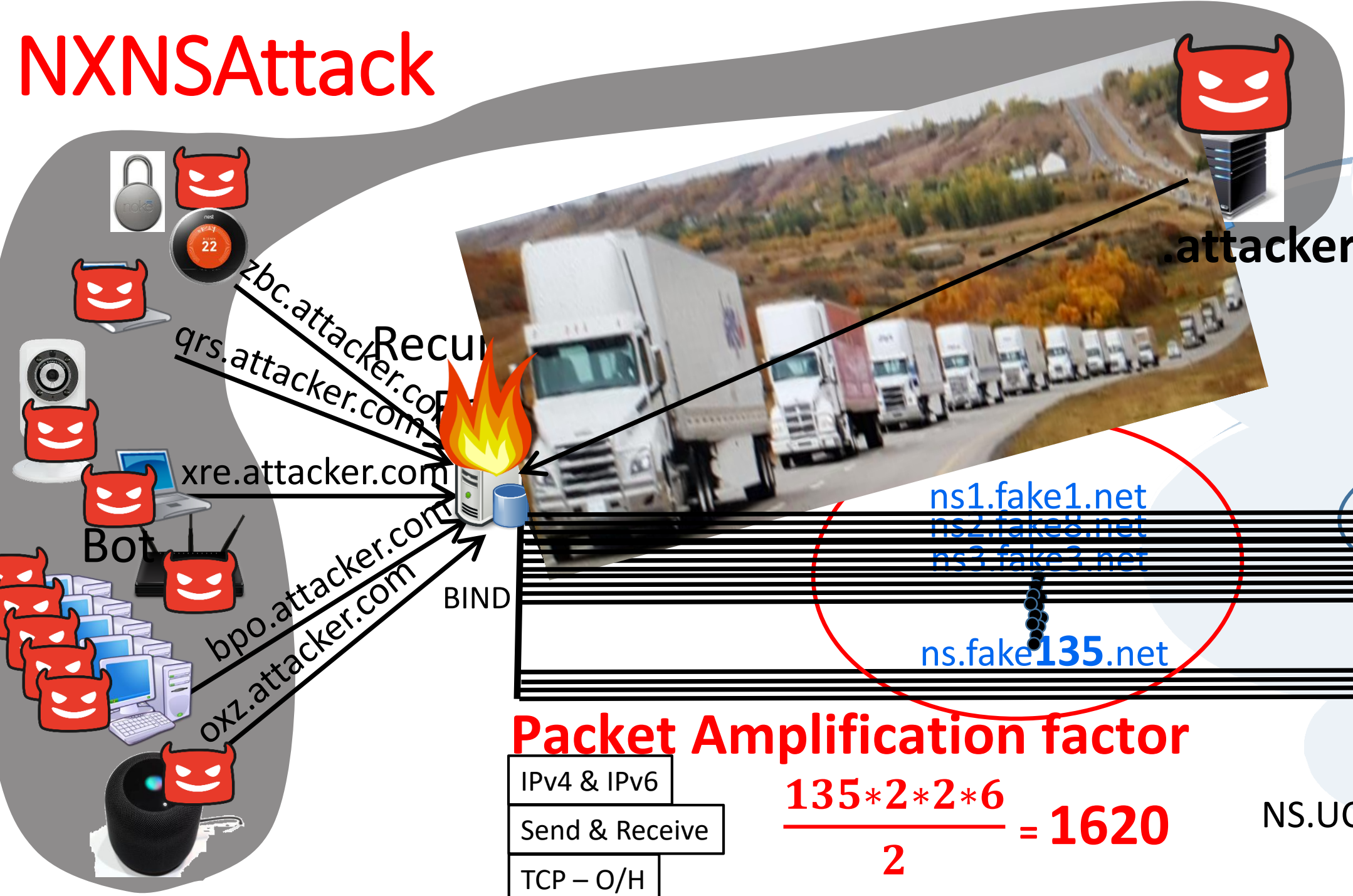
Packet Amplification factor

- IPv4 & IPv6
- Send & Receive
- TCP – O/H

$$\frac{135*2*2*6}{2} = 1620$$

Attacker cost

NXNSAttack



Packet Amplification factor

- IPv4 & IPv6
- Send & Receive
- TCP – O/H

$$\frac{135*2*2*6}{2} = 1620$$

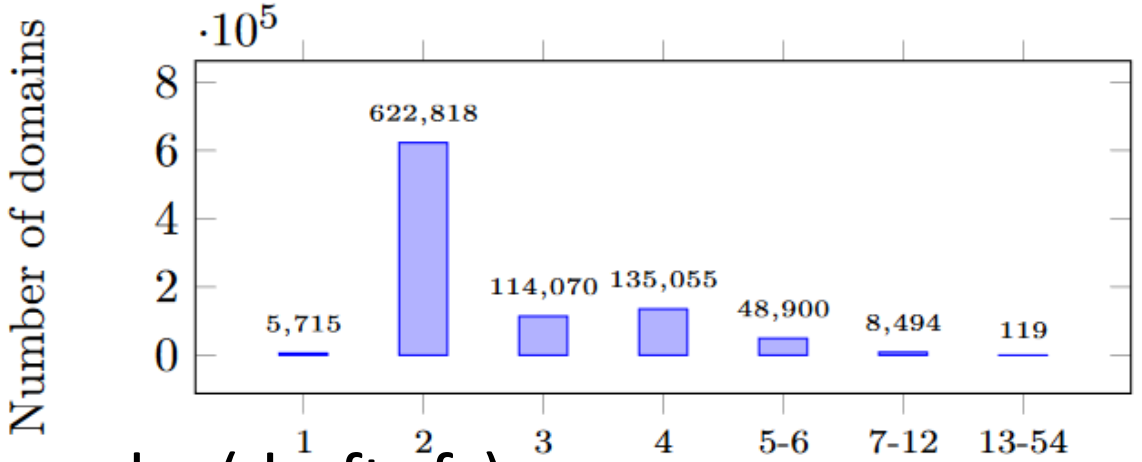
Mitigation: NXNS

- MaxFetch(k) – Resolve k at a time NS-names, not all at once
  - Amortized on several queries

- MaxBreadth – bound # of NS-names per referral response

- Detect NX NS replies (NLnetLabs)

- DNSSEC – NSEC (Petr Špaček)



- Going only downwards in the DNS hierarchy (draft rfc)

Mitigation: NXNS

- MaxFetch(k) – Resolve k at a time NS-names, not all at once
  - Amortized on several queries

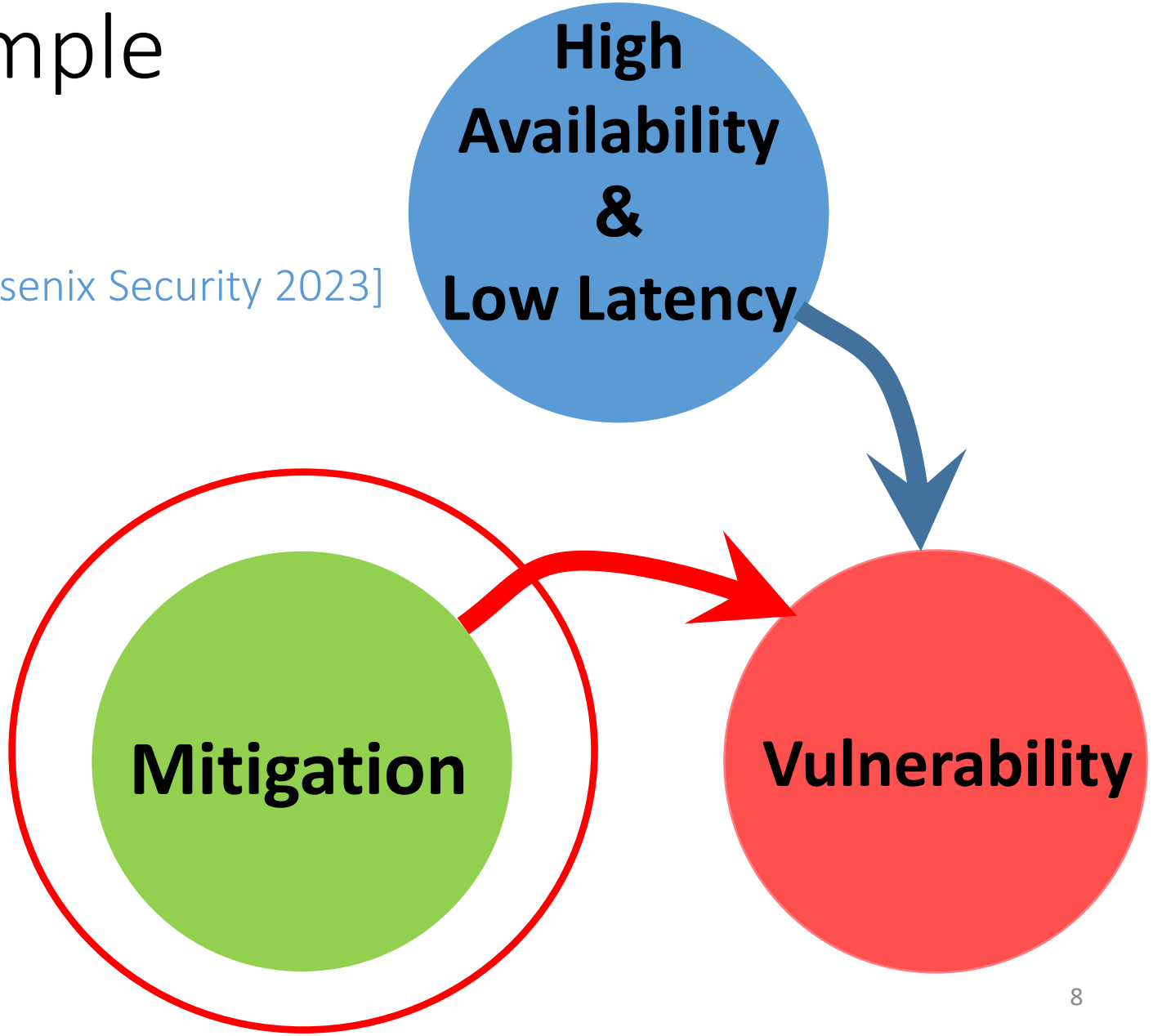
Resolve only 5

- MaxBreadth – [illegible]nd # of NS-names per referral response
- Detect NX NS repli[illegible]Labs)
- DNSSEC – NSEC (Petr Špaček)
- Going only downwards in the DNS hierarchy (draft rfc)

Second cycle example

NRDelegation Attack

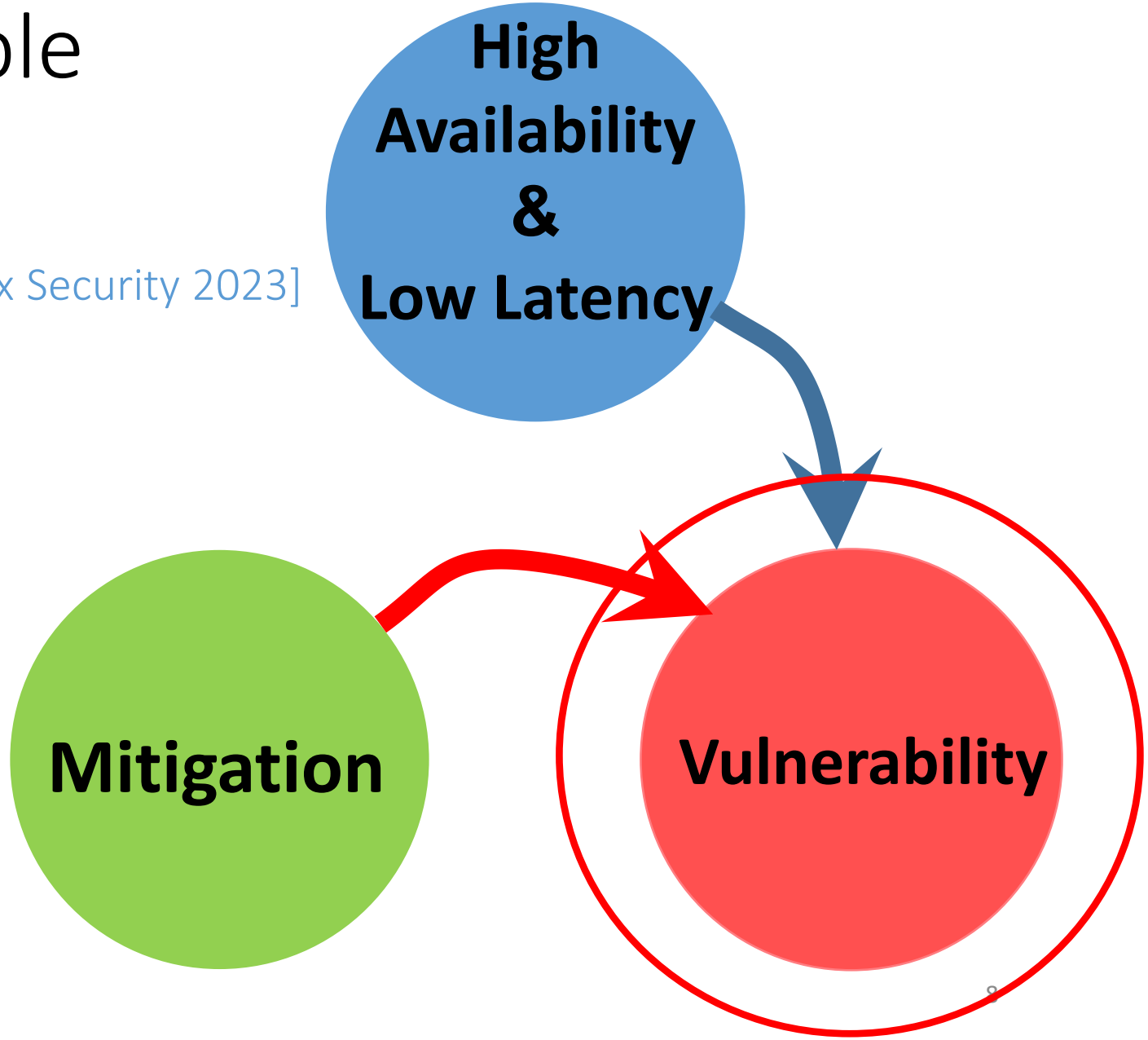
[A, Anat Bremler-Barr, Shani StajnrodUsenix Security 2023]



Second cycle example

NRDelegation Attack

[A, Anat Bremler-Barr, Shani StajnrodUsenix Security 2023]



NRDelegation Attack:

Complexity attack on the DNS Resolver

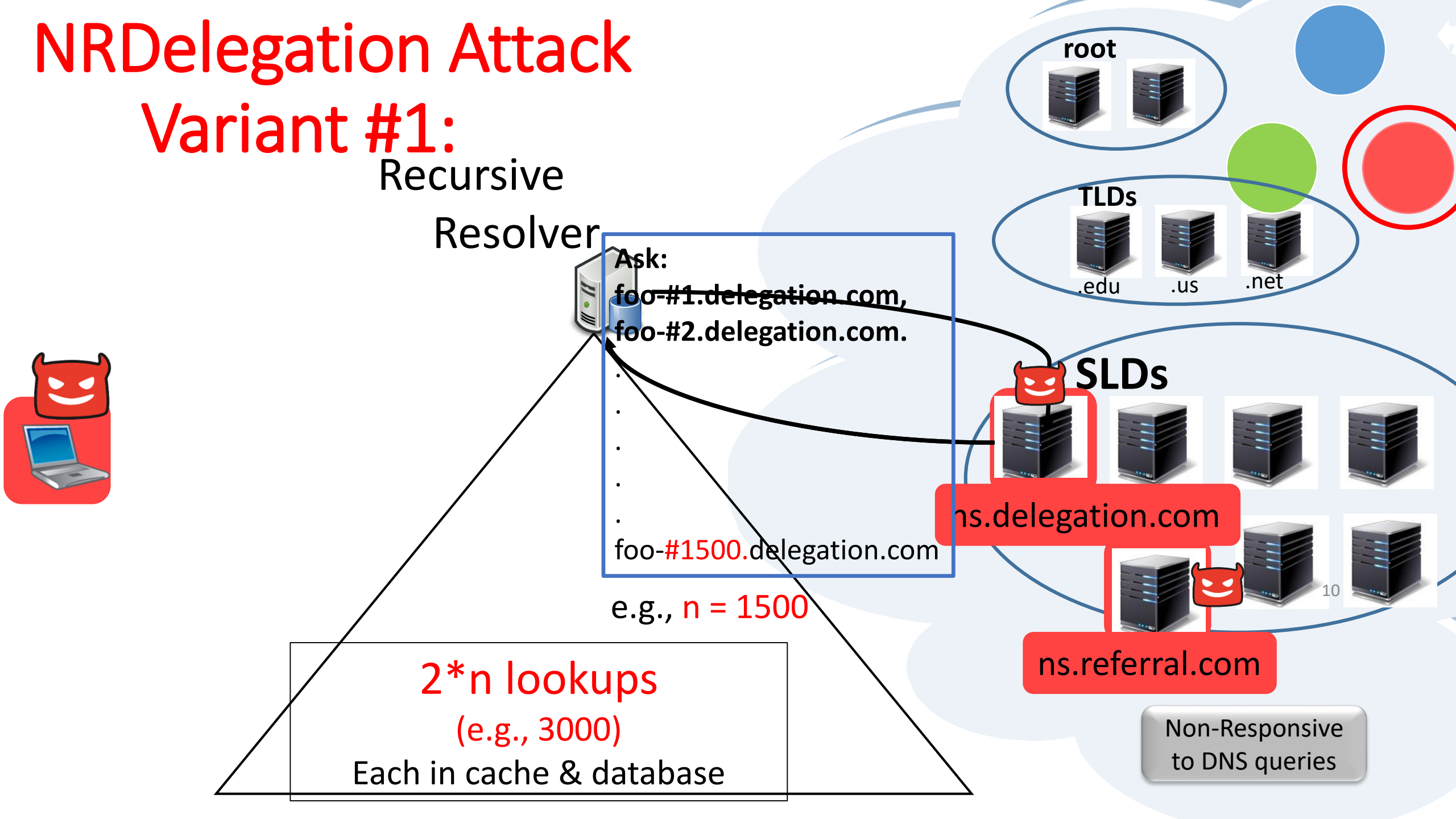
NRDelegation Attack:

Complexity attack on the DNS Resolver

Root cause – a single malicious query ➔ High CPU load.

Enabled by NXNS mitigation method !

NRDelegation Attack Variant #1:



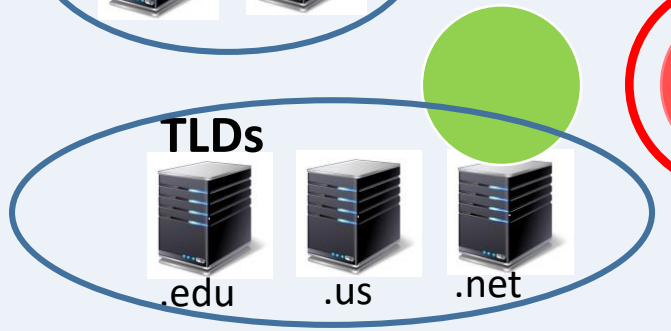
NRDelegation Attack Variant #1:

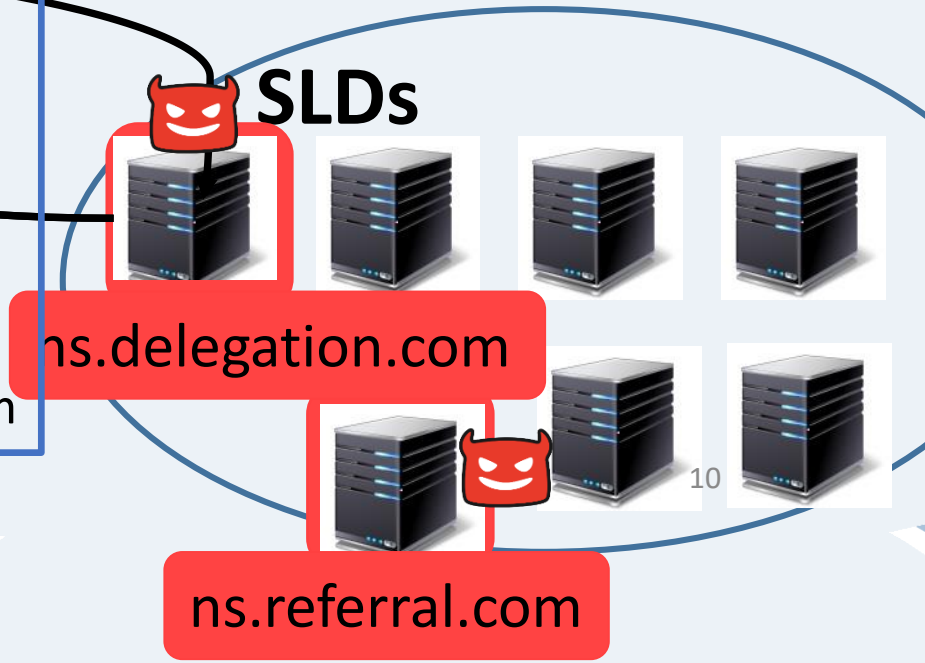
Recursive Resolver

2*n lookups
(e.g., 3000)
Each in cache & database

Ask:
~~foo-#1.delegation.com,~~
foo-#2.delegation.com.
.
.
.
.
.
foo-#1500.delegation.com

1. 2*n lookups (=3000)





Non-Responsive to DNS queries

NRDelegation Attack Variant #1:

Recursive Resolver

NXNS Mitigation

Ask:

~~**foo-#1.delegation.com,**~~

foo-#2.delegation.com.

.

.

.

.

.

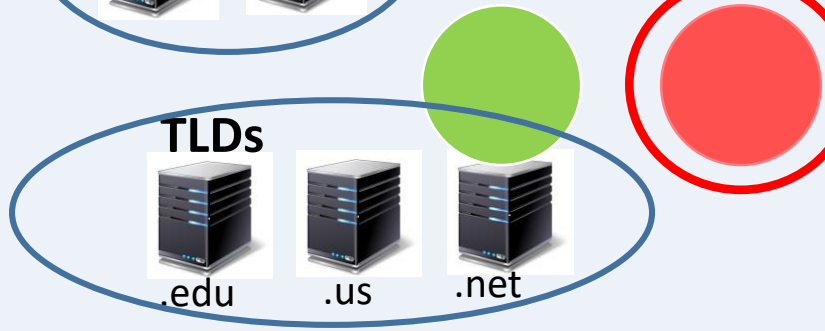
foo-#1500.delegation.com

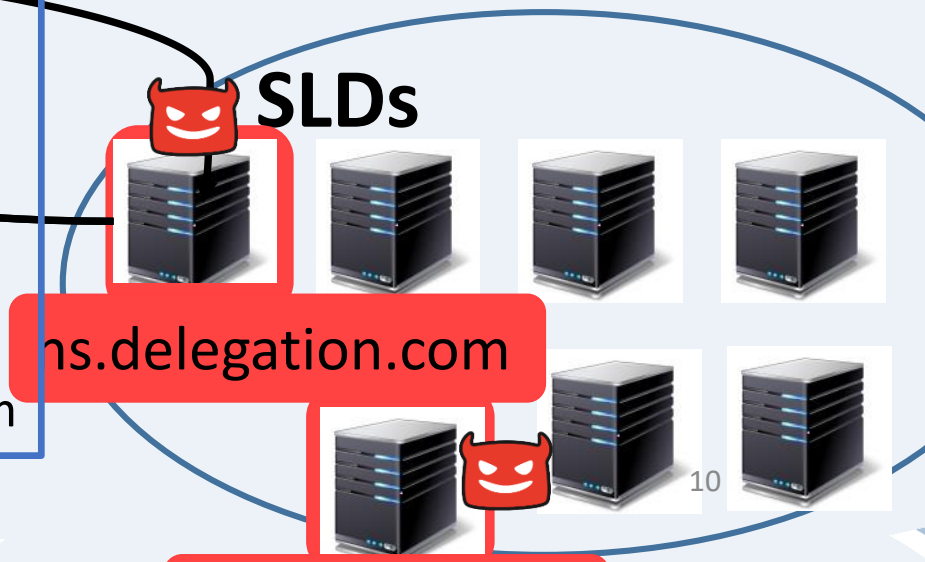
1. 2*n lookups (=3000)

n > referral-limit=5

Resolve only 5 NS names,

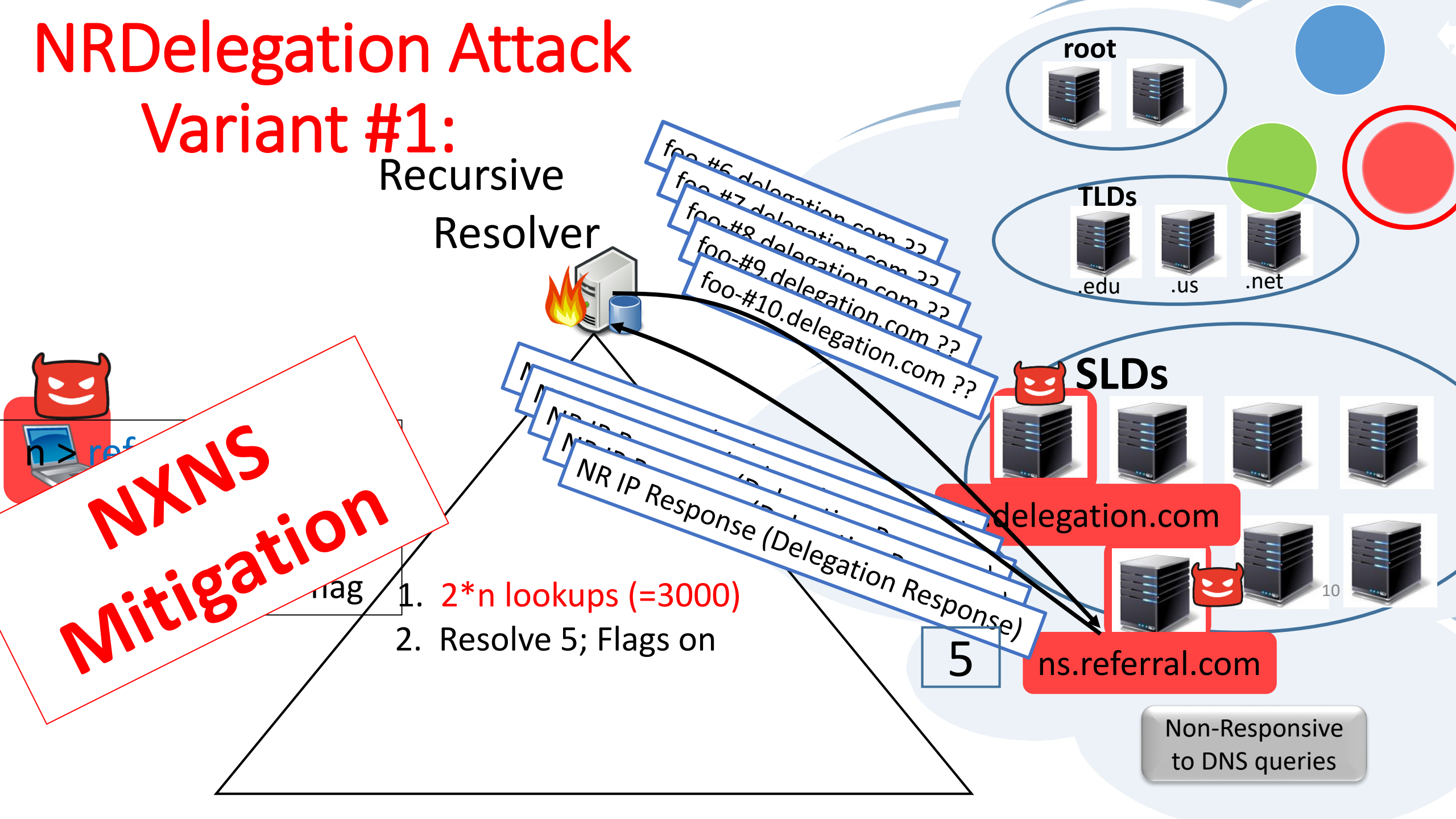
turn on "No_Fetch" flag



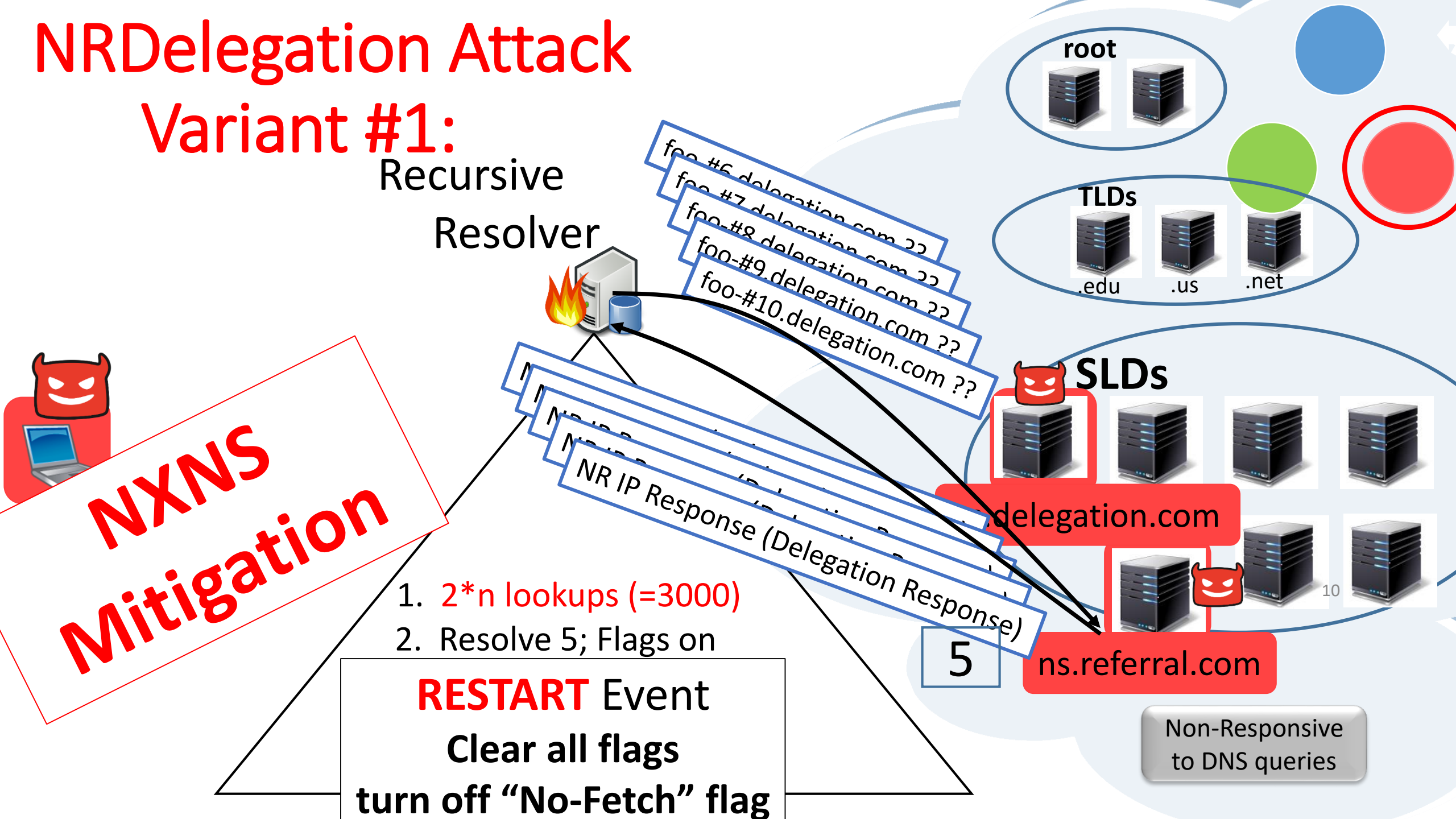


Non-Responsive to DNS queries

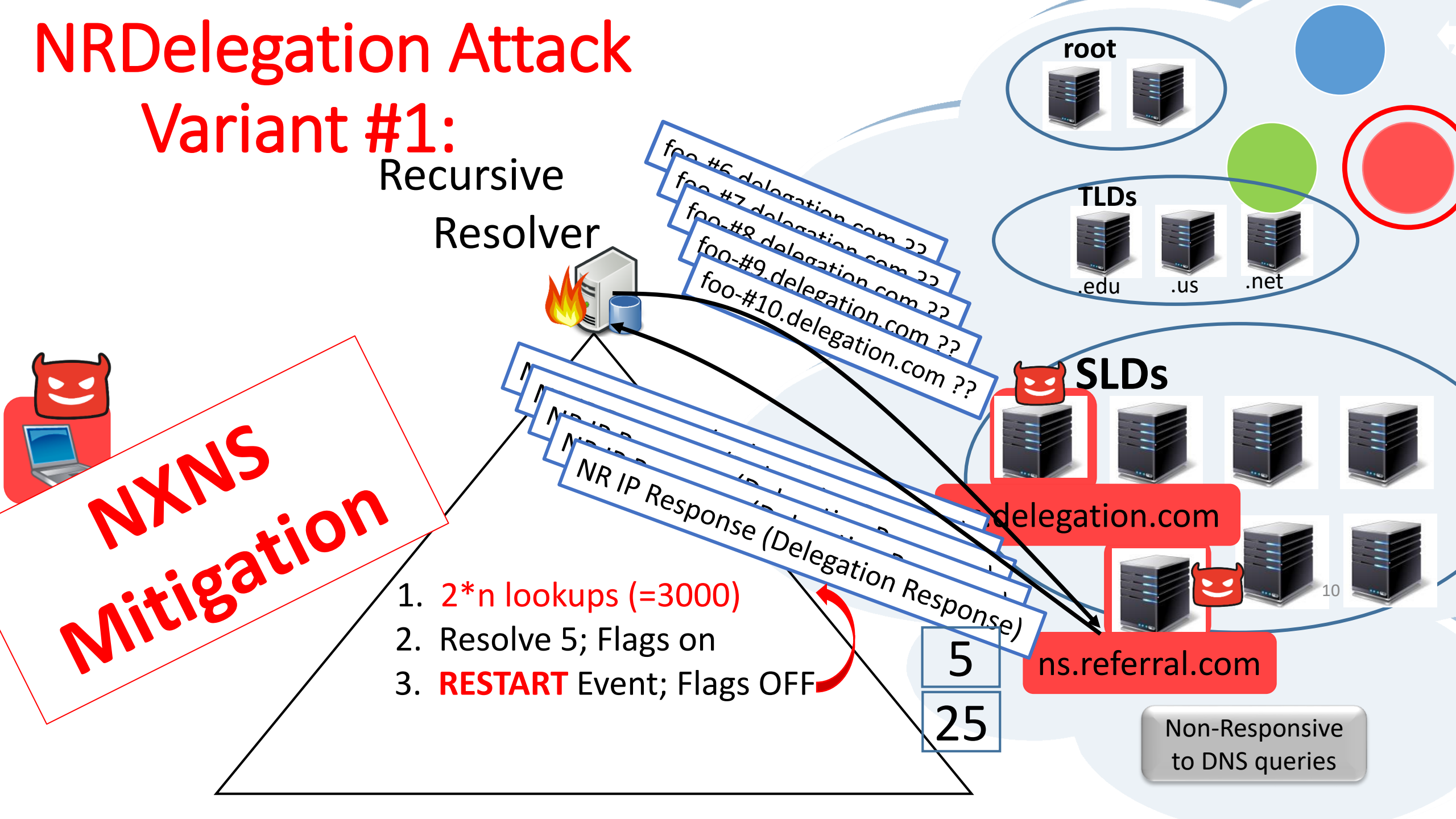
NRDelegation Attack Variant #1:



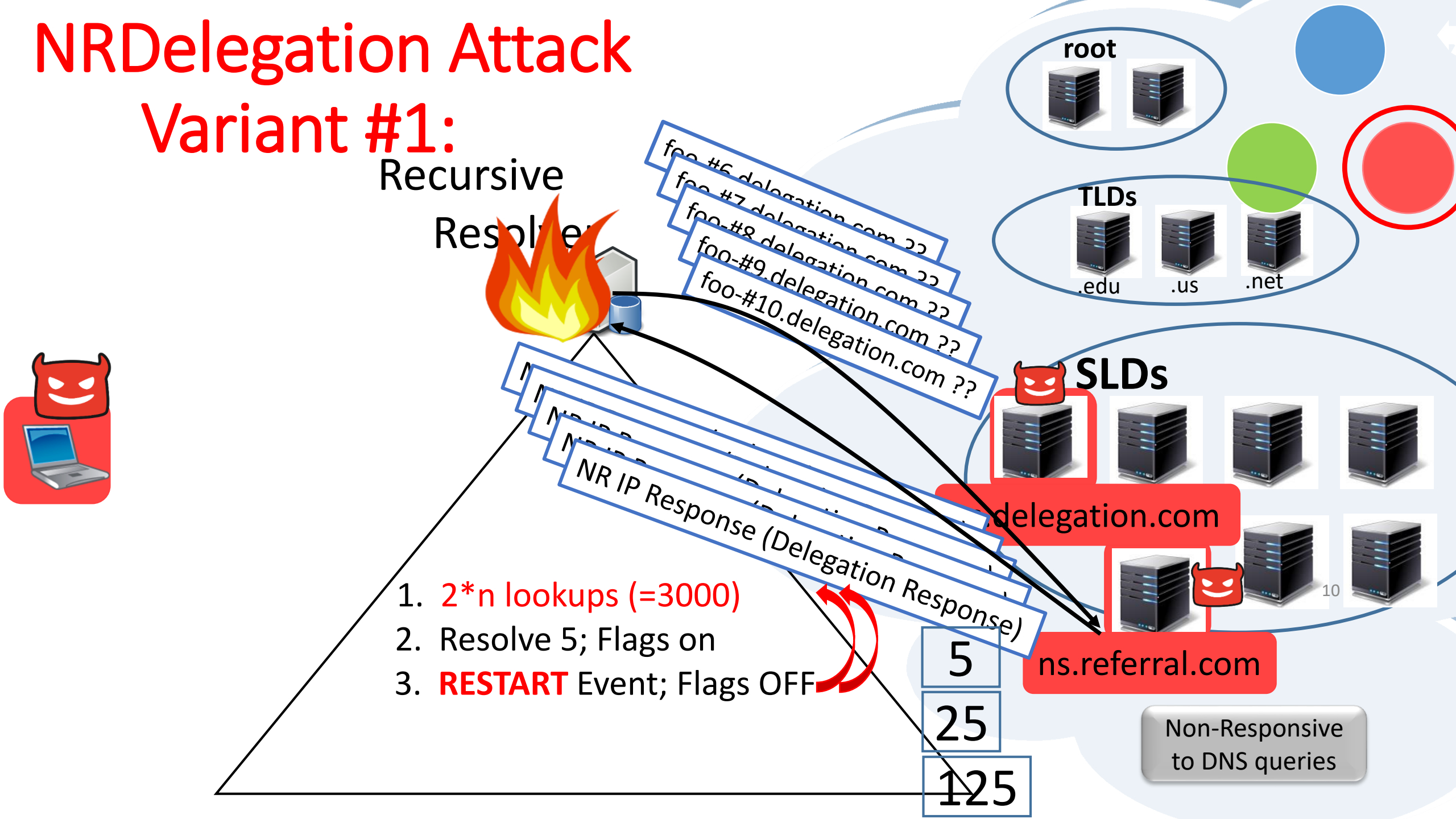
NRDelegation Attack Variant #1:



NRDelegation Attack Variant #1:

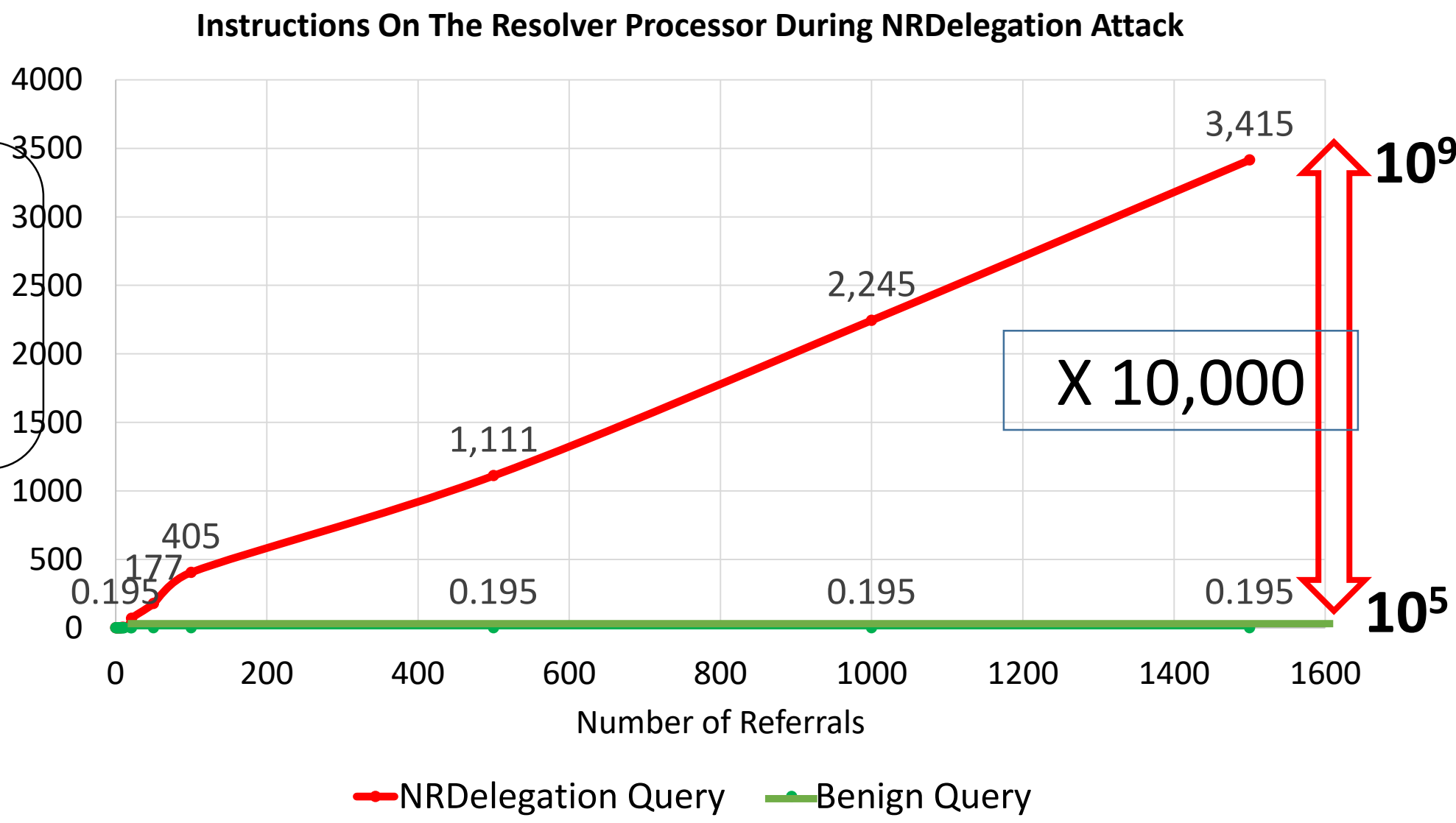


NRDelegation Attack Variant #1:

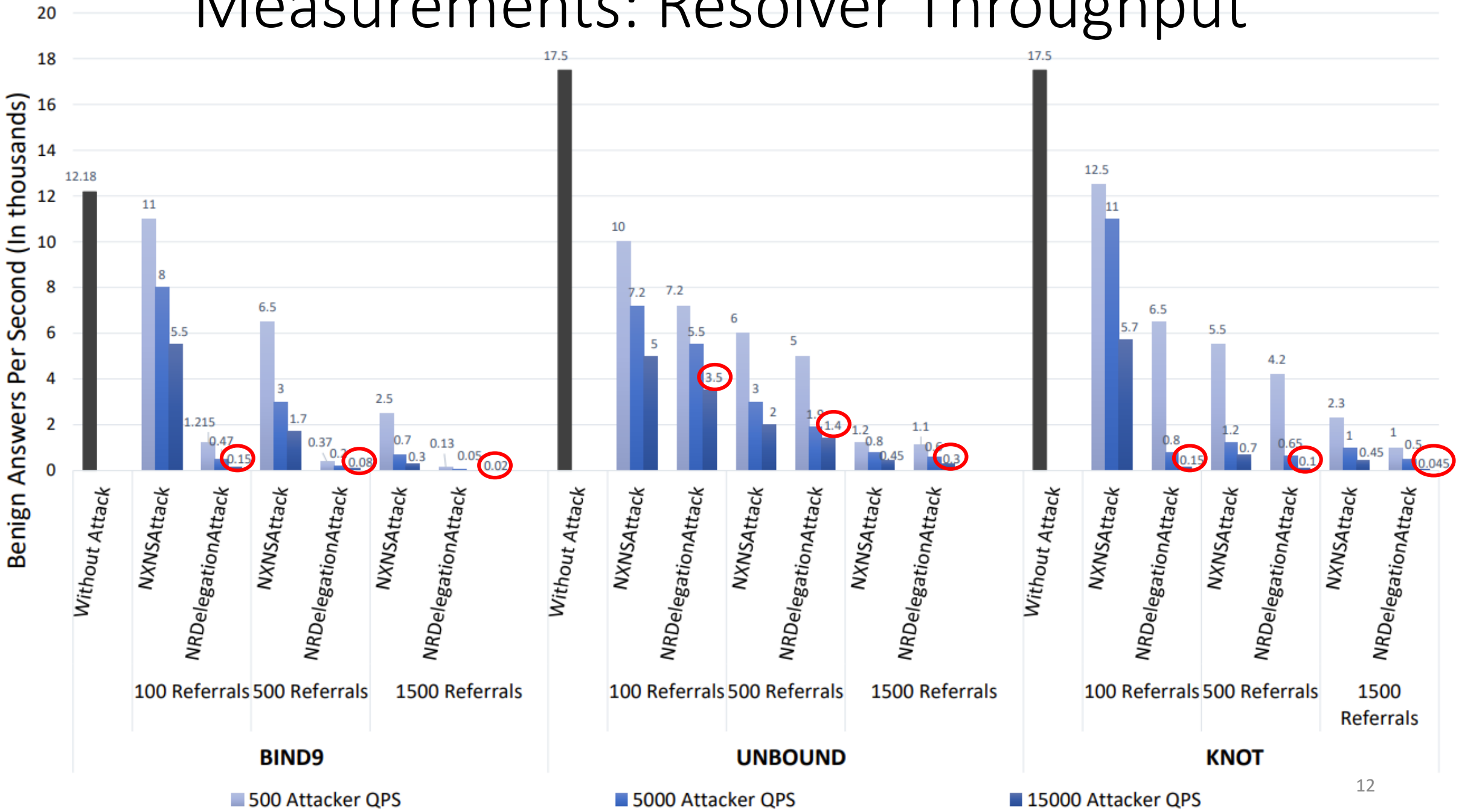


NRDelegation Complexity Amplification

Machine Instructions /req.
(millions)

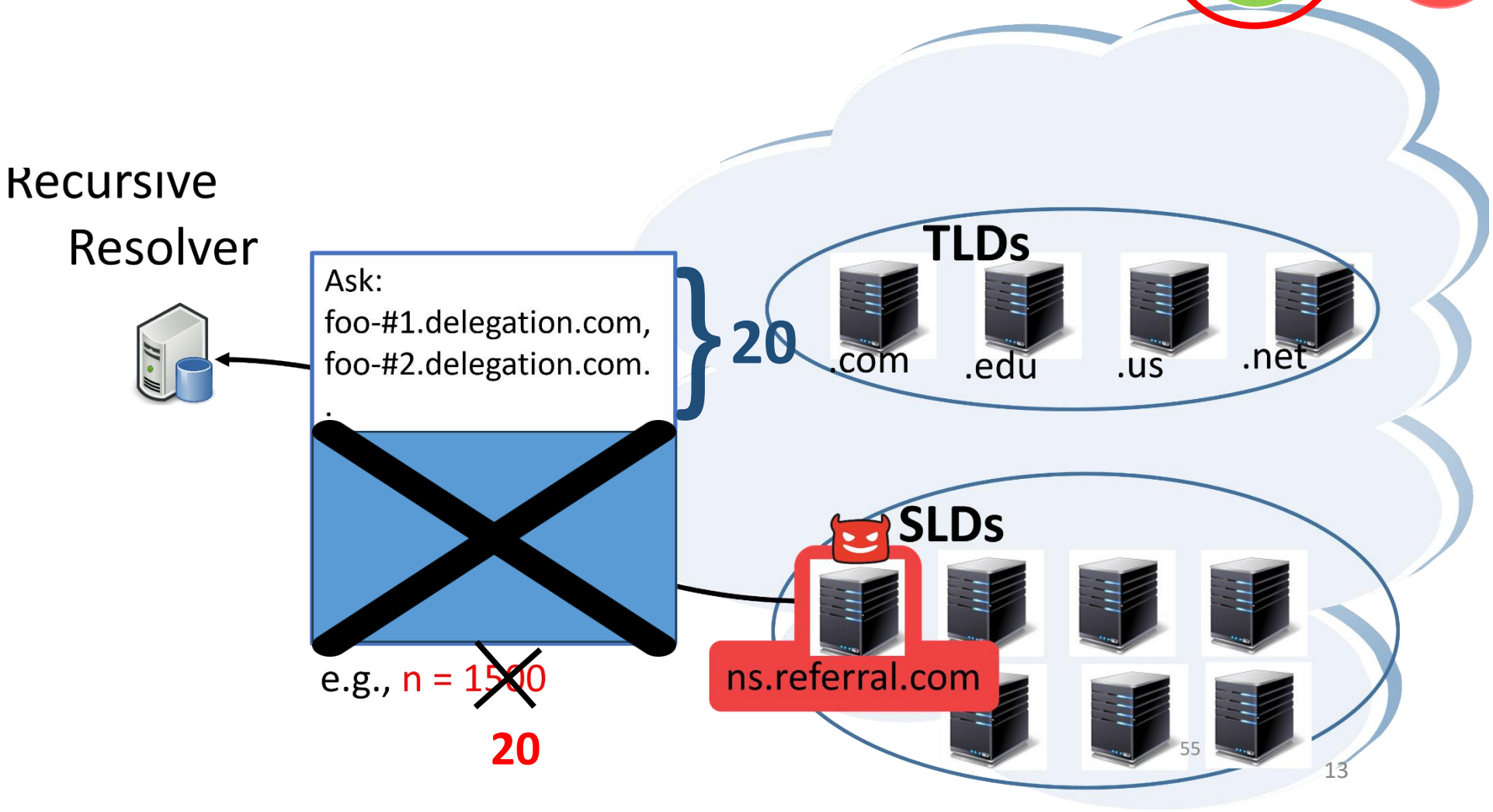


Measurements: Resolver Throughput



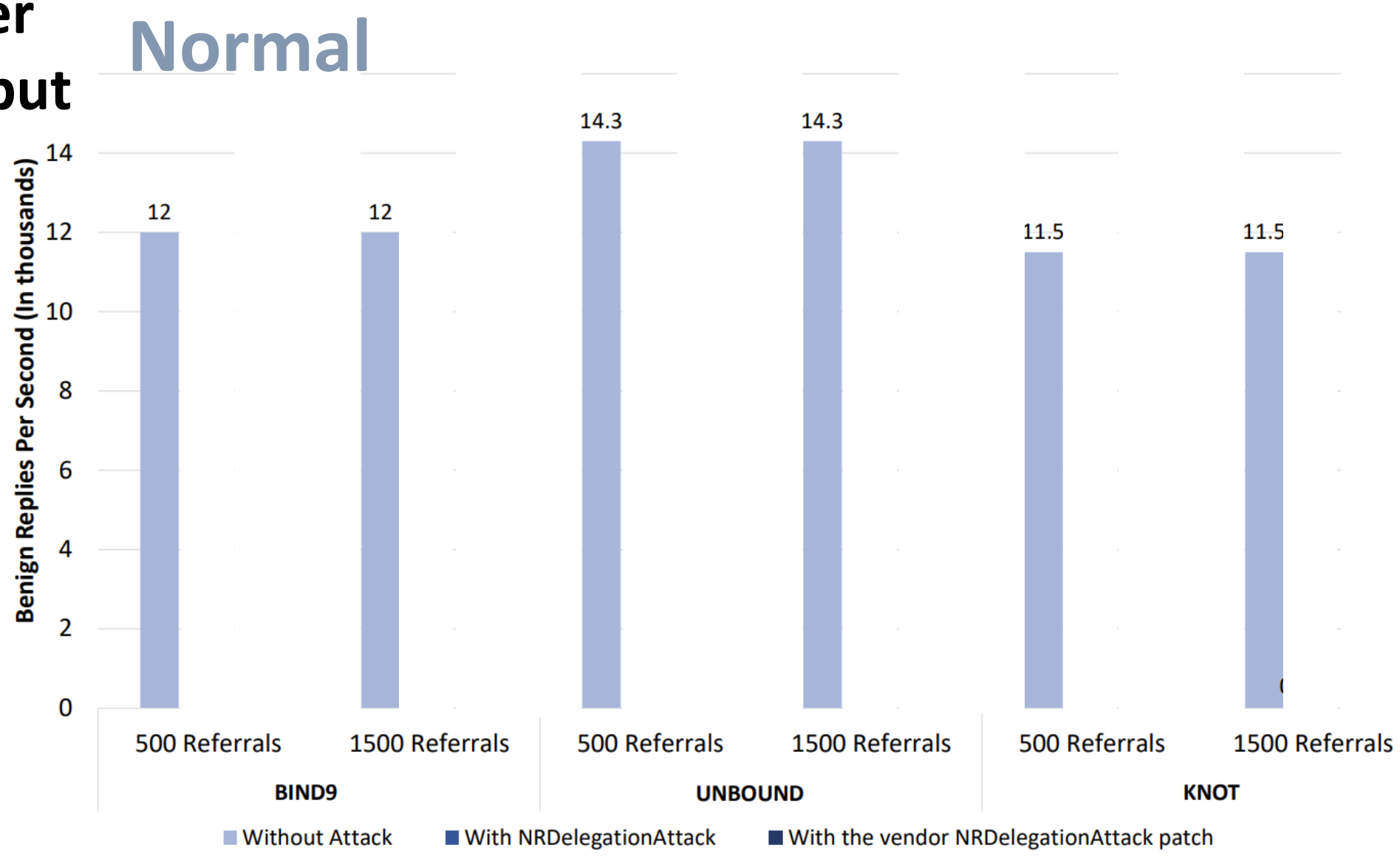
Mitigation: NRDelegation Attack

- **Process only *k (= 20)* NS names in the referral response**



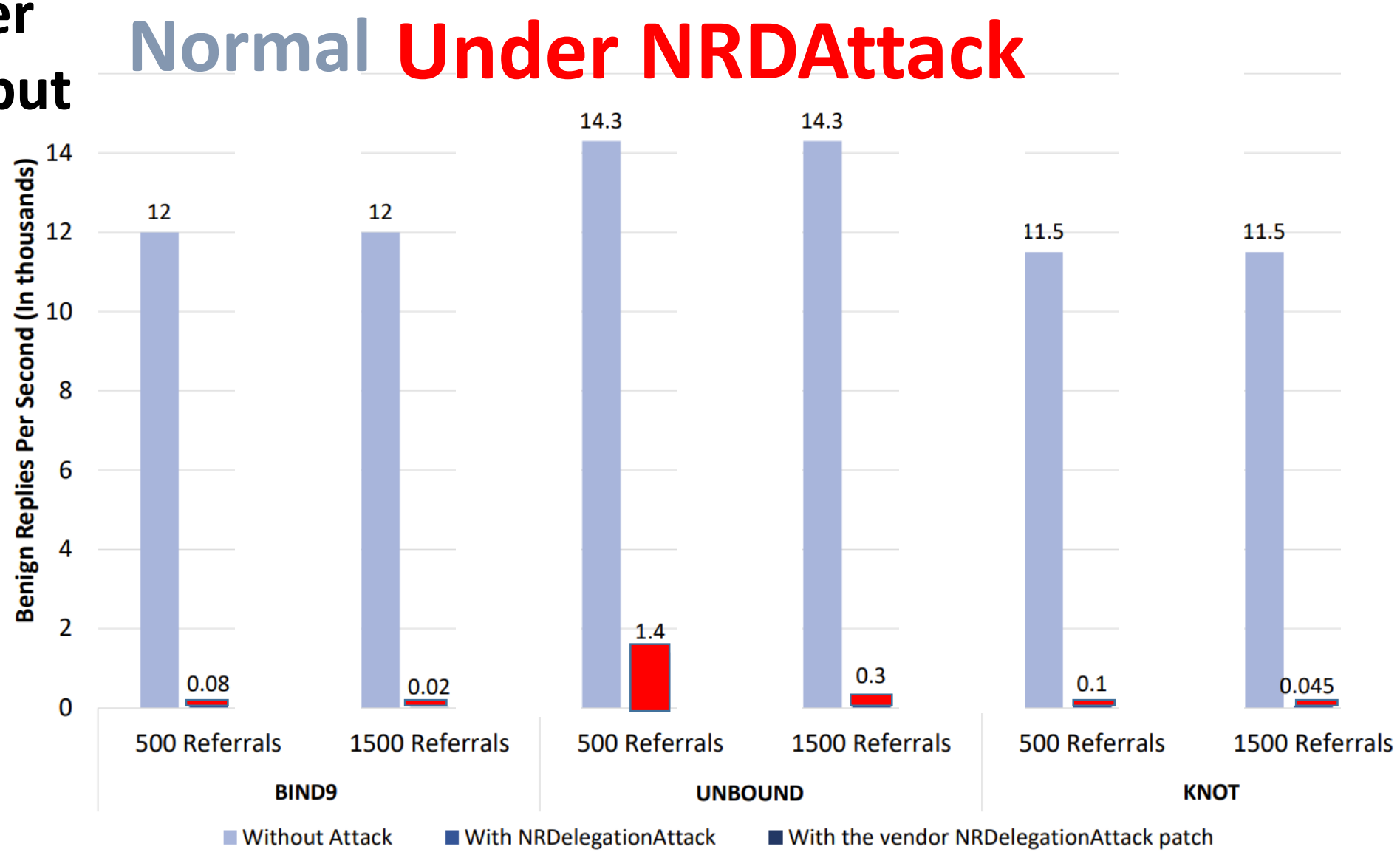
Throughput Using **NRDelegation** Mitigation

Resolver throughput



Throughput Using **NRDelegation** Mitigation

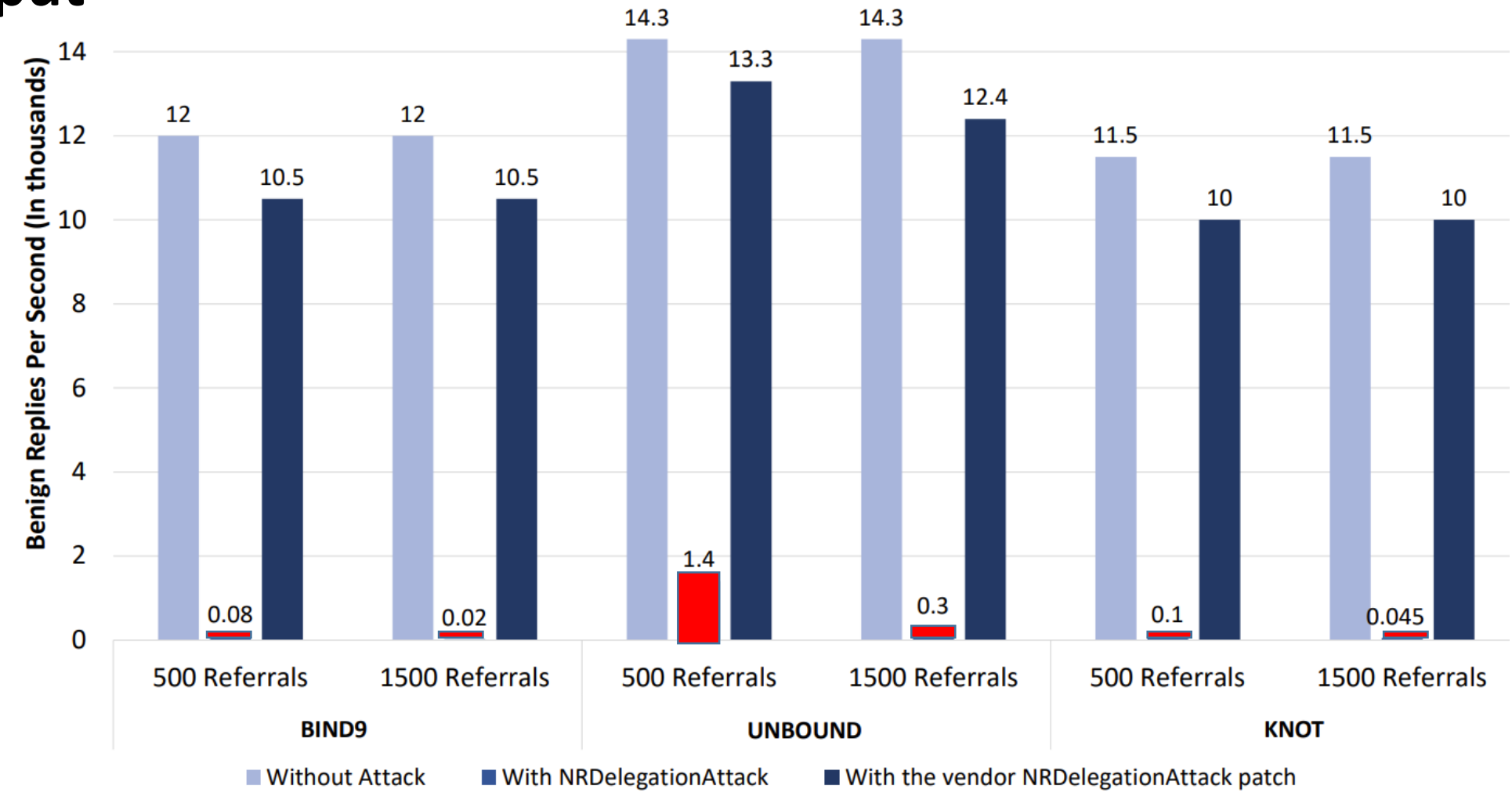
Resolver throughput



Throughput Using **NRDelegation** Mitigation

Resolver throughput

Normal **Under NRDAttack** **w/Mitigation**

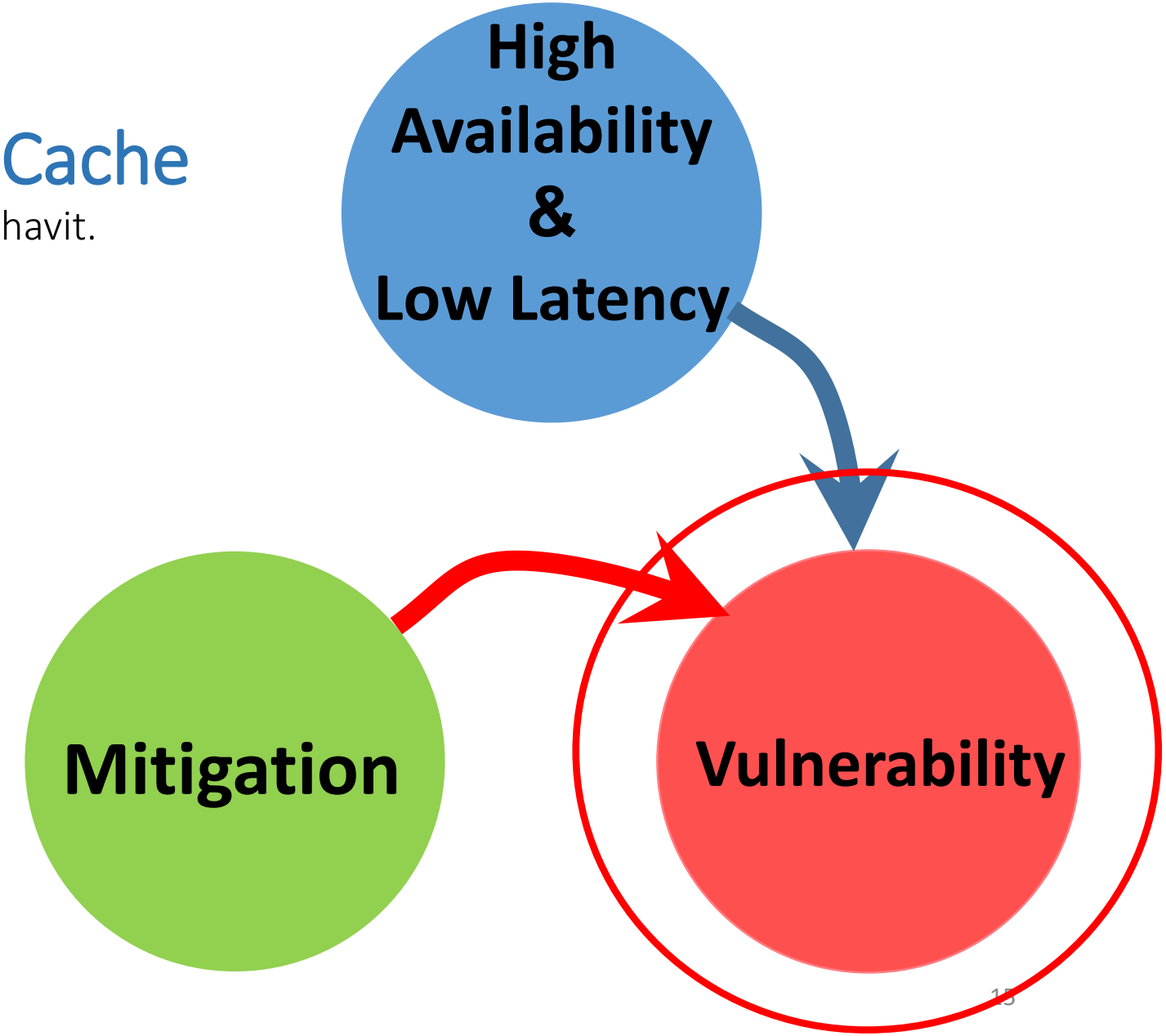


Third cycle example

CacheFlush: Flushing DNS Cache

[A, Anat Bremler-Barr, Shoham Danino, Yuval Shavit. Usenix Security 2024]

Flush the resolver **cache**



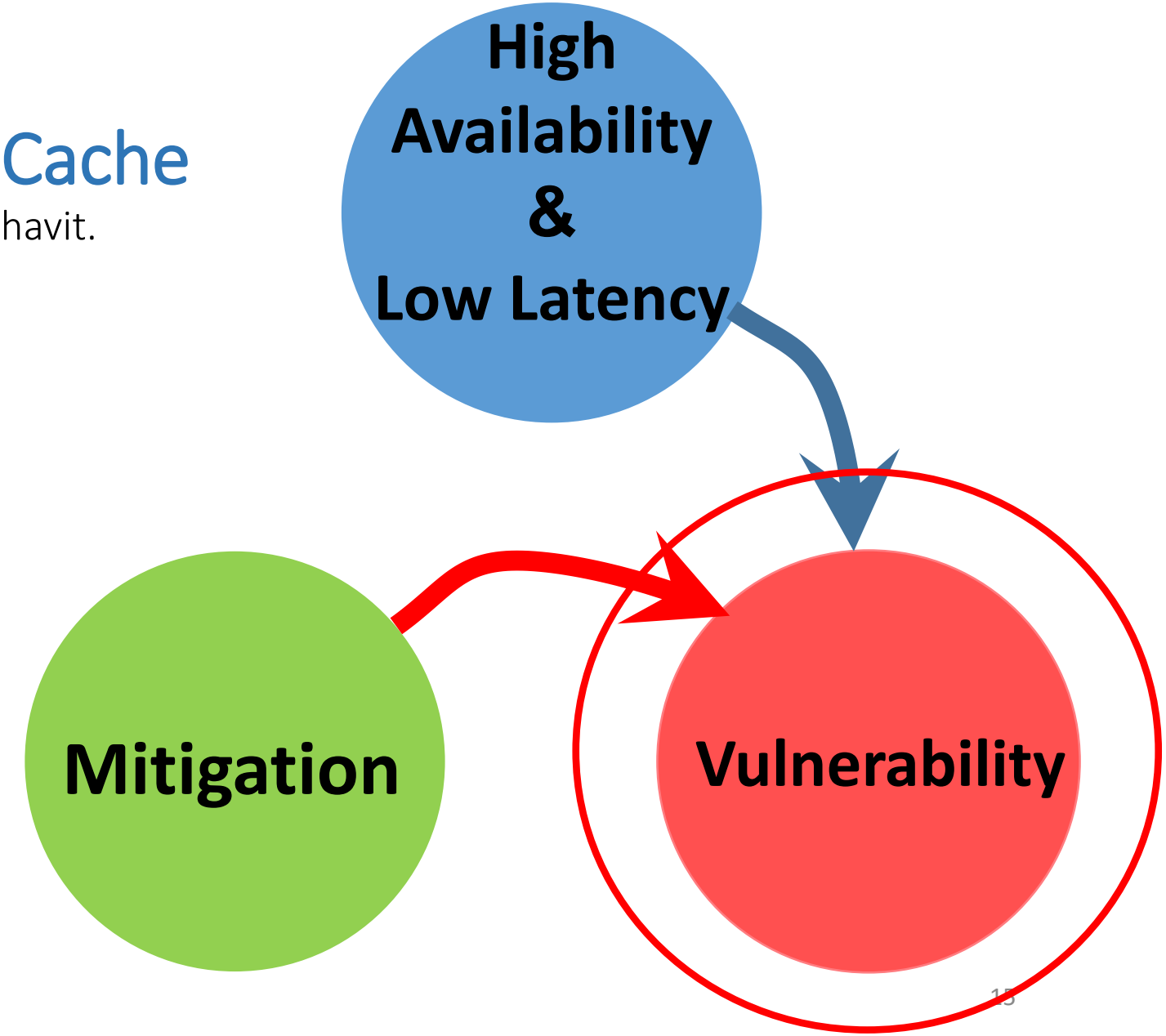
Third cycle example

CacheFlush: Flushing DNS Cache

[A, Anat Bremler-Barr, Shoham Danino, Yuval Shavit. Usenix Security 2024]

Flush the resolver **cache**

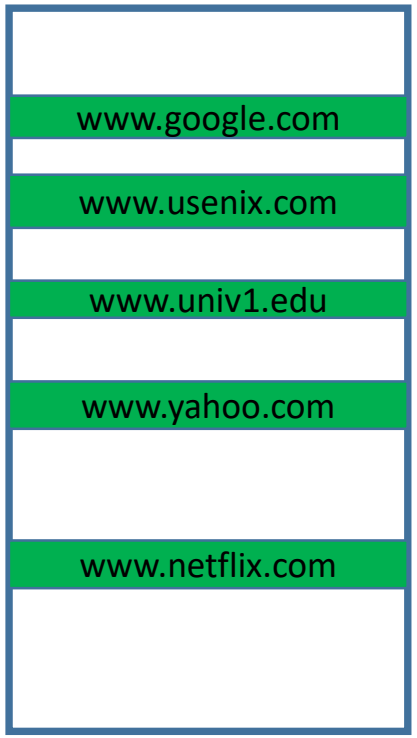
the **benign** cache !



- NXNS ➔ Resolve only 5 referrals
- NRD ➔ Ignore beyond 20 referrals

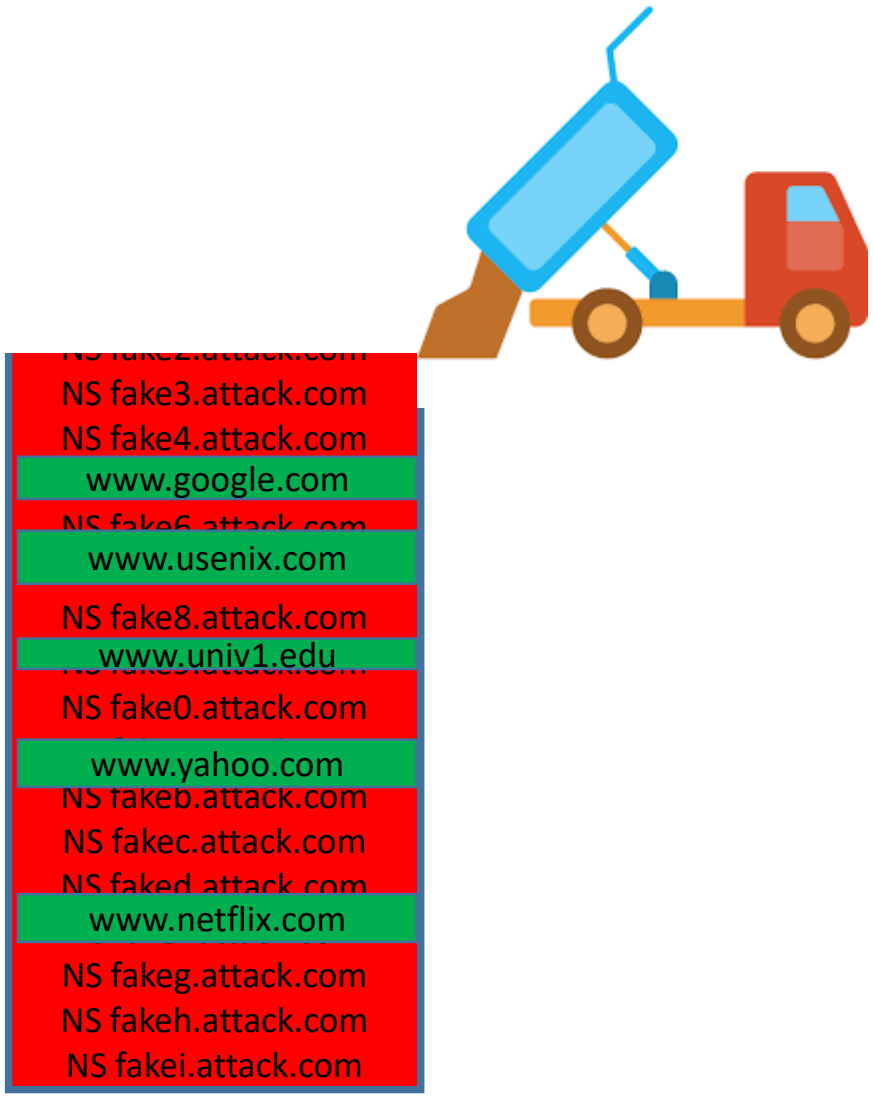
- NXNS ➔ Resolve only 5 referrals
- NRD ➔ Ignore beyond 20 referrals

- **But keep all RR in Benign Cache !**



Flushing the DNS Cache (LRU)

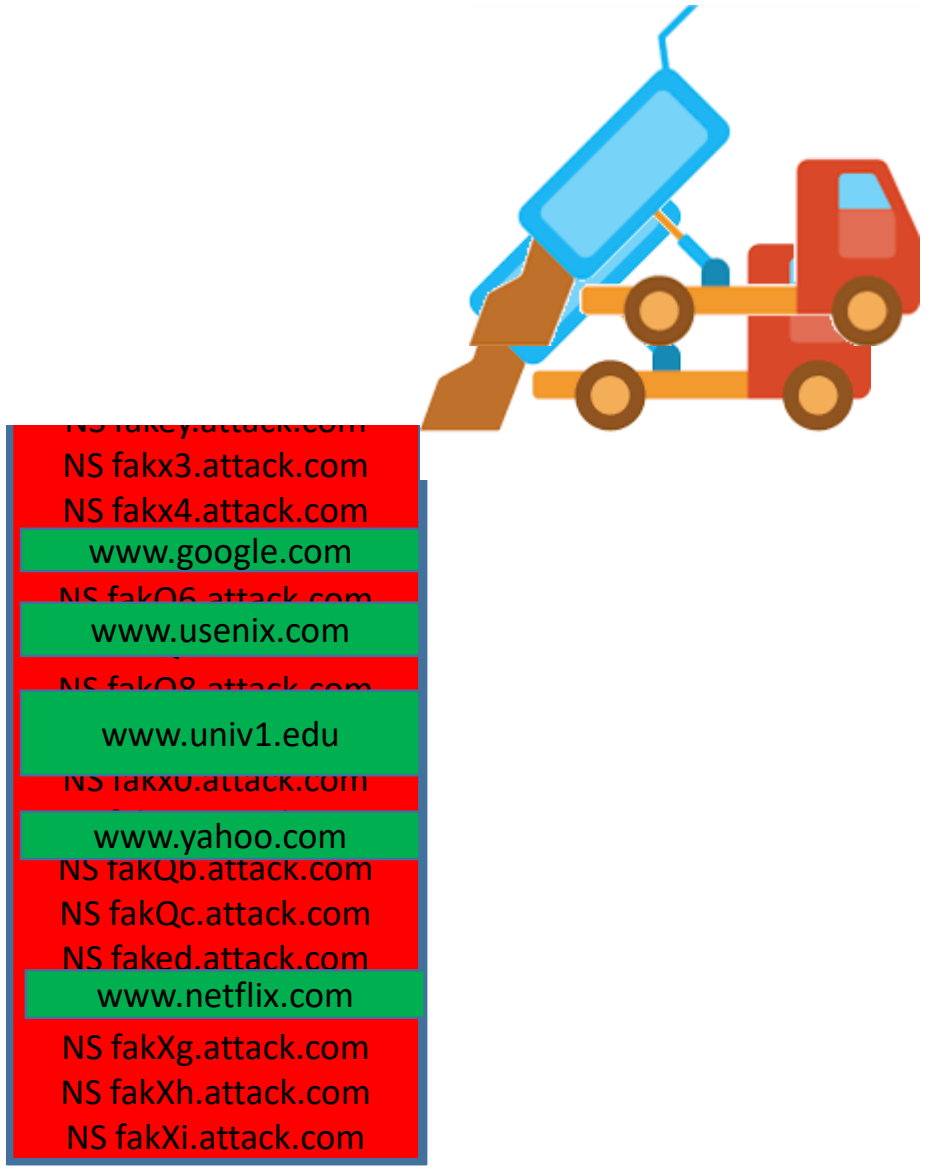
Flushing the DNS Cache (LRU)



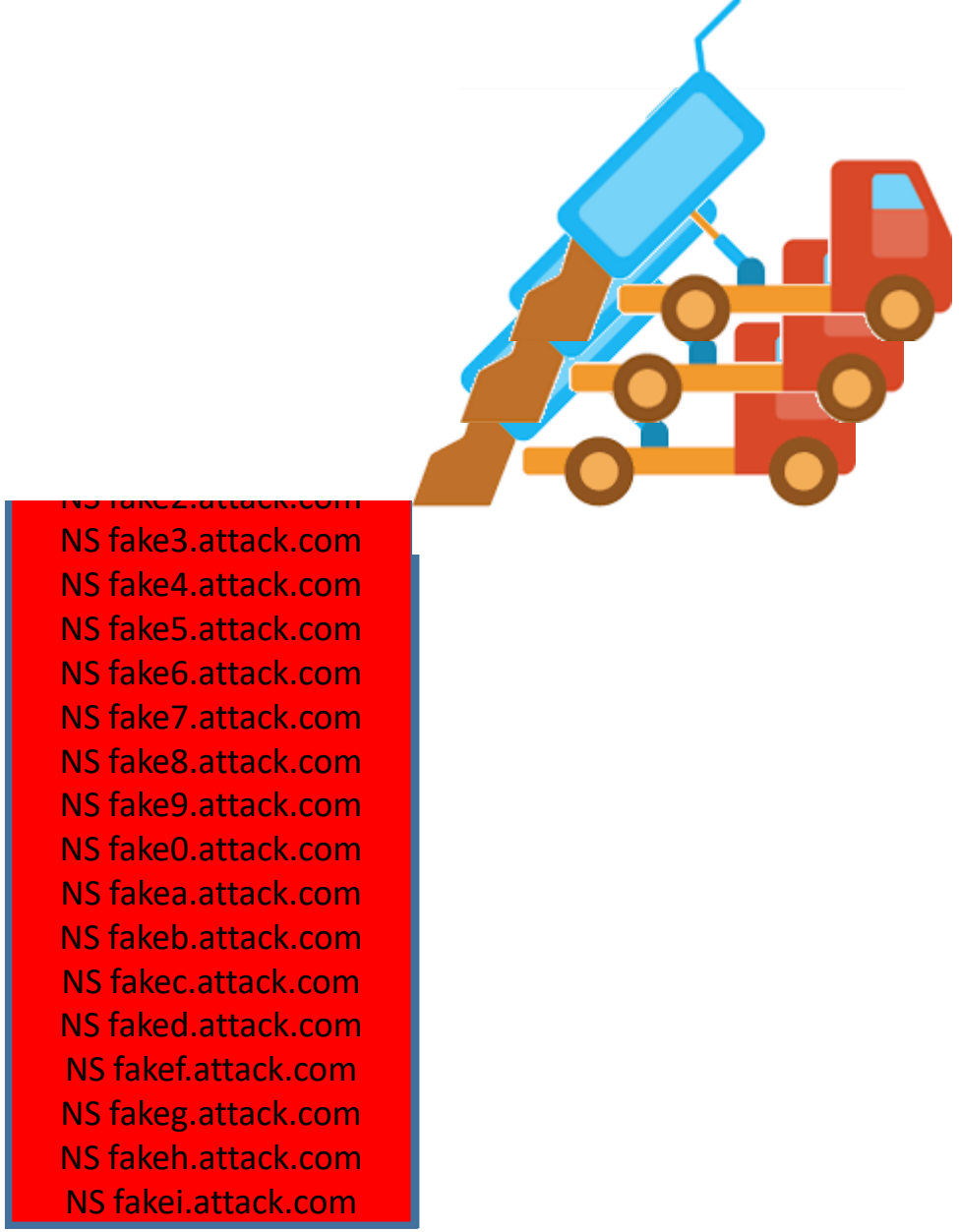
Flushing the DNS Cache (LRU)

NS fakx3.attack.com
NS fakx4.attack.com
NS fakx5.attack.com
NS fakQ6.attack.com
NS fakQ7.attack.com
NS fakQ8.attack.com
NS fakW9.attack.com
NS fakx0.attack.com
NS fakQa.attack.com
NS fakQb.attack.com
NS fakQc.attack.com
NS faked.attack.com
NS fakef.attack.com
NS fakXg.attack.com
NS fakXh.attack.com
NS fakXi.attack.com

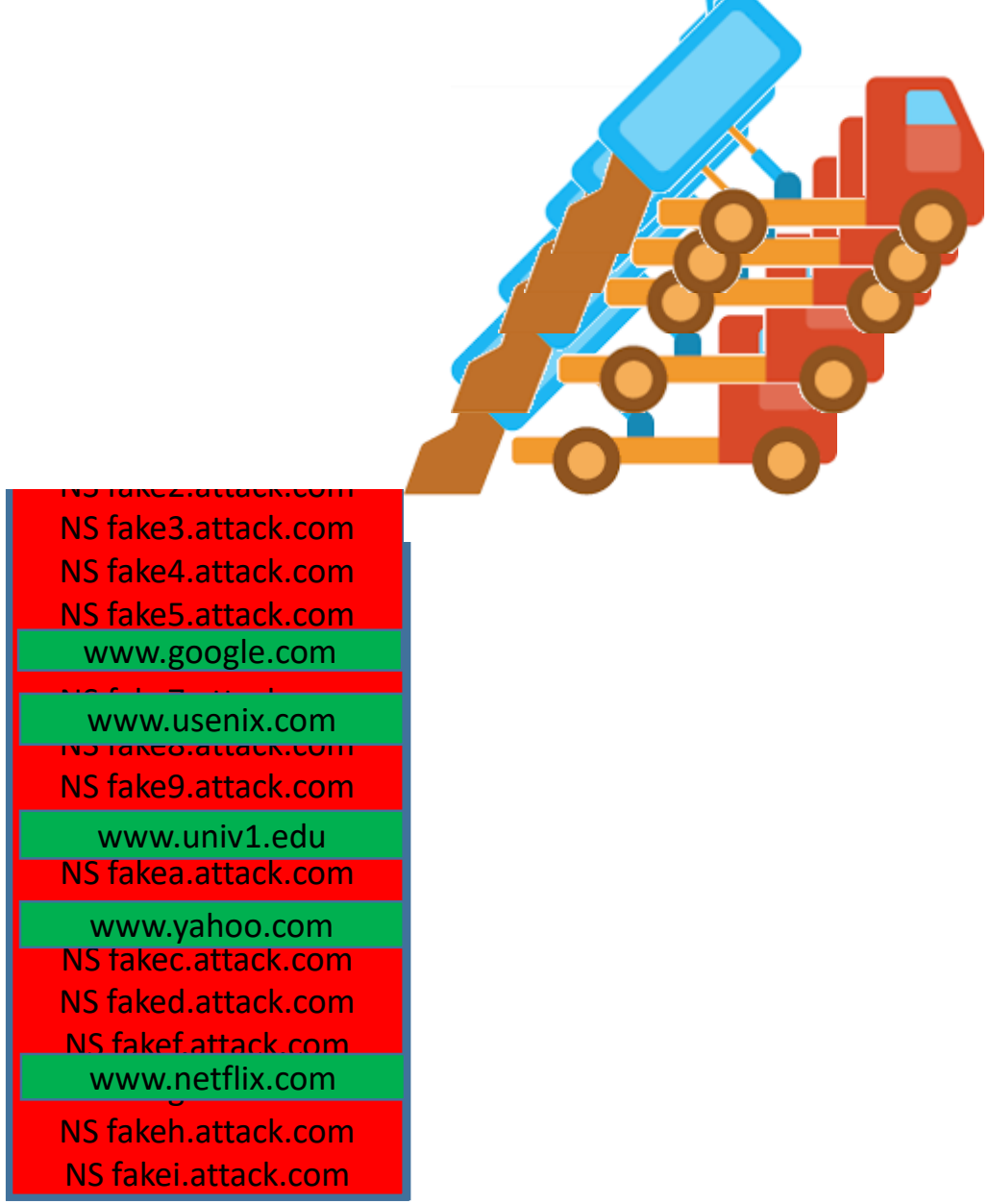
Flushing the DNS Cache (LRU)



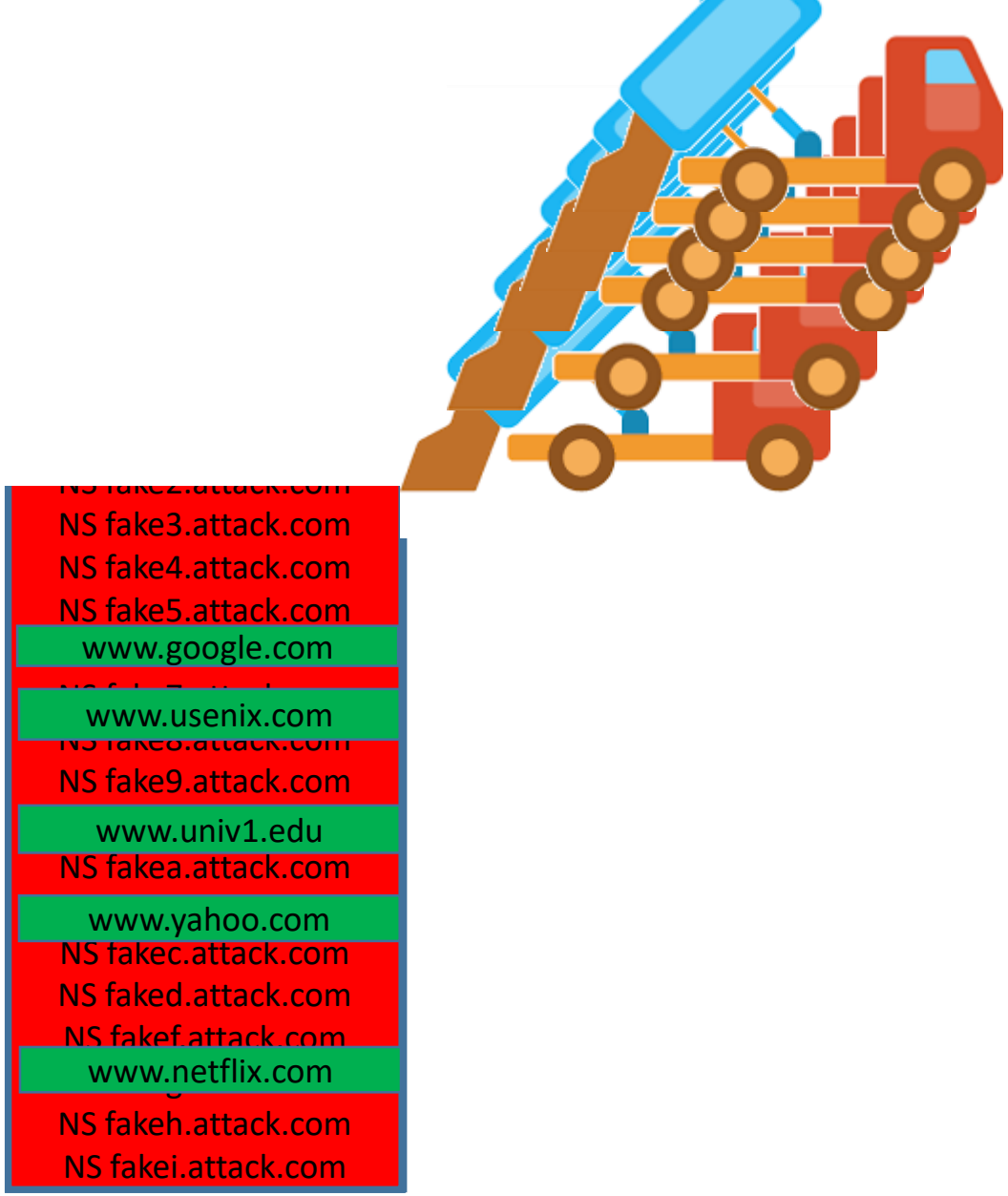
Flushing the DNS Cache (LRU)



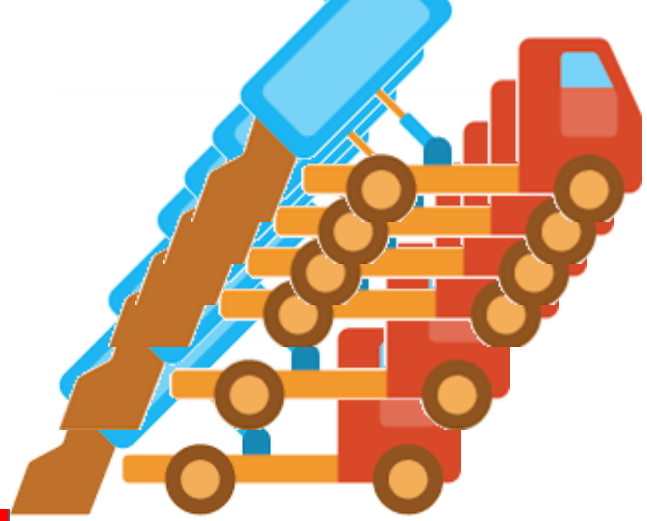
Flushing the DNS Cache (LRU)



Flushing the DNS Cache (LRU)



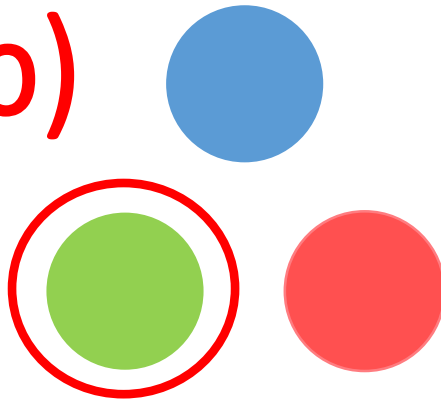
Flushing the DNS Cache (LRU)



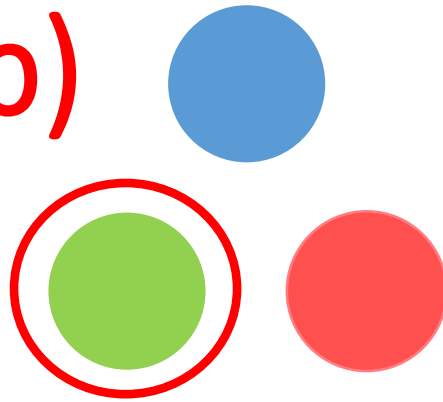
Knot not affected.
Why?

Flushing the DNS Cache (LRU)

Mitigation: CacheFlush (& KeyTrap)



Mitigation: CacheFlush (& KeyTrap)



Discard any thing beyond 100 RR of any type and shape

Restrictions on RR processing

	Resolve	Validate/Process	Store/Cache
≤2019	All	All	All
NXNS 2020	≤ 5	All	All
NRDelegation 2023	≤ 5	≤ 20	All
CacheFlush 2024	≤ 5	≤ 20	≤ 100
KeyTrap 2024			≤ 100 Validate < 10

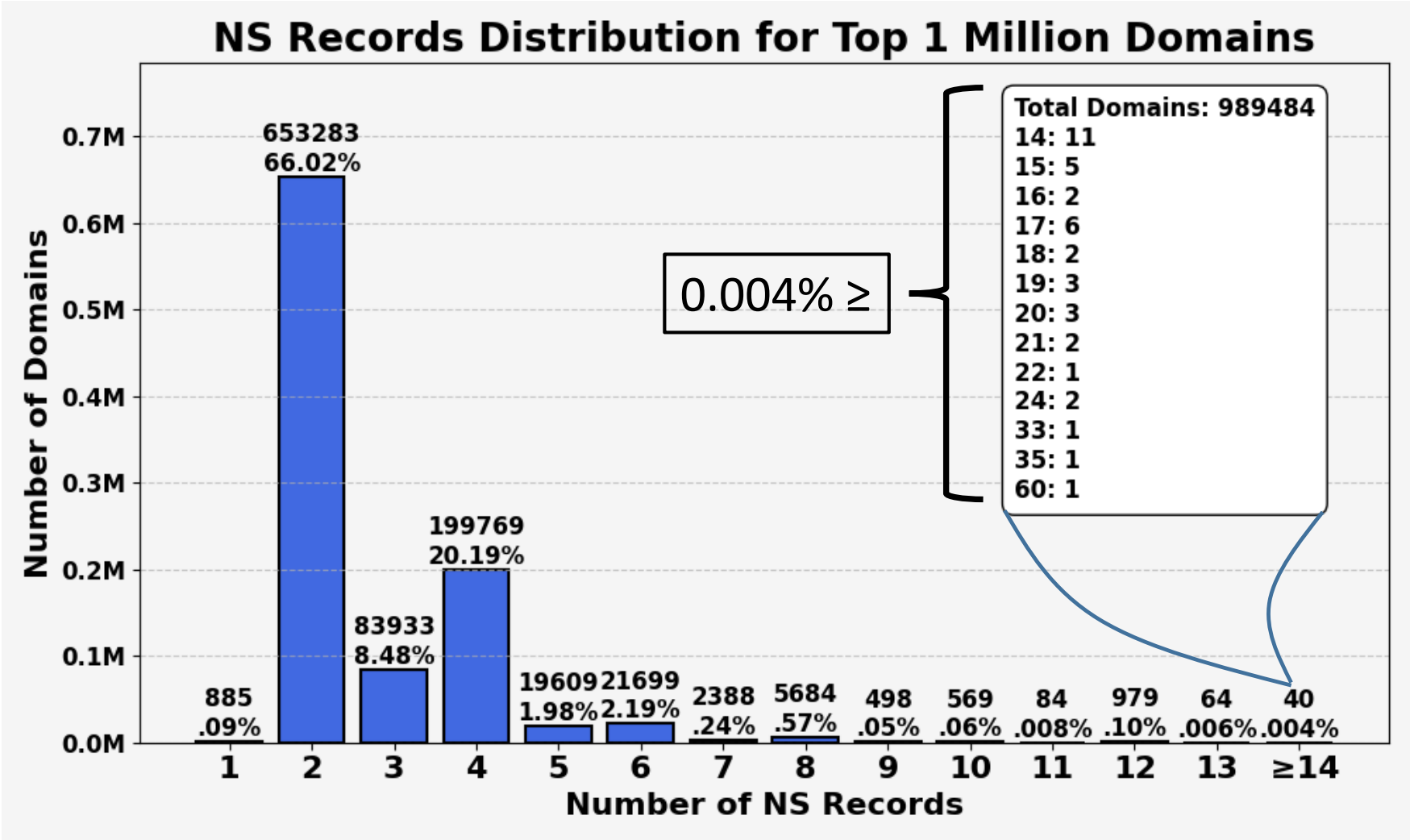
What should MAX #RR be?

- 100?

What should MAX #RR be?

- 100?

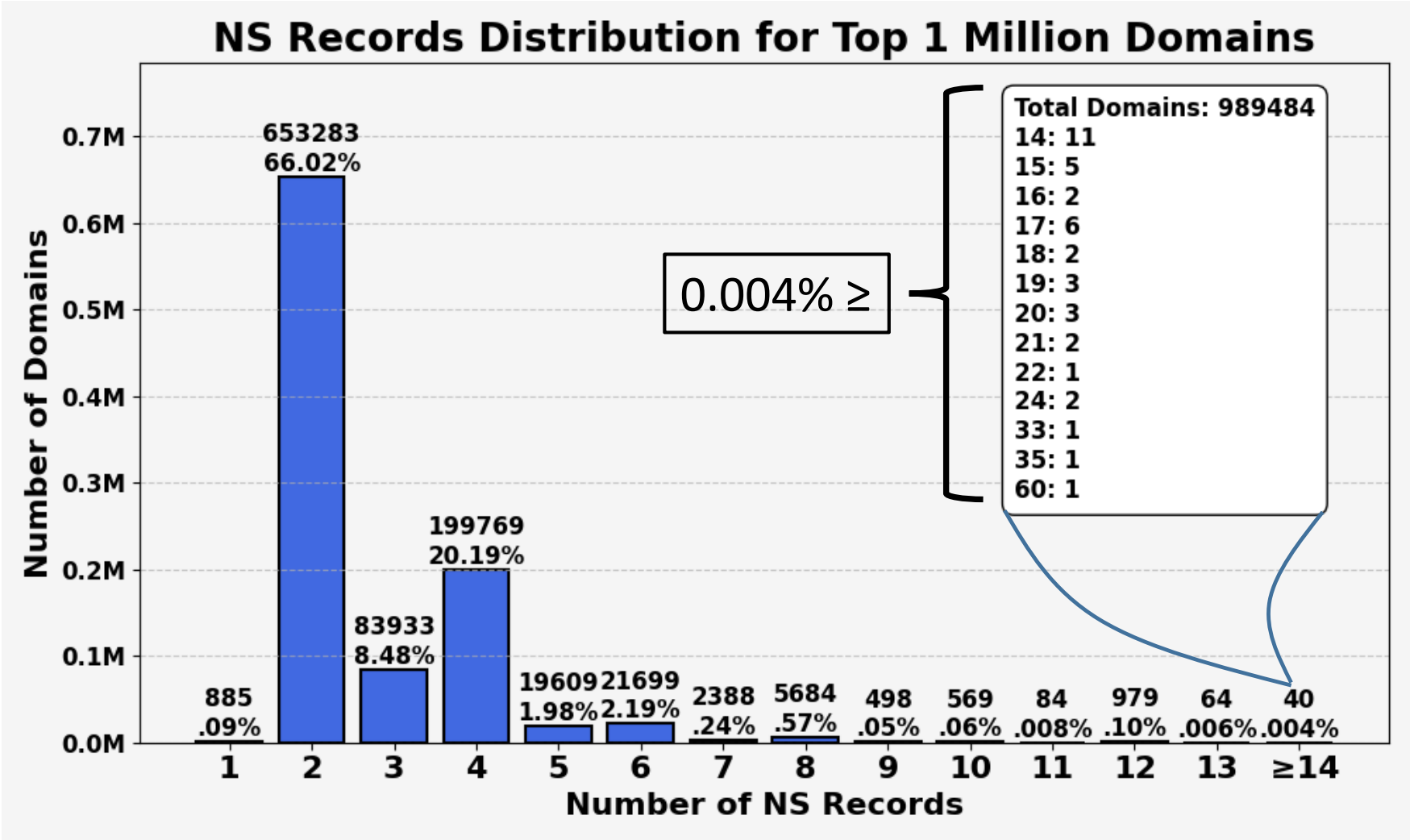
Measured 17/10/2024



What should MAX #RR be?

- 100? 20?

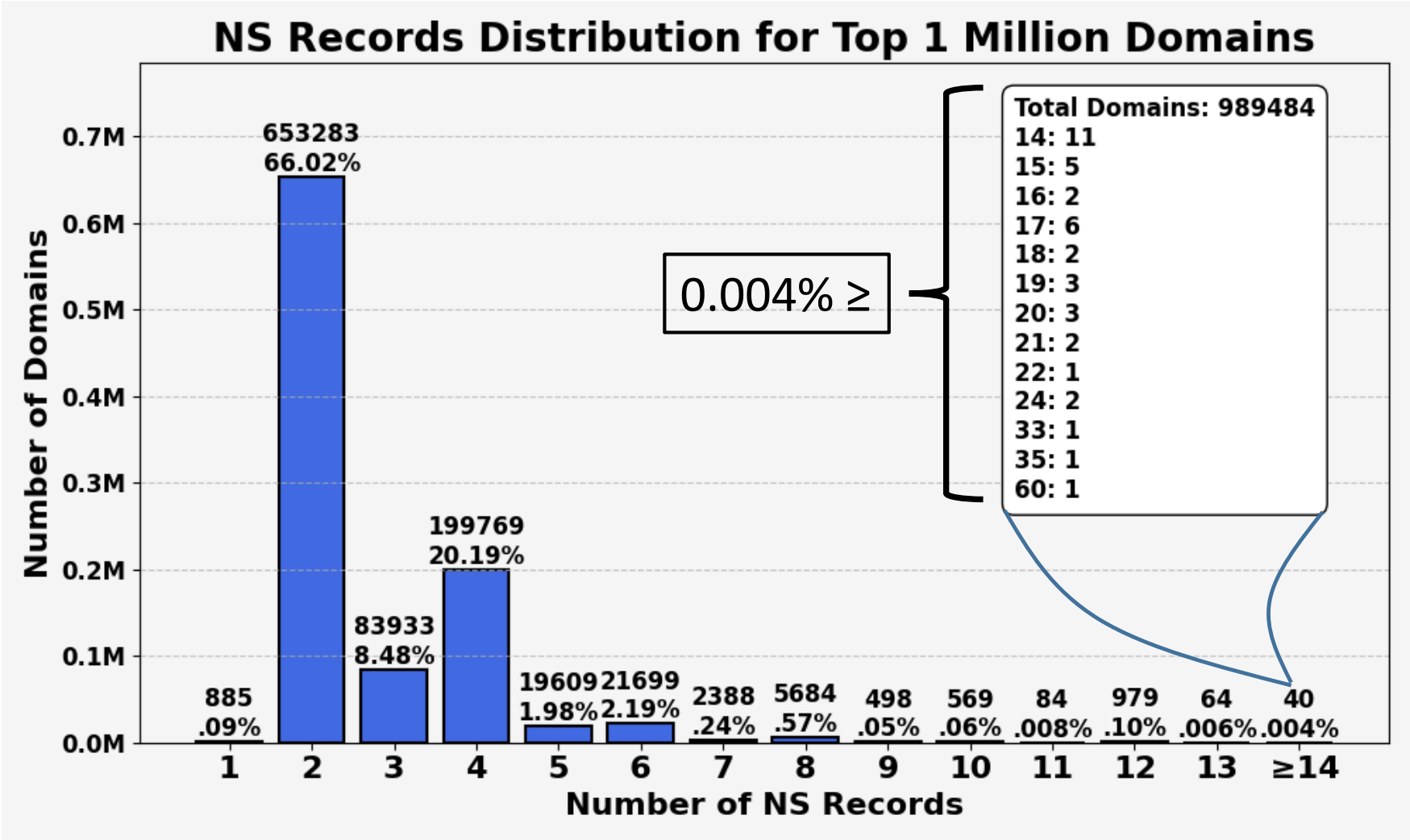
Measured 17/10/2024



What should MAX #RR be?

- 100? 20? 13?

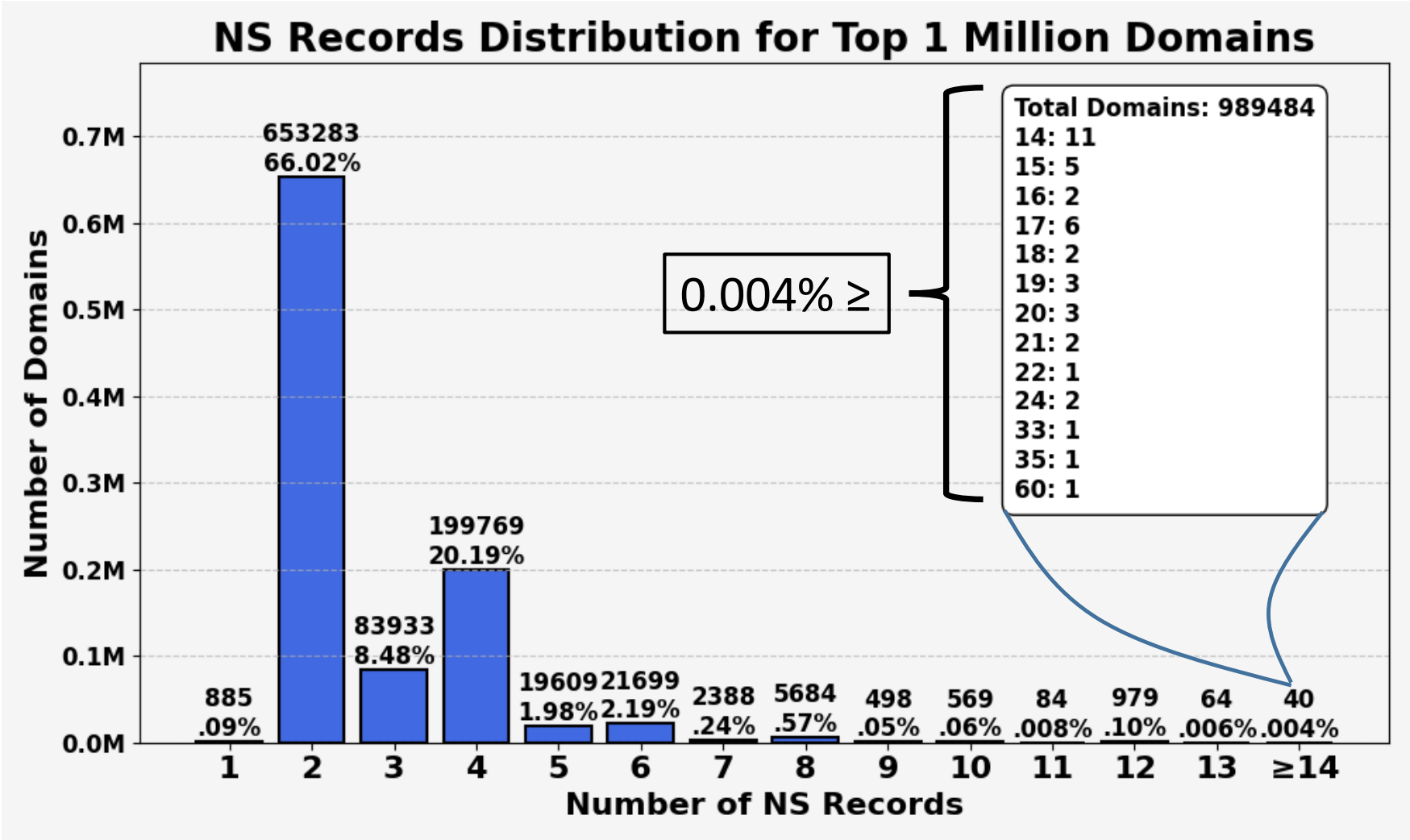
Measured 17/10/2024



What should MAX #RR be?

- 100? 20? 13? 10?

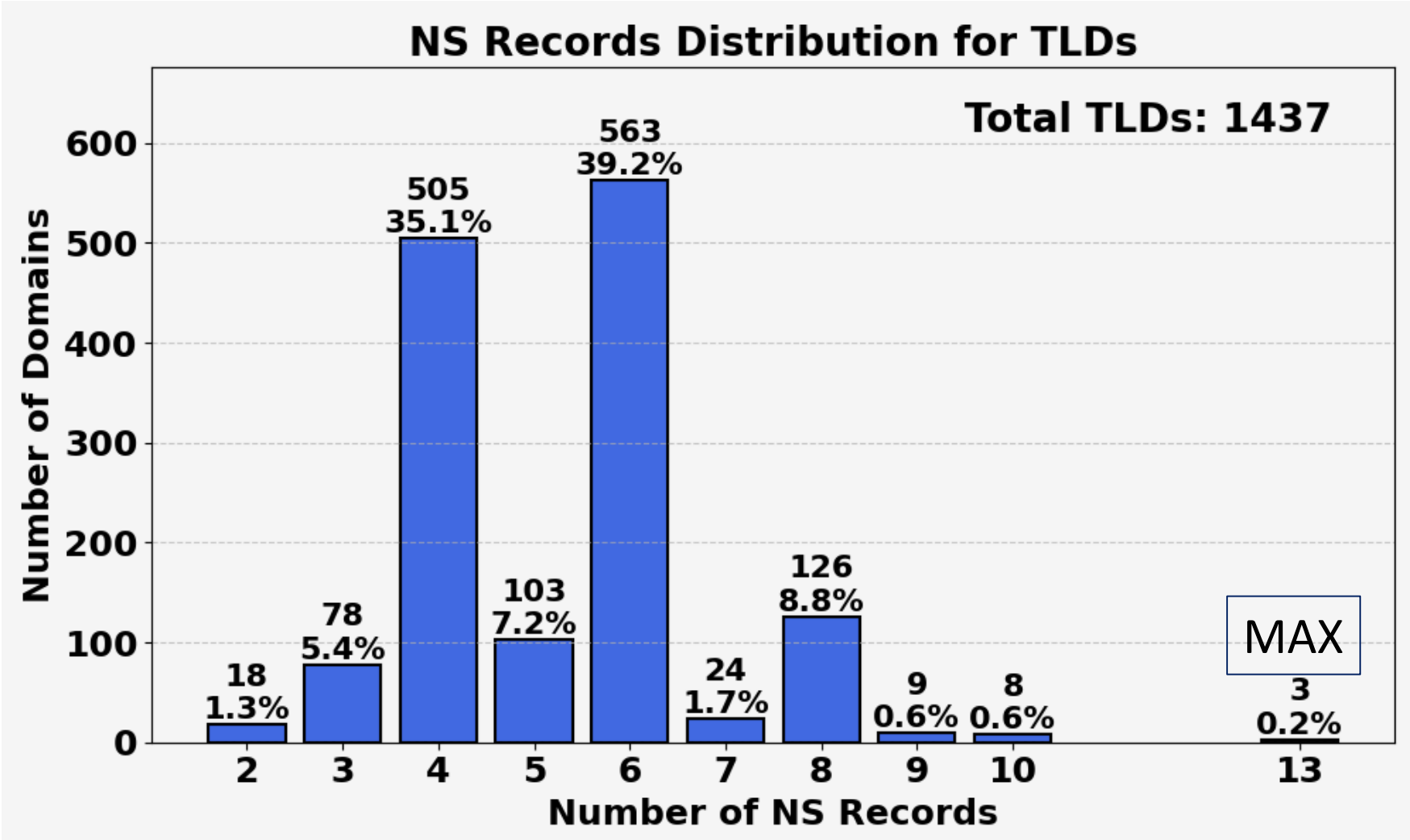
Measured 17/10/2024



What should MAX #RR be?

- 100? 20? 13? 10?

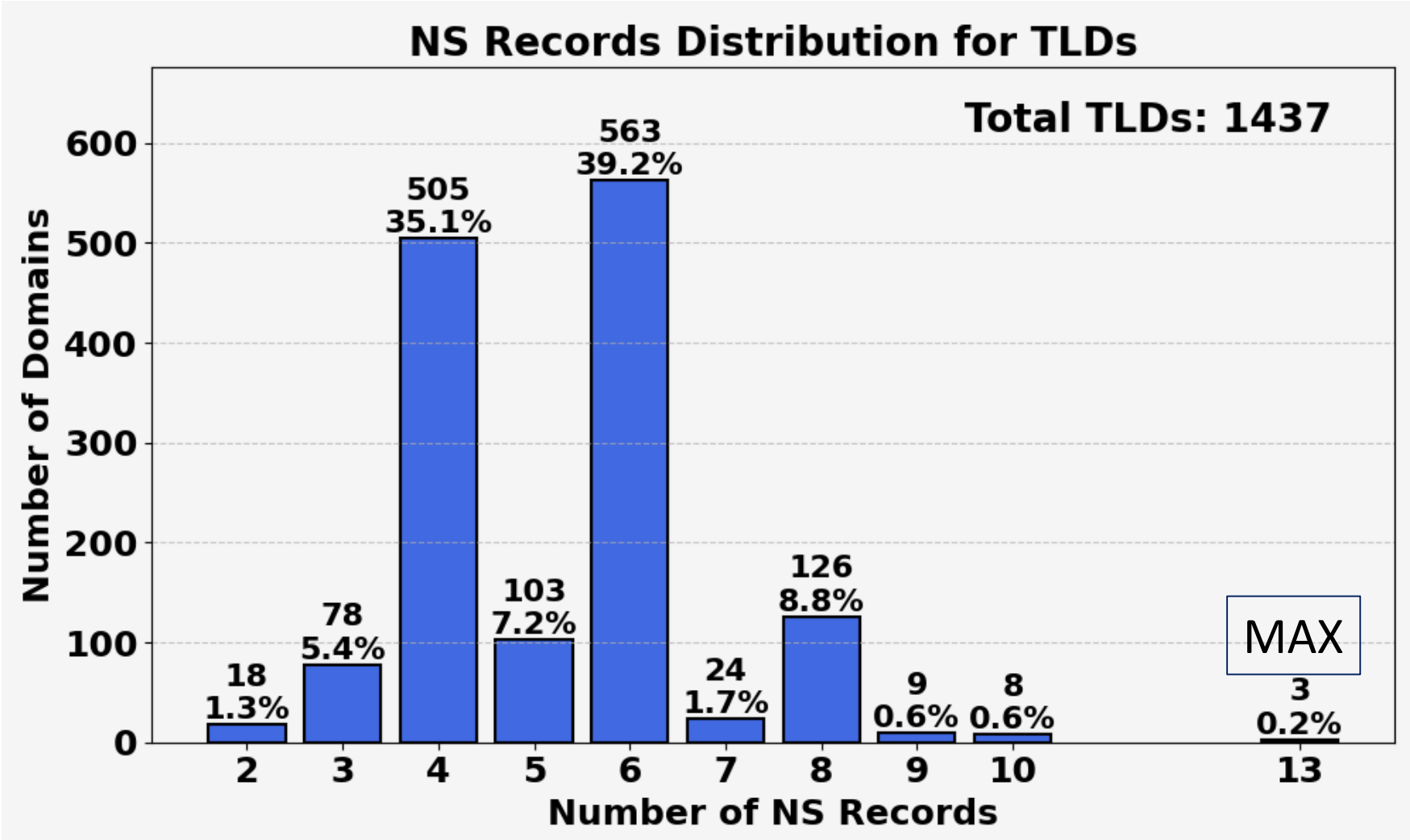
Measured 17/10/2024



What should MAX #RR be?

- 100? 20? 13? 10?

Measured 17/10/2024

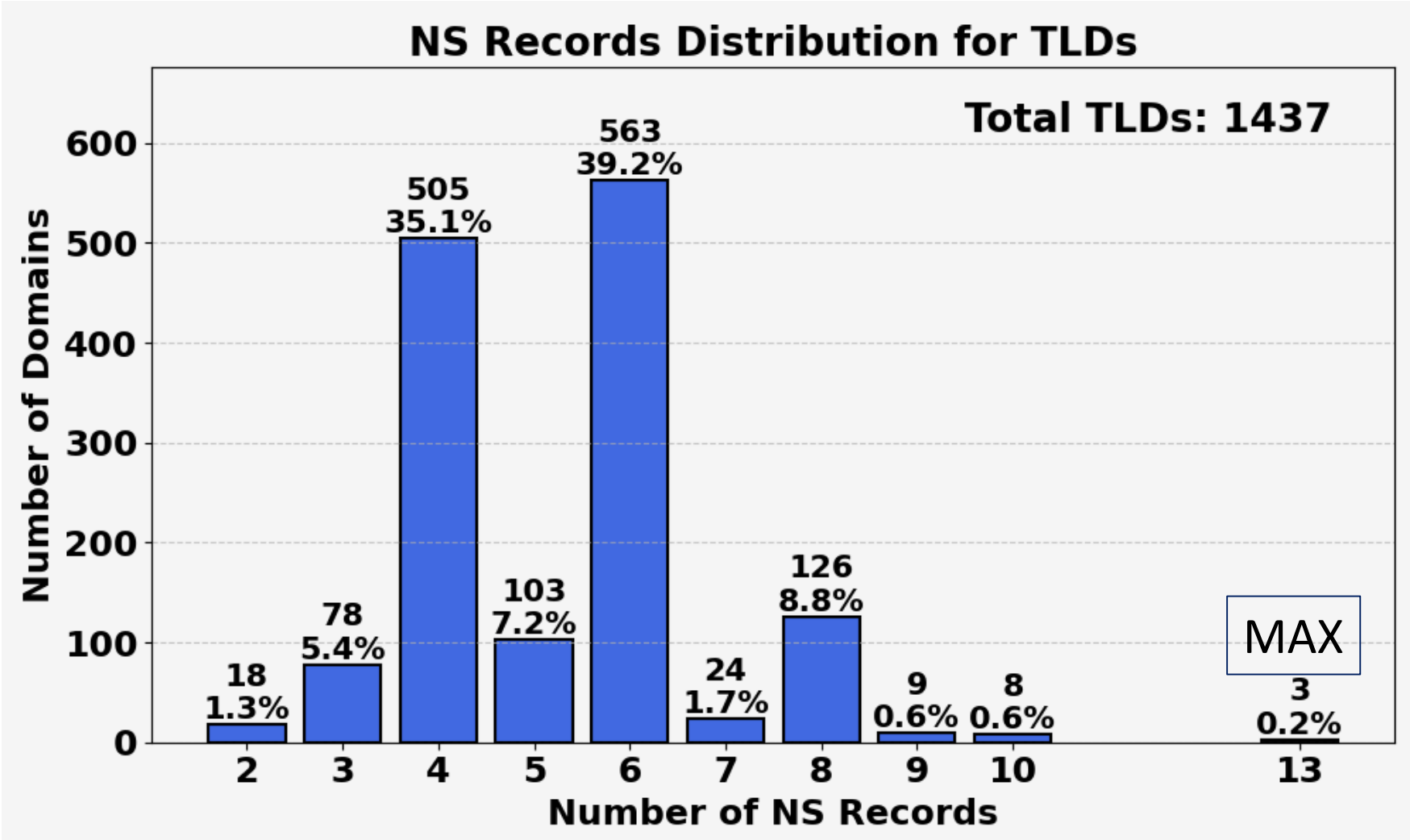


What should MAX #RR be?

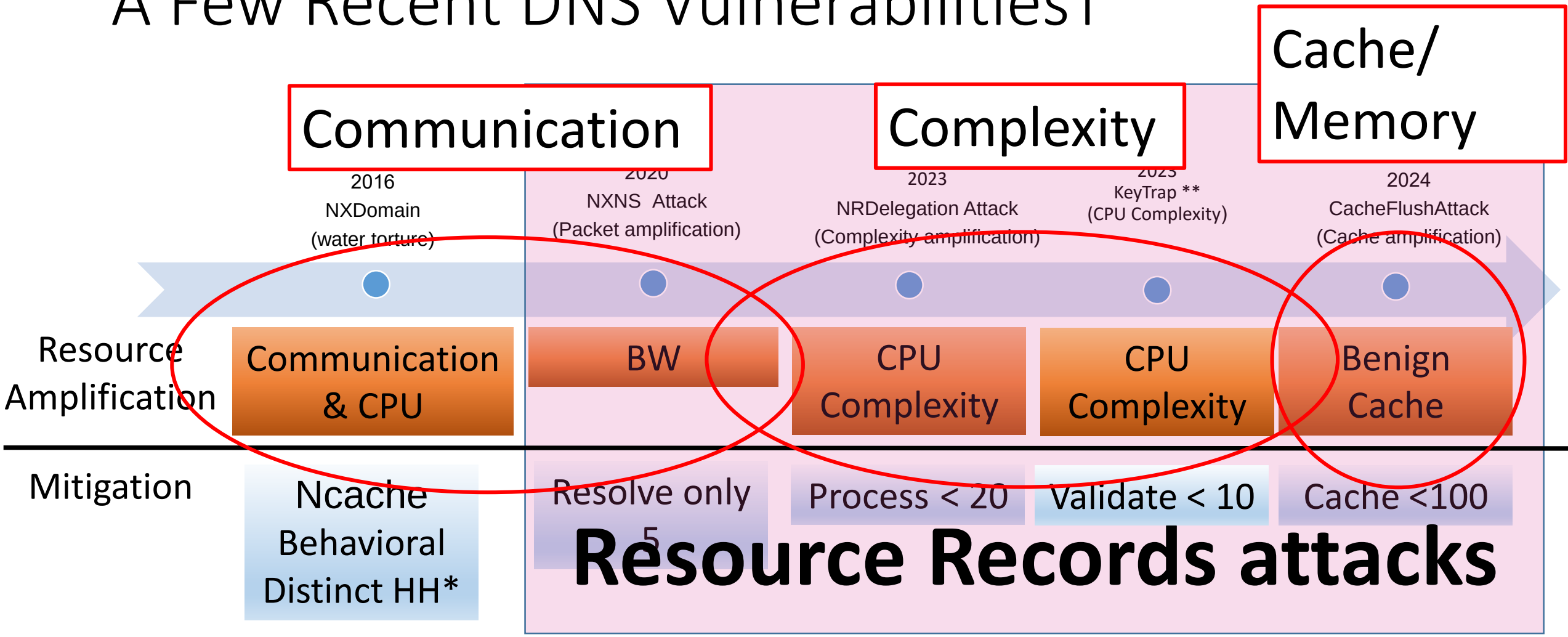
- 100? 20? 13? 10?

For all kinds & shape of RR

Measured 17/10/2024



A Few Recent DNS Vulnerabilities†*⁑



*[*Landau Feibish, A, Bremler-Barr, Cohen, Shagam: Mitigating NXDomain DDoS attacks by distinct heavy hitters sketches*]

[*Heftrig Schulmann Vogel Waidner* **KeyTrap Denial-of-Service Algorithmic Complexity Attacks on DNS]

⁑ unlike [Xiang Li: **DNSBomb**]

⁂ Building Blocks [Duan, Bearzi, Vieli, Tanir, Xu, Basin, Perrig, Liu, and Tellenbach: **Systemization of DNS Self-Amplification:**]

Conclusions & Questions

Conclusions & Questions

- Thorough analysis of Resolver complexities (CS analysis)

Conclusions & Questions

- Thorough analysis of Resolver complexities (CS analysis)
- QA ? Stress testing lab?

Conclusions & Questions

- Thorough analysis of Resolver complexities (CS analysis)
- QA ? Stress testing lab?
- Formal automatic verification methods

Conclusions & Questions

- Thorough analysis of Resolver complexities (CS analysis)
- QA ? Stress testing lab?
- Formal automatic verification methods
- Software Eng: Unbounded arrays, buffer overflows, Large Const

Thank you

Questions?