

Recent DNS DDoS Attacks: Academic Perspective

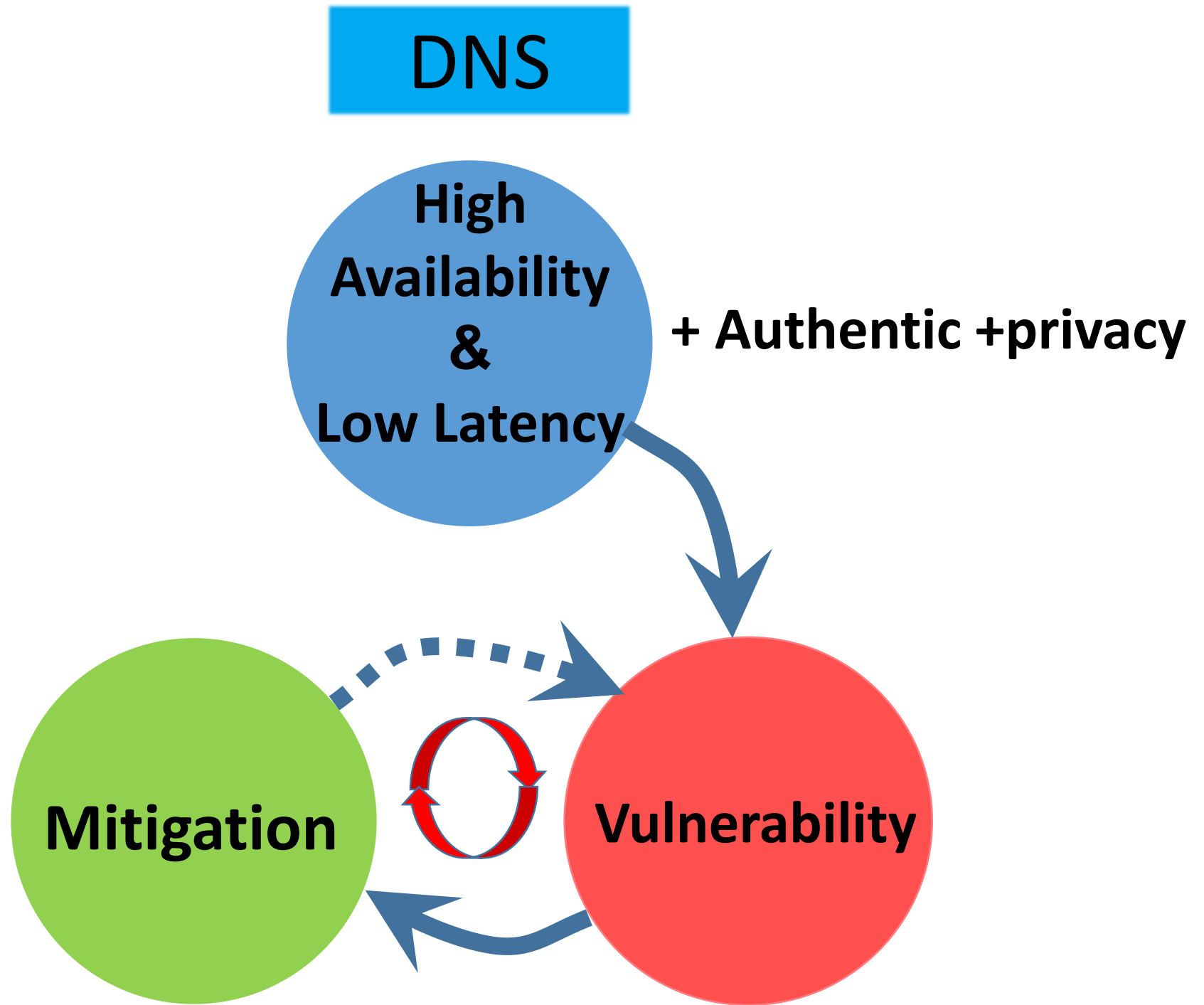
DNS OARC 43 -- 10/26/24

Yehuda Afek and Anat Bremler-Barr

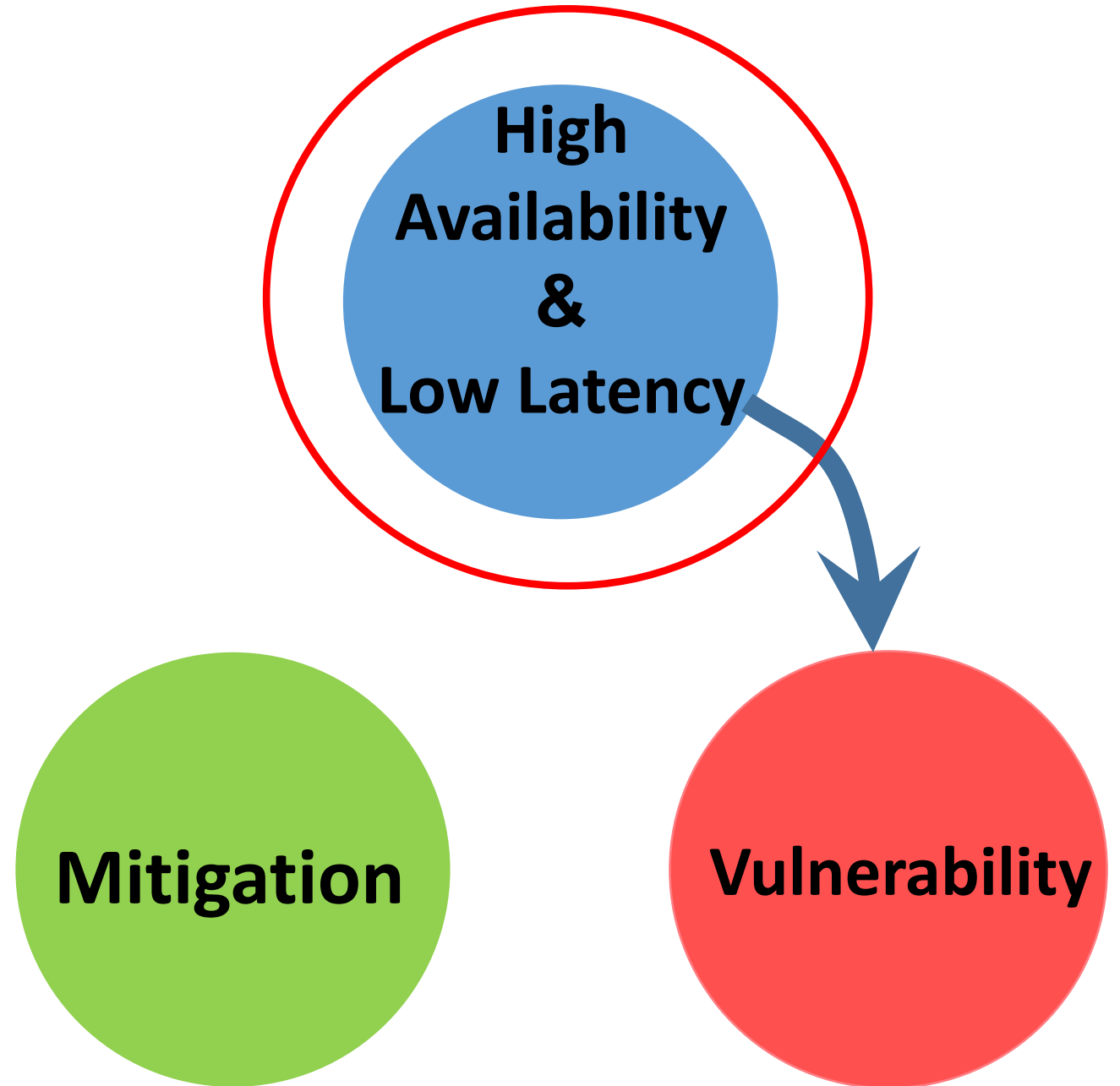
Tel-Aviv University



Outline



cycle 1 example



Referral Response & Glue Records

A request cs.ucla.edu

ucla.edu NS ns1.ucla.net
ucla.edu NS ns2.ucla.net
ucla.edu NS ns3.ucla.net
ucla.edu NS ns4.ucla.net
no glue records

.edu TLD

TLDs

.net

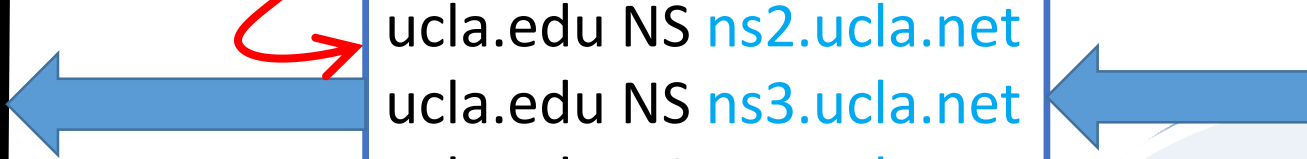
ns1.ucla.net ns2

ns3.ucla.net ns4

Recursive
Resolver

cs.ucla.edu ??

Empty cache



Referral Response & Glue Records

Which referred NS is the **Quickest** ?

A request cs.ucla.edu

ucla.edu NS ns1.ucla.net
ucla.edu NS ns2.ucla.net
ucla.edu NS ns3.ucla.net
ucla.edu NS ns4.ucla.net
no glue records

.edu TLD

'A' request ns1.ucla.net
'A' request ns2.ucla.net
'A' request ns3.ucla.net
'A' request ns4.ucla.net

cs.ucla.edu ??

Empty cache

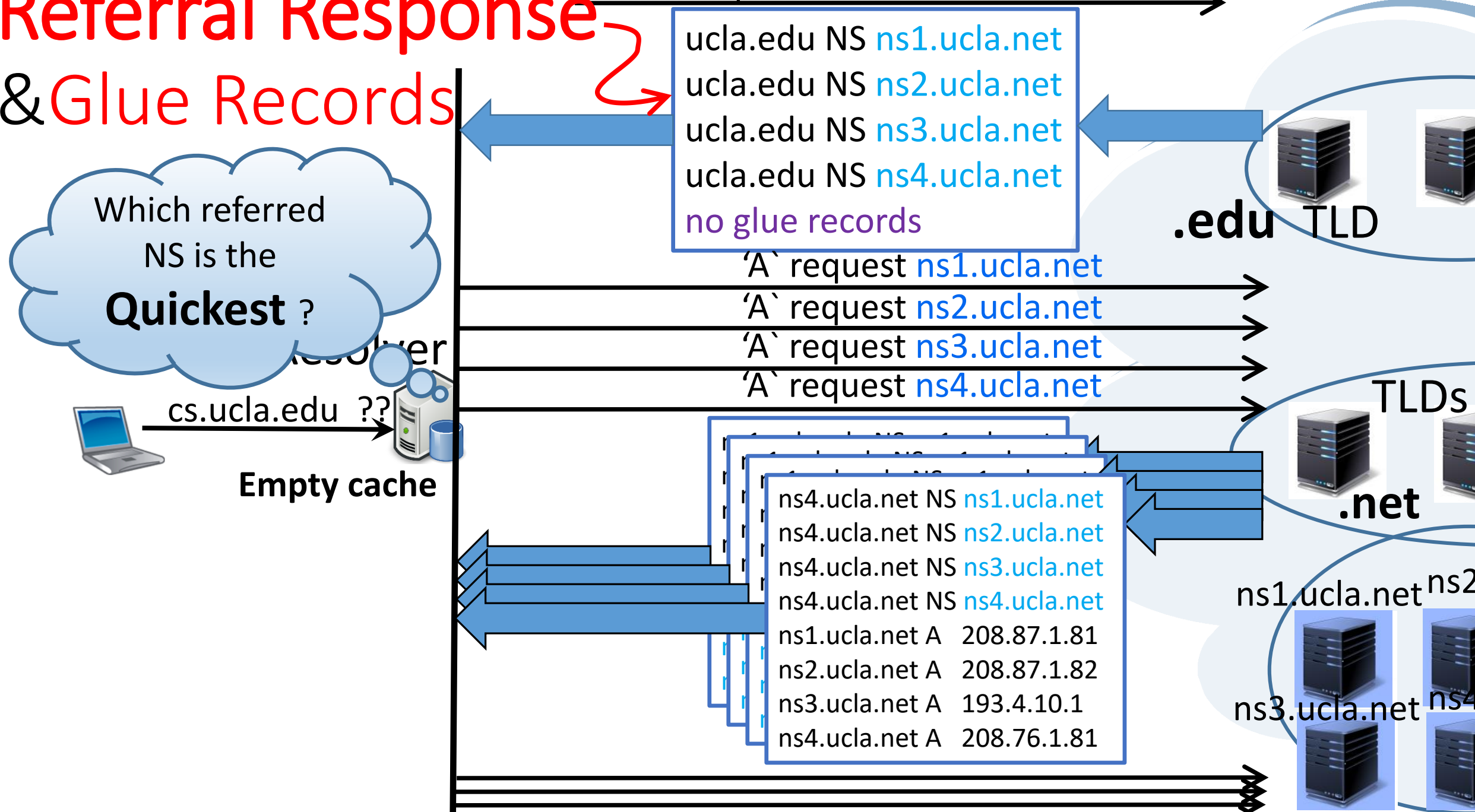
TLDs

.net

ns4.ucla.net NS ns1.ucla.net
ns4.ucla.net NS ns2.ucla.net
ns4.ucla.net NS ns3.ucla.net
ns4.ucla.net NS ns4.ucla.net
ns1.ucla.net A 208.87.1.81
ns2.ucla.net A 208.87.1.82
ns3.ucla.net A 193.4.10.1
ns4.ucla.net A 208.76.1.81

ns1.ucla.net ns2

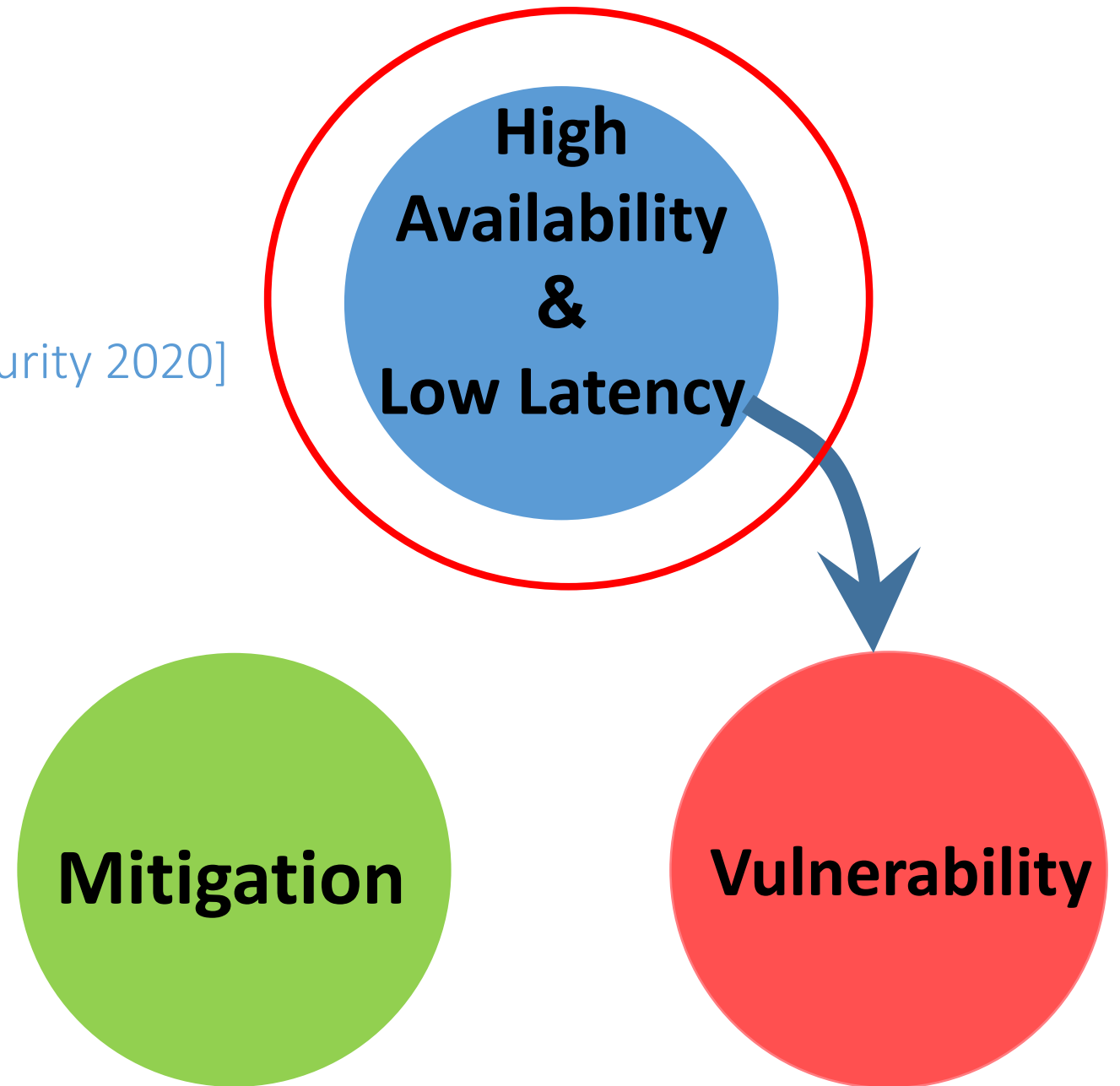
ns3.ucla.net ns4



First cycle example

NXNS Attack

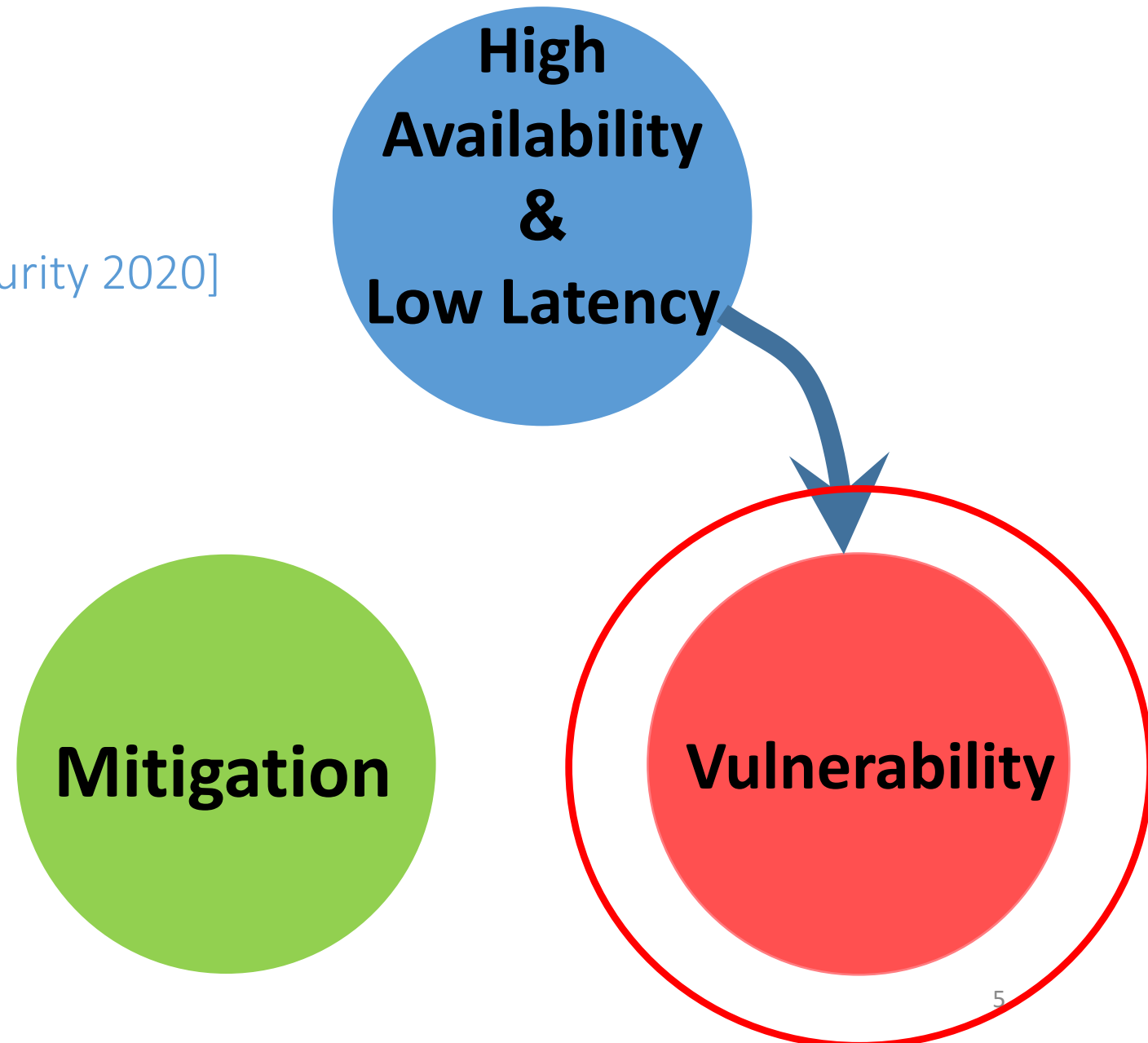
[A, Anat Bremler-Barr, Lior Shafir. [Usenix Security 2020](#)]



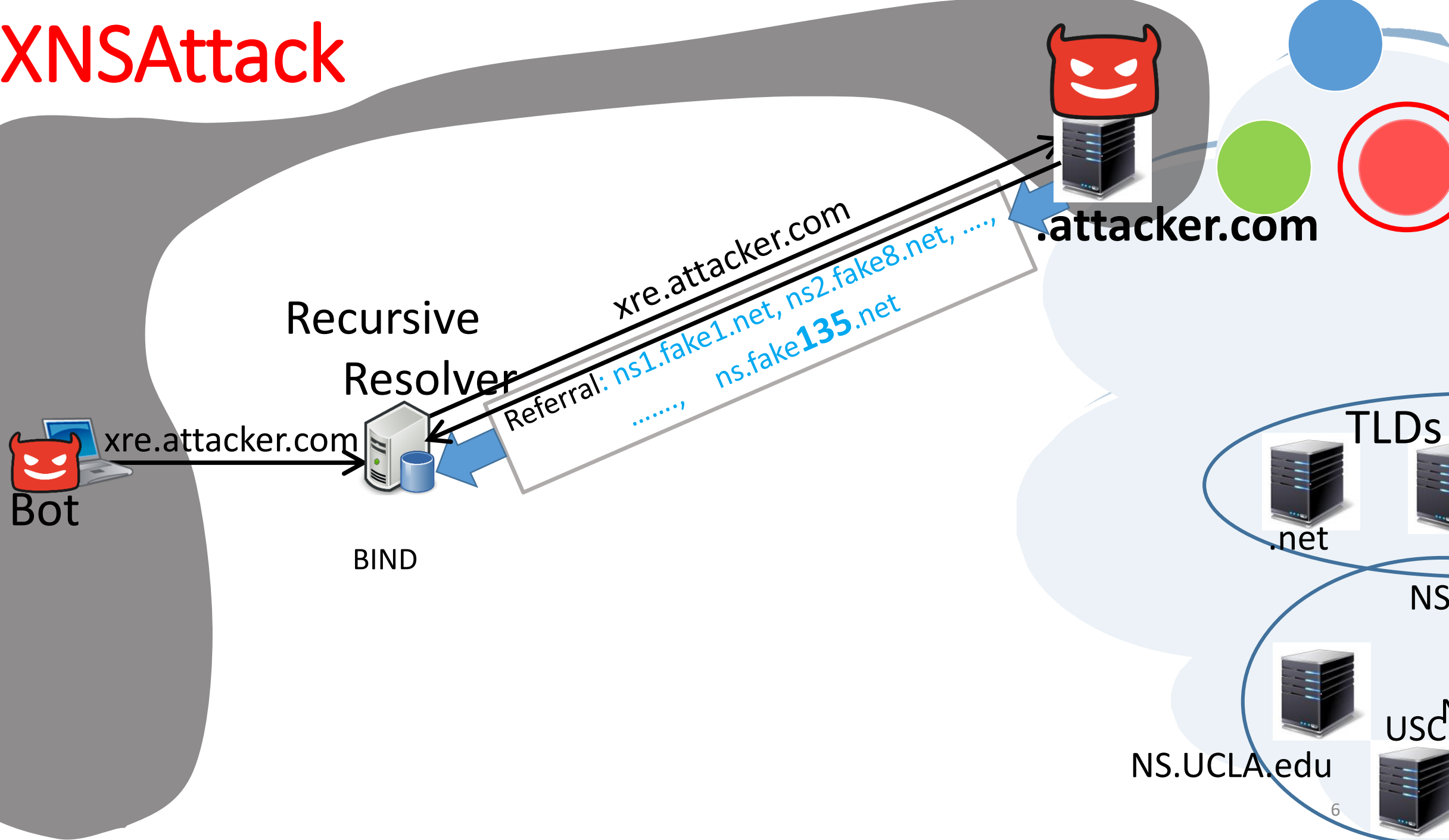
First cycle example

NXNS Attack

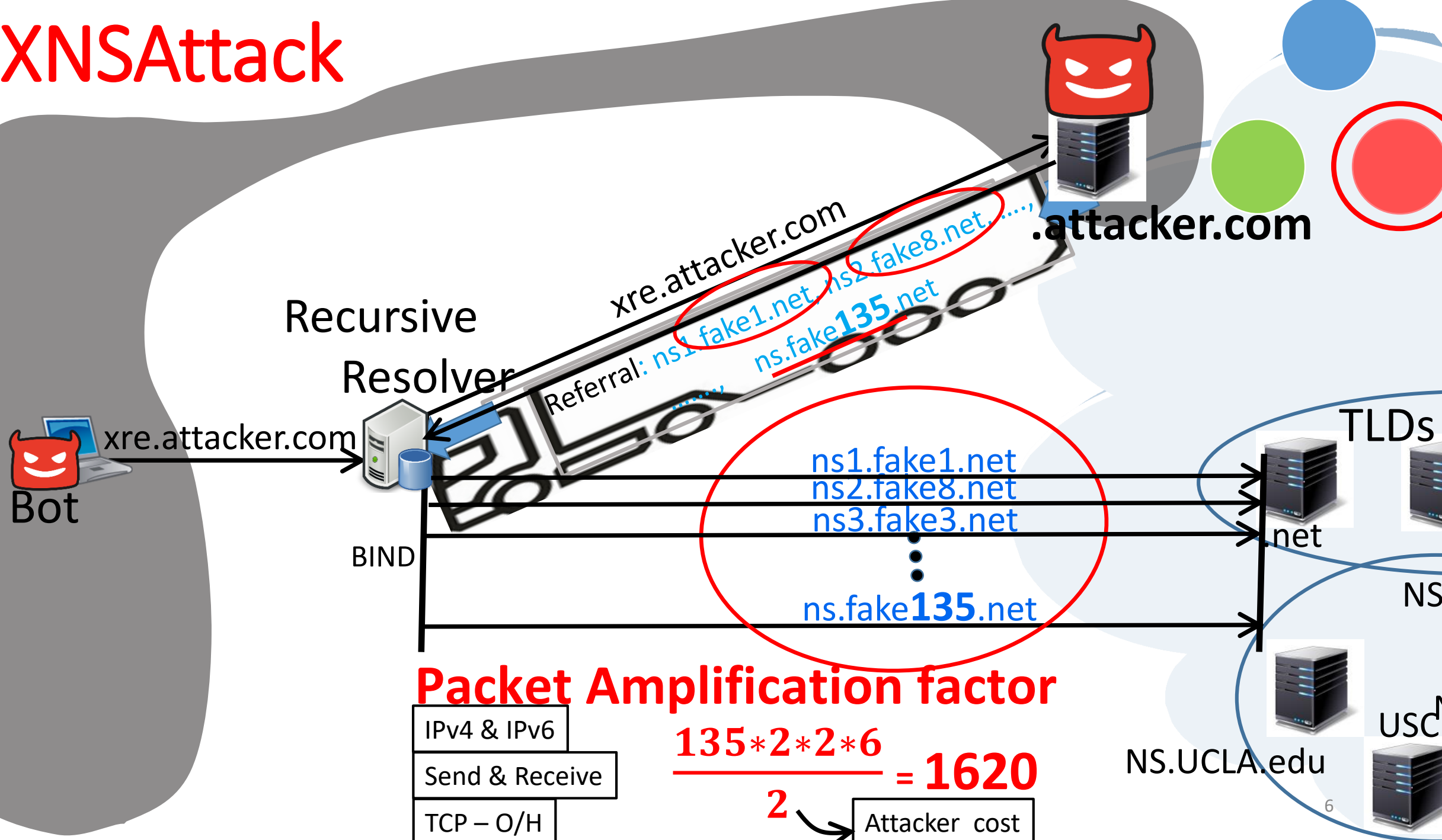
[A, Anat Bremler-Barr, Lior Shafir. [Usenix Security 2020](#)]



NXNSAttack



NXNSAttack

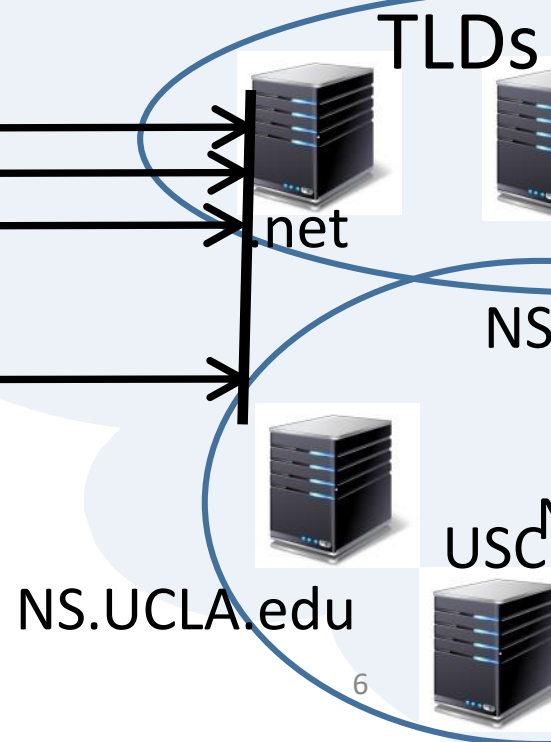


Packet Amplification factor

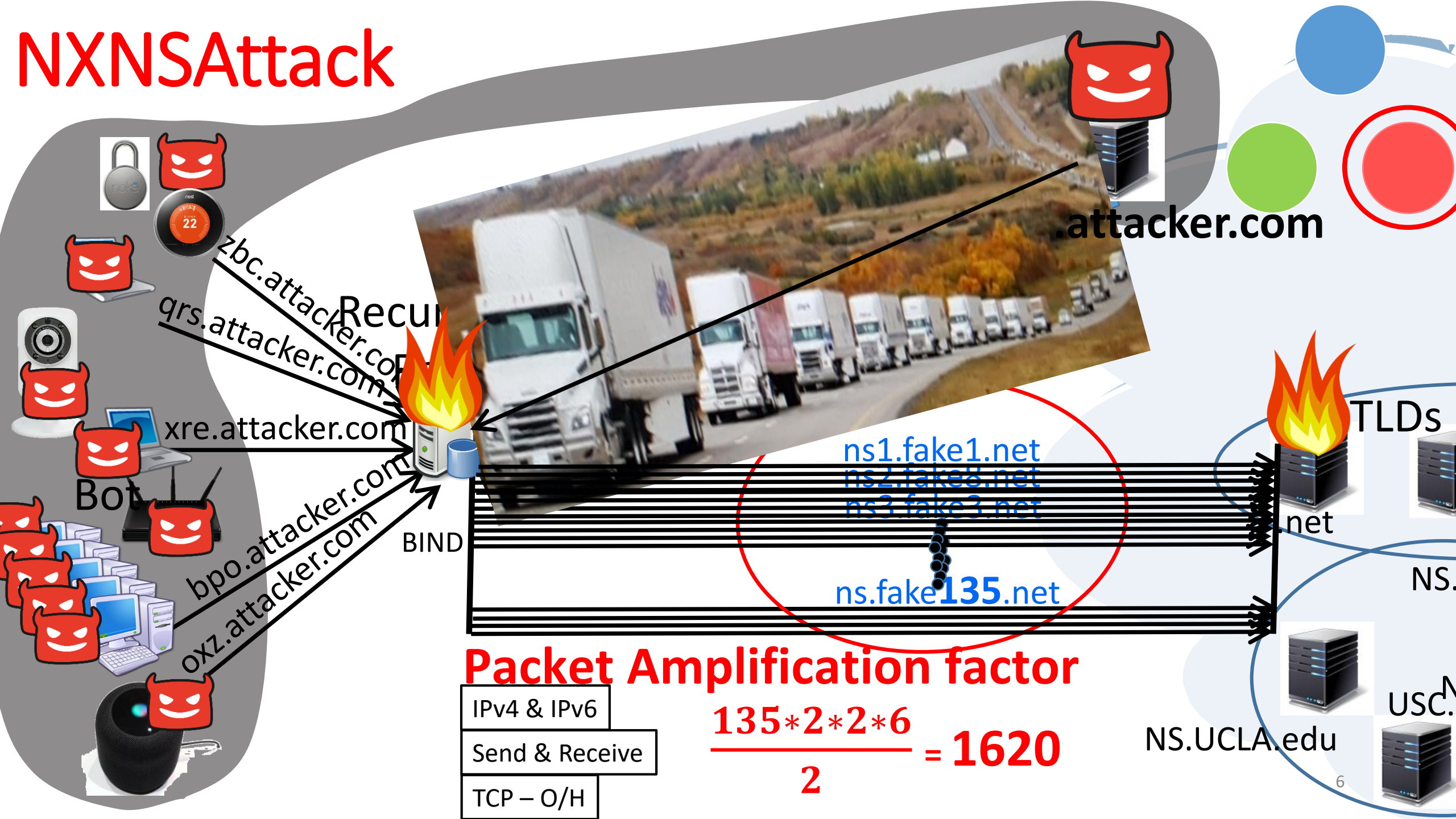
- IPv4 & IPv6
- Send & Receive
- TCP - O/H

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

Attacker cost



NXNSAttack



zbc.attacker.com
qrs.attacker.com
xre.attacker.com
bpo.attacker.com
oxz.attacker.com

BIND

ns1.fake1.net
ns2.fake2.net
ns3.fake3.net
ns.fake135.net

TLDs

net

NS.

USC.

NS.UCLA.edu

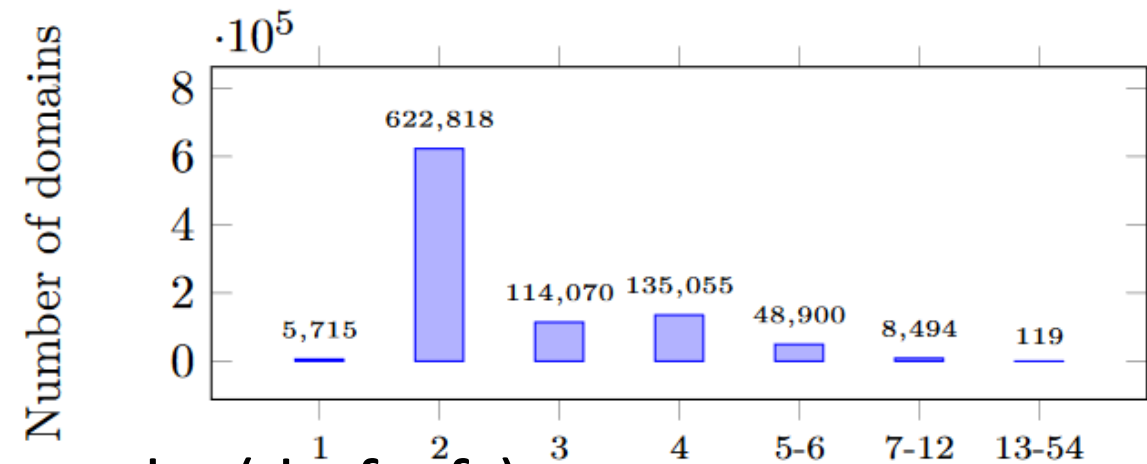
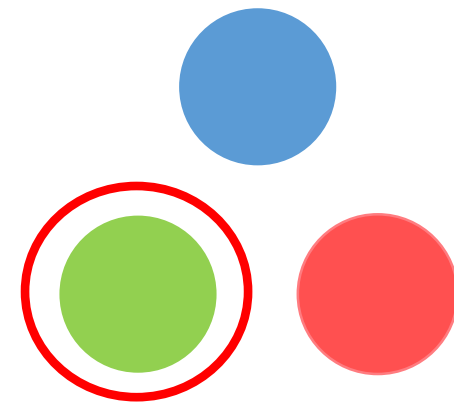
Packet Amplification factor

- 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

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

- Detect NX NS replies (RFC 4861)

- DNSSEC – NSEC (Petr Špaček)

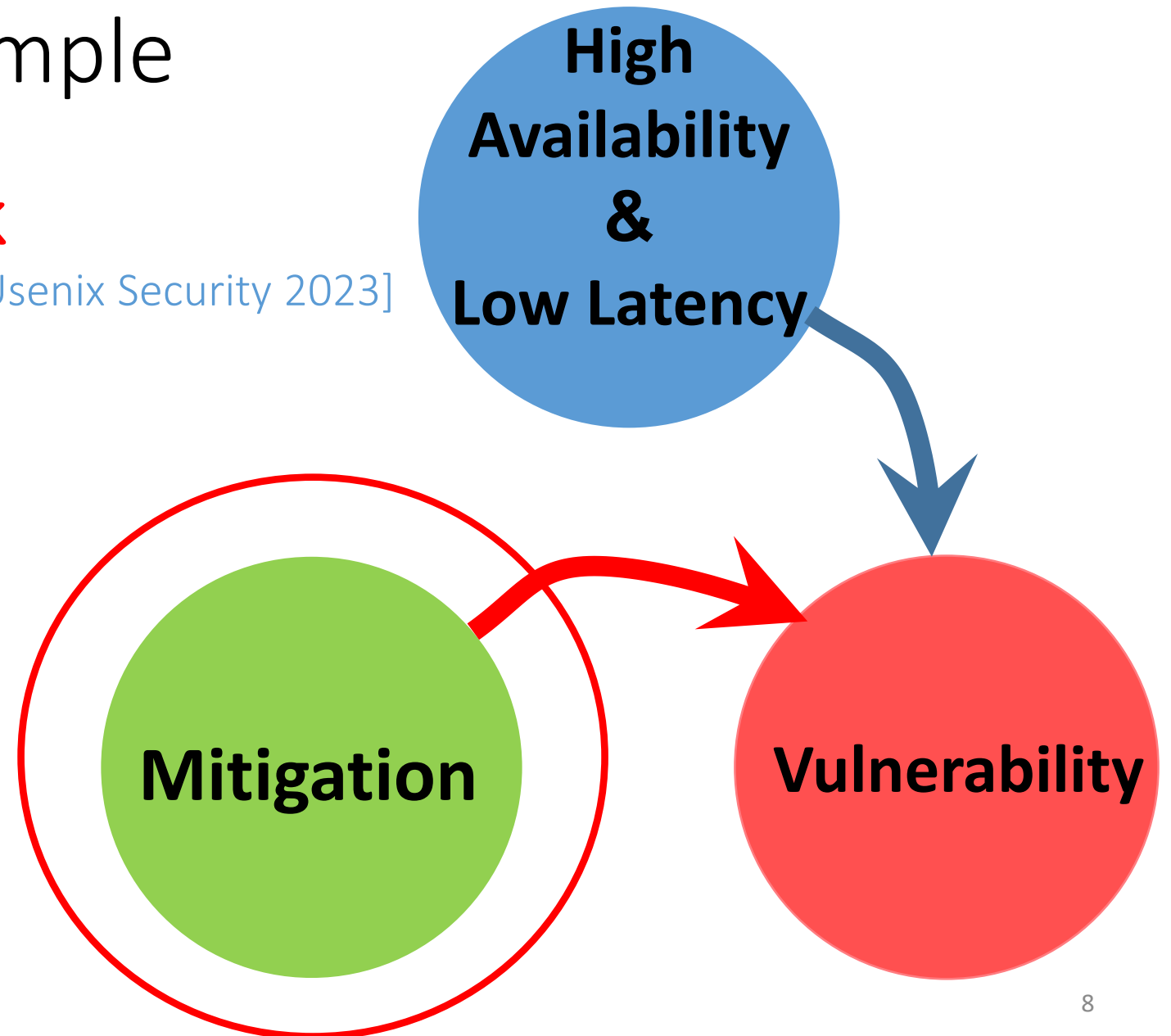
- Going only downwards in the DNS hierarchy (draft rfc)

Resolve only 5

Second cycle example

NRDelegation Attack

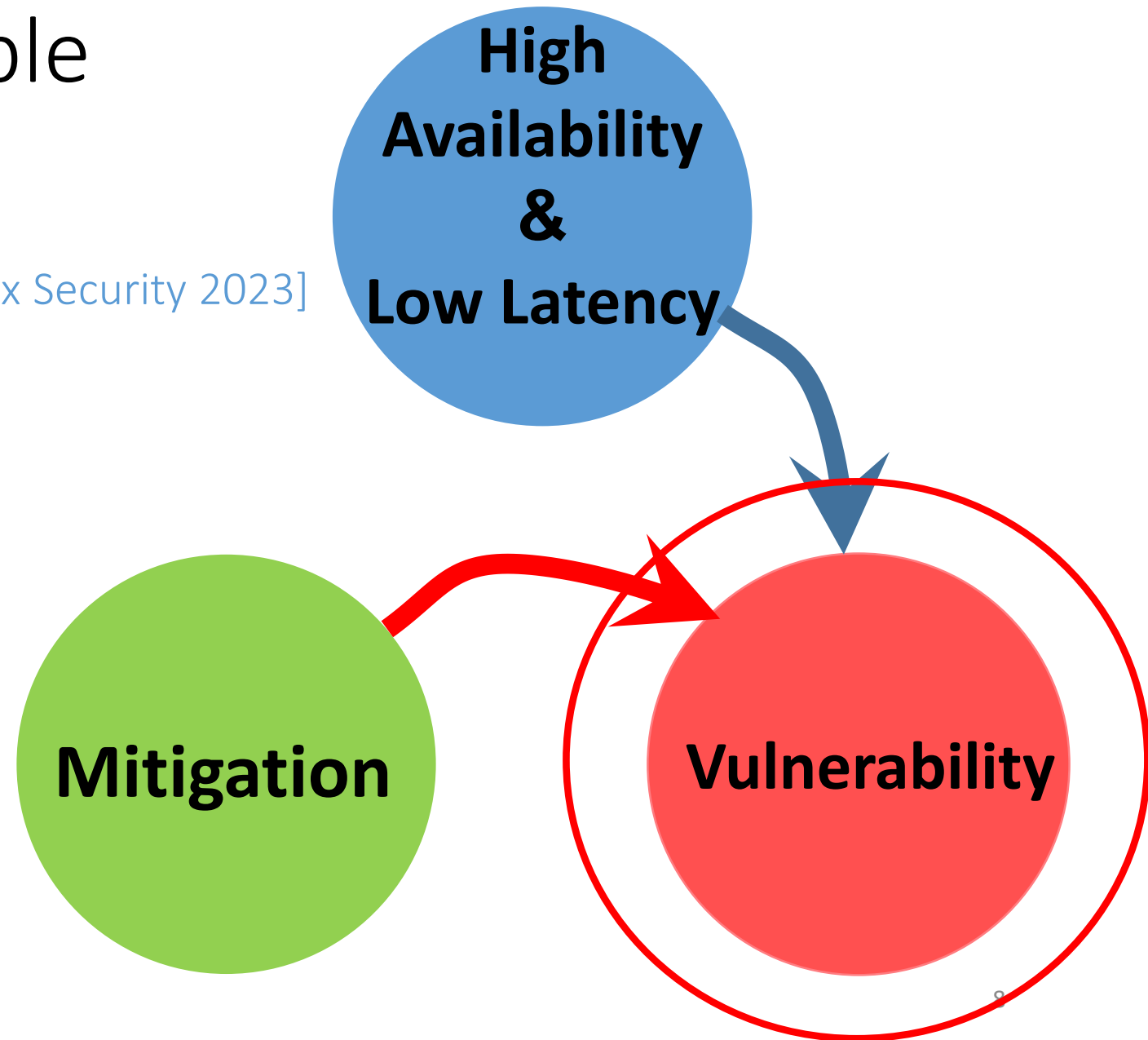
[A, Anat Bremler-Barr, Shani Stajnsrod [Usenix Security 2023](#)]



Second cycle example

NRDelegation Attack

[A, Anat Bremler-Barr, Shani Stajnsrod [Usenix Security 2023](#)]



NRDelegation Attack:

Complexity attack on the DNS Resolver

NRDelegation Attack:

Complexity attack on the DNS Resolver

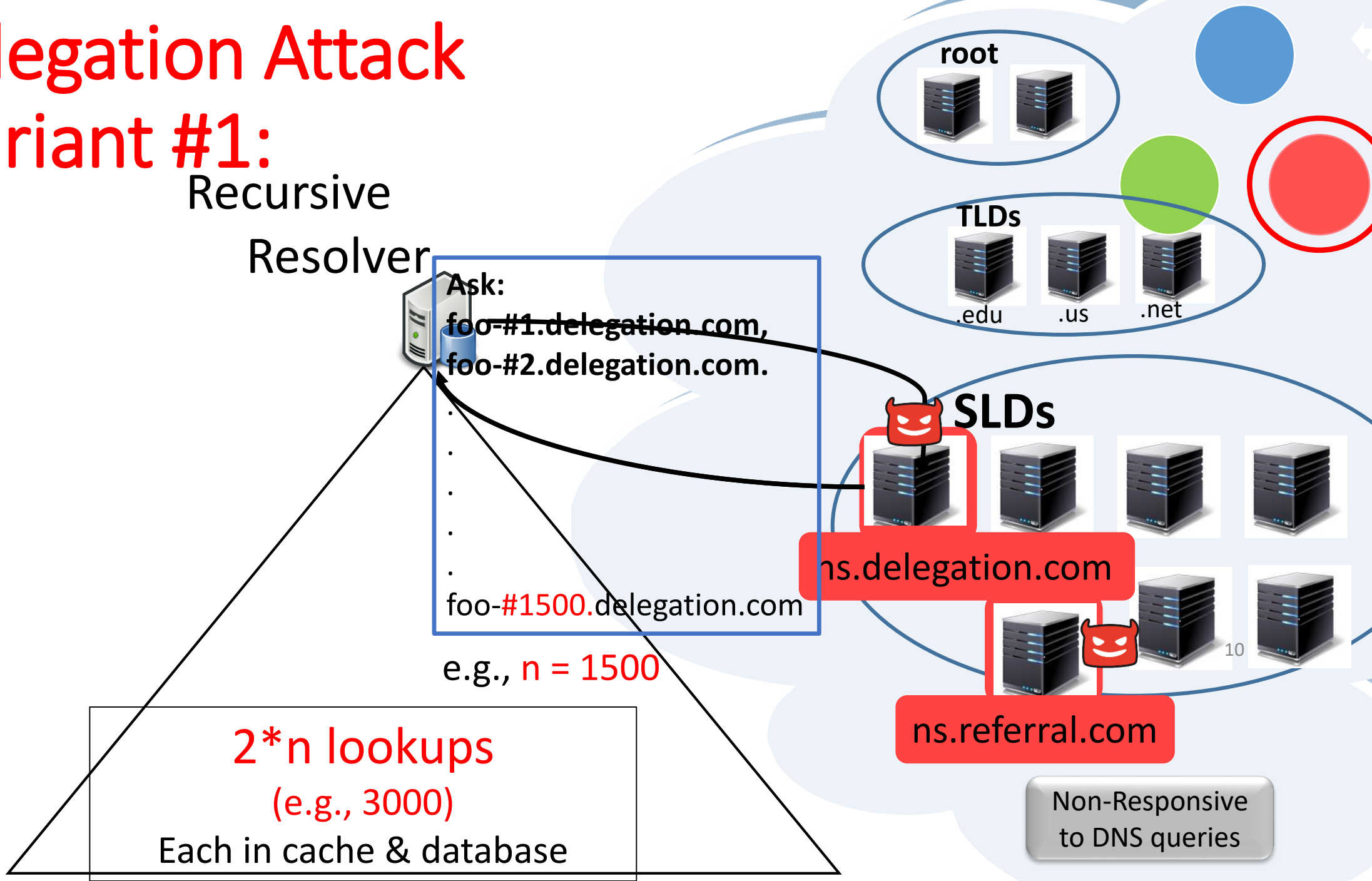
Root cause – a single malicious query → High CPU load.

Enabled by NXNS mitigation method !

NRDelegation Attack

Variant #1:

Recursive
Resolver



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

e.g., `n = 1500`

`2*n lookups`
(e.g., 3000)

Each in cache & database



`ns.delegation.com`

`ns.referral.com`

Non-Responsive
to DNS queries

NRDelegation Attack

Variant #1:

Recursive

Resolver

$2 * n$ lookups

(e.g., 3000)

Each in cache & database



1. $2 * n$ lookups (=3000)

Ask:

foo-#1.delegation.com,
foo-#2.delegation.com.
.
.
.
.
.
.
foo-#1500.delegation.com

root



TLDs



.edu

.us

.net

SLDs



ns.delegation.com



10

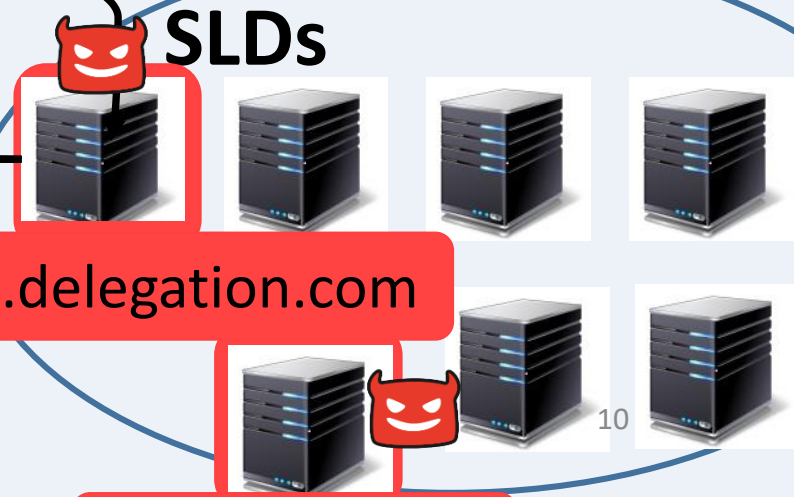
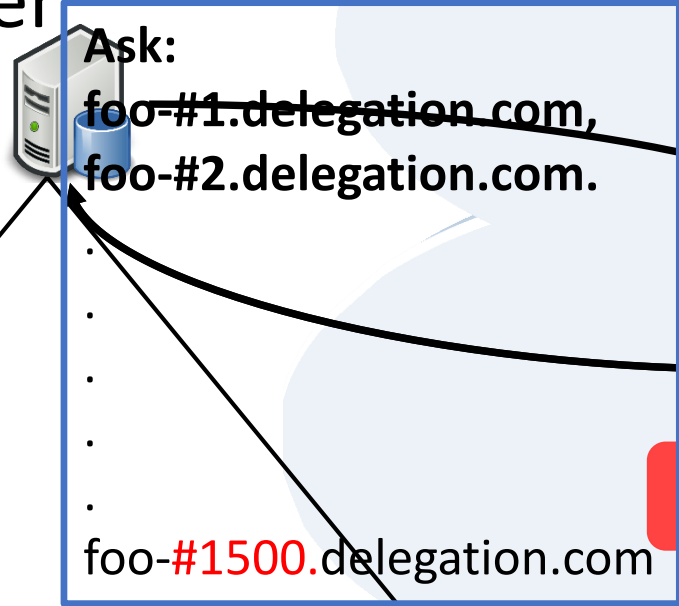
ns.referral.com


Non-Responsive
to DNS queries

NRDelegation Attack

Variant #1:

Recursive
Resolver




**NXNS
Mitigation**

1. $2 * n$ lookups (=3000)
 $n > \text{referral-limit} = 5$
Resolve only 5 NS names,
turn on "No Fetch" flag

Non-Responsive
to DNS queries

NRDelegation Attack

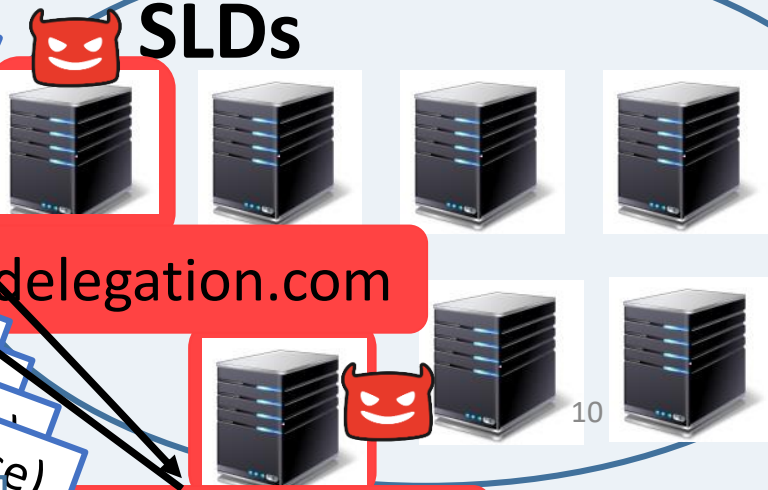
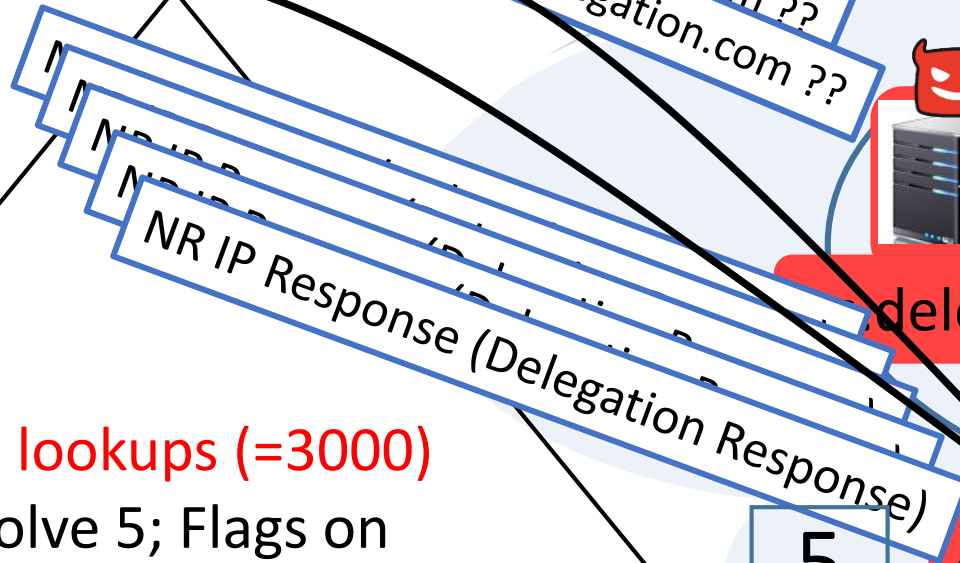
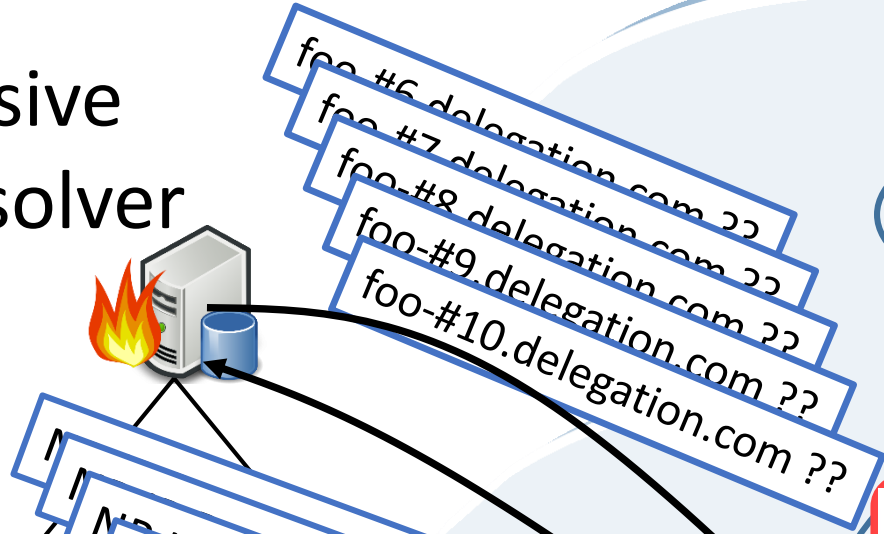
Variant #1:

Recursive
Resolver



**NXNS
Mitigation**

- 1. $2 * n$ lookups (=3000)
- 2. Resolve 5; Flags on



Non-Responsive
to DNS queries

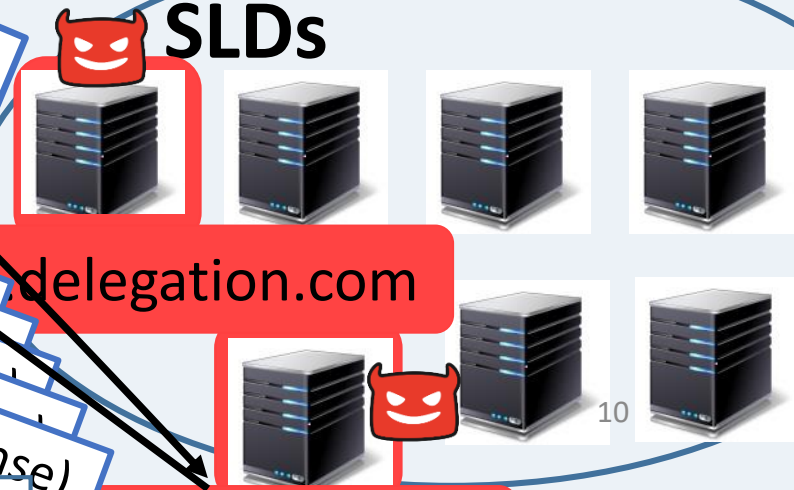
NRDelegation Attack

Variant #1:

Recursive
Resolver



foo-#6.delegation.com ??
foo-#7.delegation.com ??
foo-#8.delegation.com ??
foo-#9.delegation.com ??
foo-#10.delegation.com ??



**NXNS
Mitigation**

1. $2 * n$ lookups (=3000)
2. Resolve 5; Flags on

NR IP Response (Delegation Response)

5

RESTART Event
Clear all flags
turn off "No-Fetch" flag

ns.referral.com

Non-Responsive
to DNS queries

NRDelegation Attack

Variant #1:

Recursive
Resolver

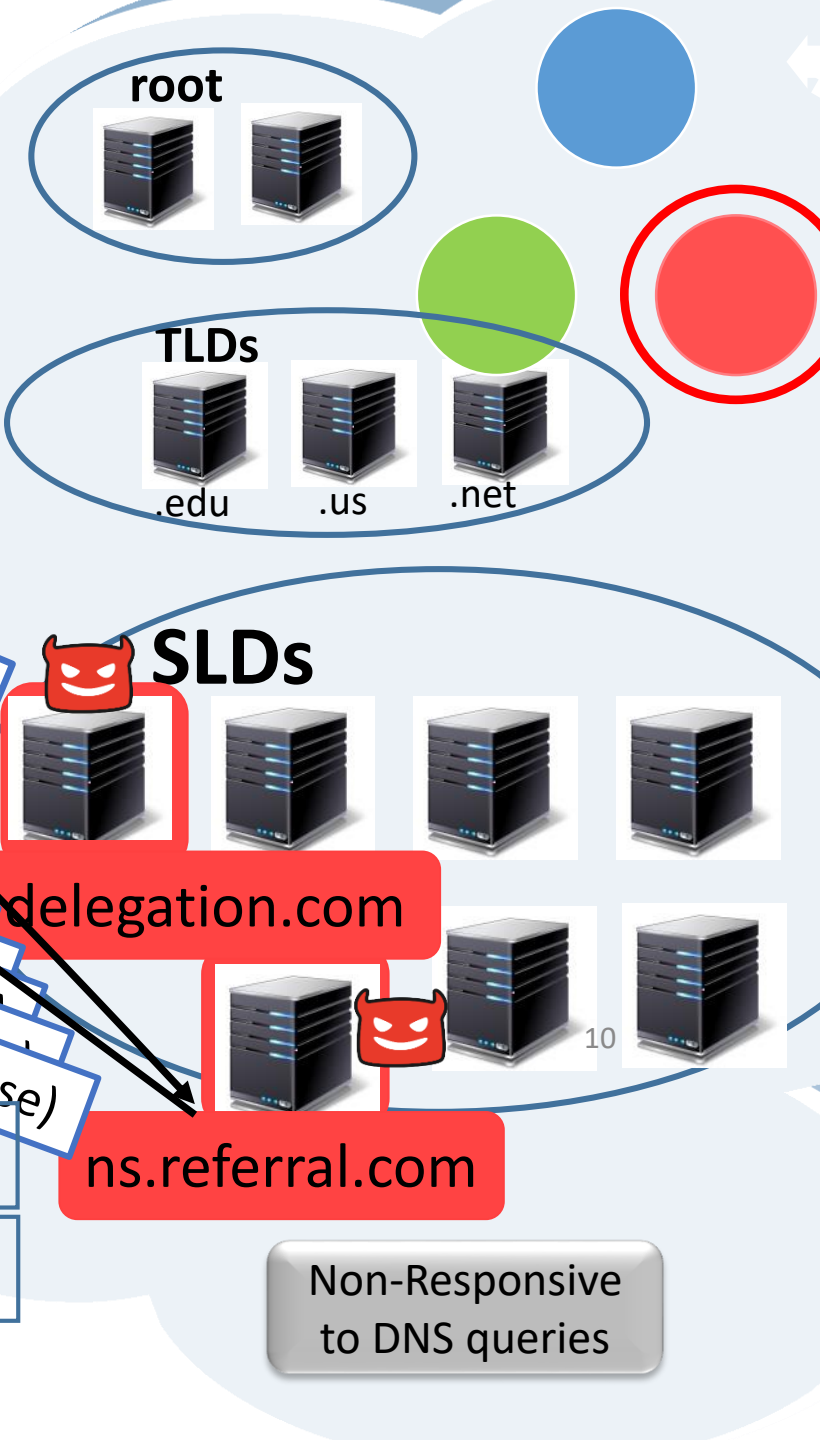
**NXNS
Mitigation**

1. $2 * n$ lookups (=3000)
2. Resolve 5; Flags on
3. **RESTART** Event; Flags OFF



foo-#6.delegation.com ??
foo-#7.delegation.com ??
foo-#8.delegation.com ??
foo-#9.delegation.com ??
foo-#10.delegation.com ??

NR IP Response (Delegation Response)



5
25

Non-Responsive
to DNS queries

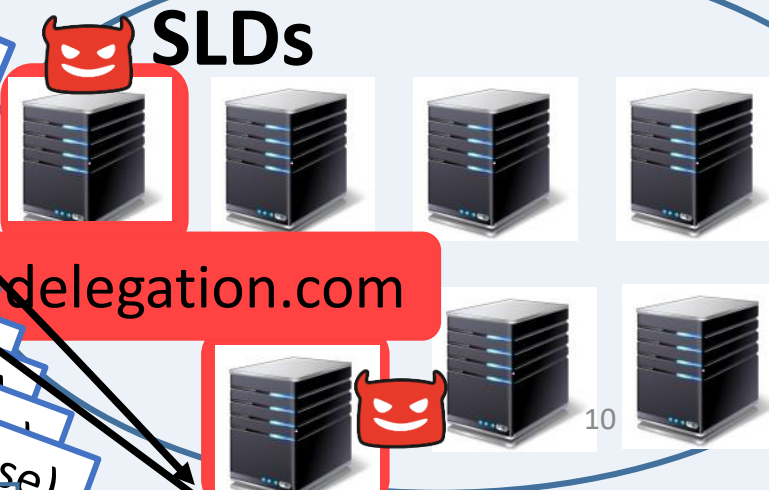
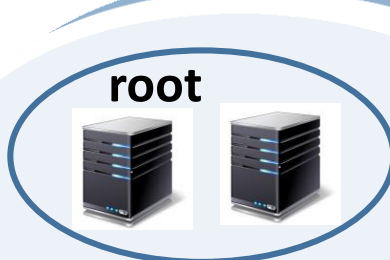
NRDelegation Attack

Variant #1:

Recursive Resolver



foo-#6.delegation.com ??
foo-#7.delegation.com ??
foo-#8.delegation.com ??
foo-#9.delegation.com ??
foo-#10.delegation.com ??



NR IP Response (Delegation Response)

1. $2 * n$ lookups (=3000)
2. Resolve 5; Flags on
3. **RESTART** Event; Flags OFF

5
25
125

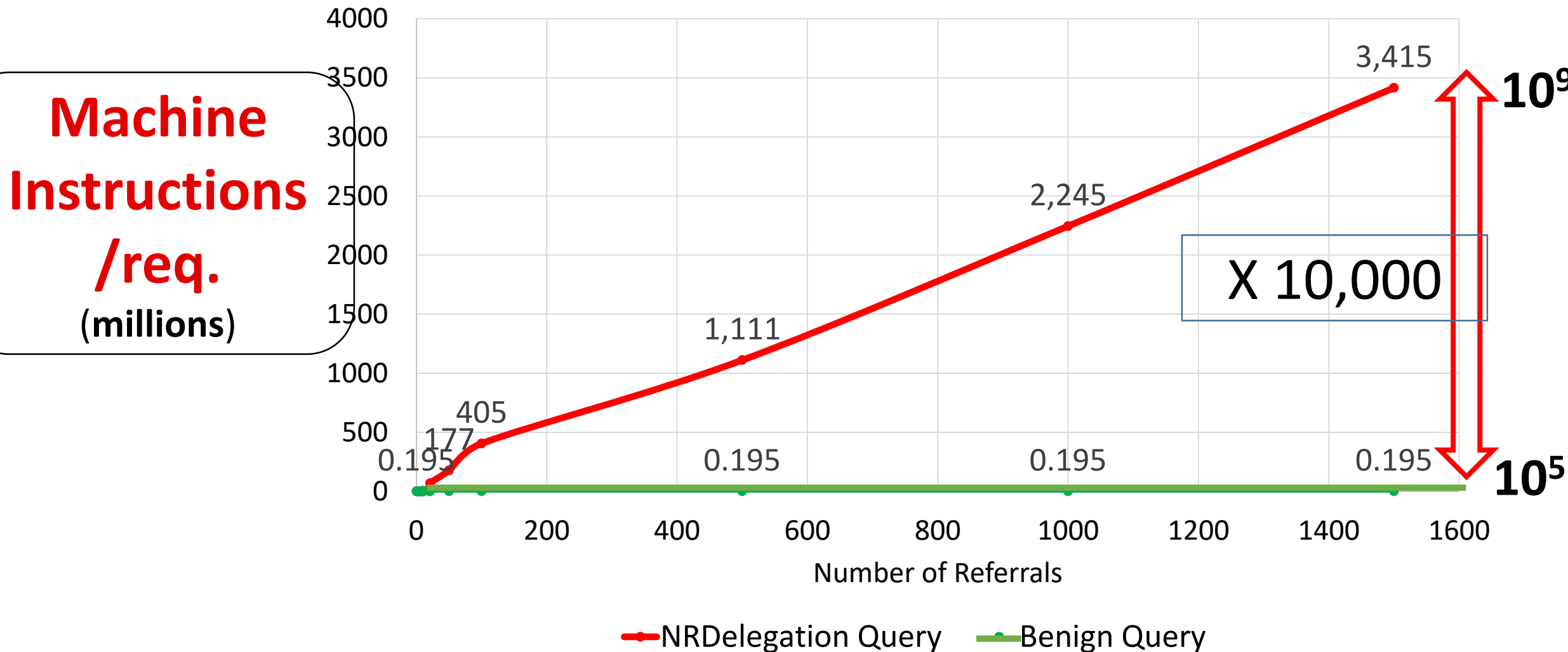
ns.referral.com

Non-Responsive to DNS queries

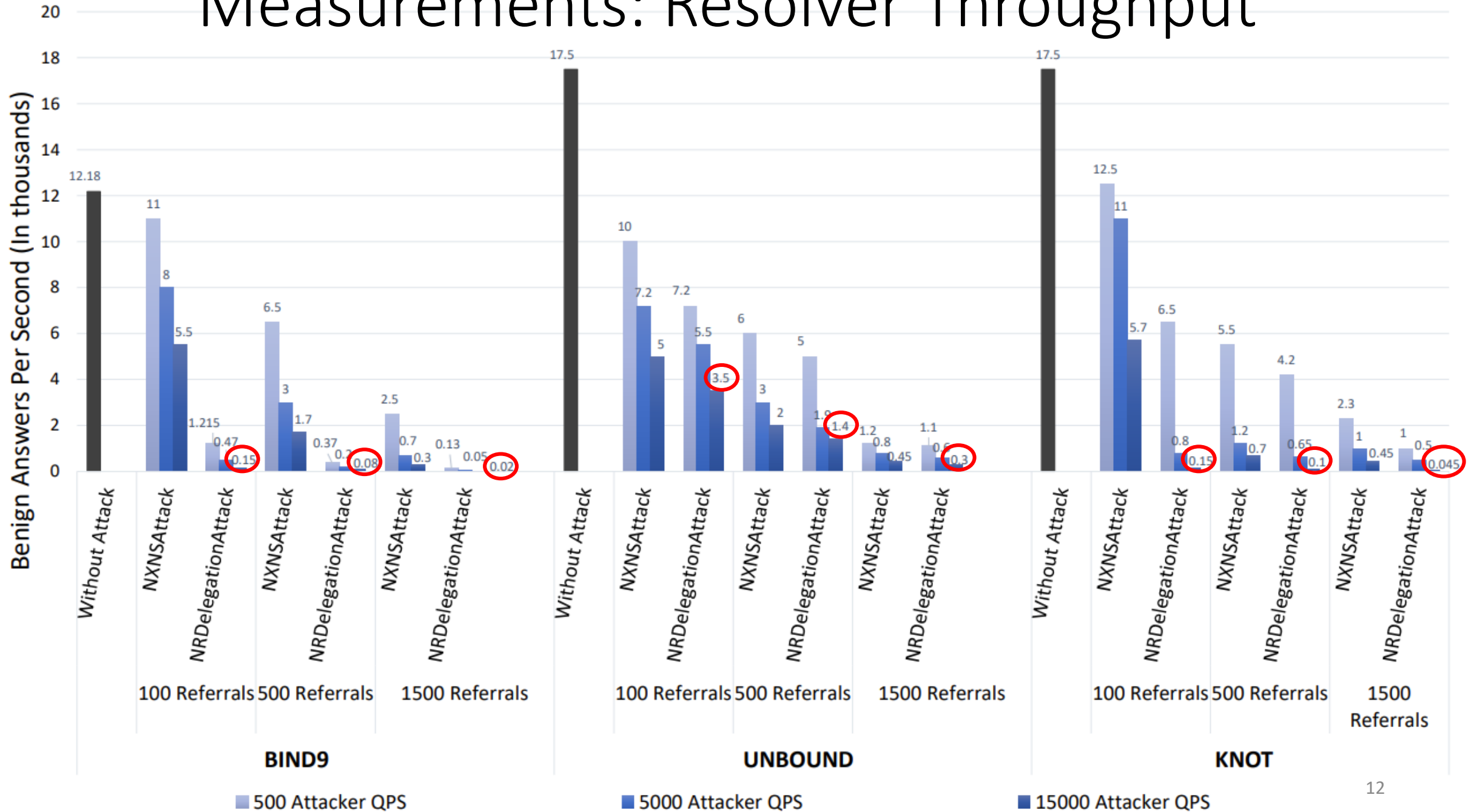


NRDelegation Complexity Amplification

Instructions On The Resolver Processor During NRDelegation Attack



Measurements: Resolver Throughput



Mitigation: NRDelegation Attack

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

Recursive
Resolver



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

20

TLDs



.com



.edu



.us



.net

SLDs

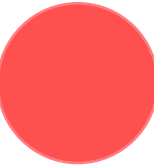
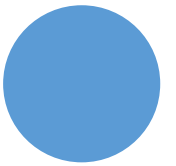
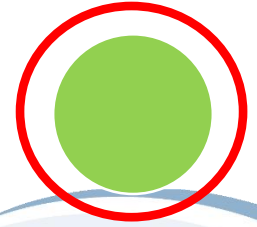


ns.referral.com



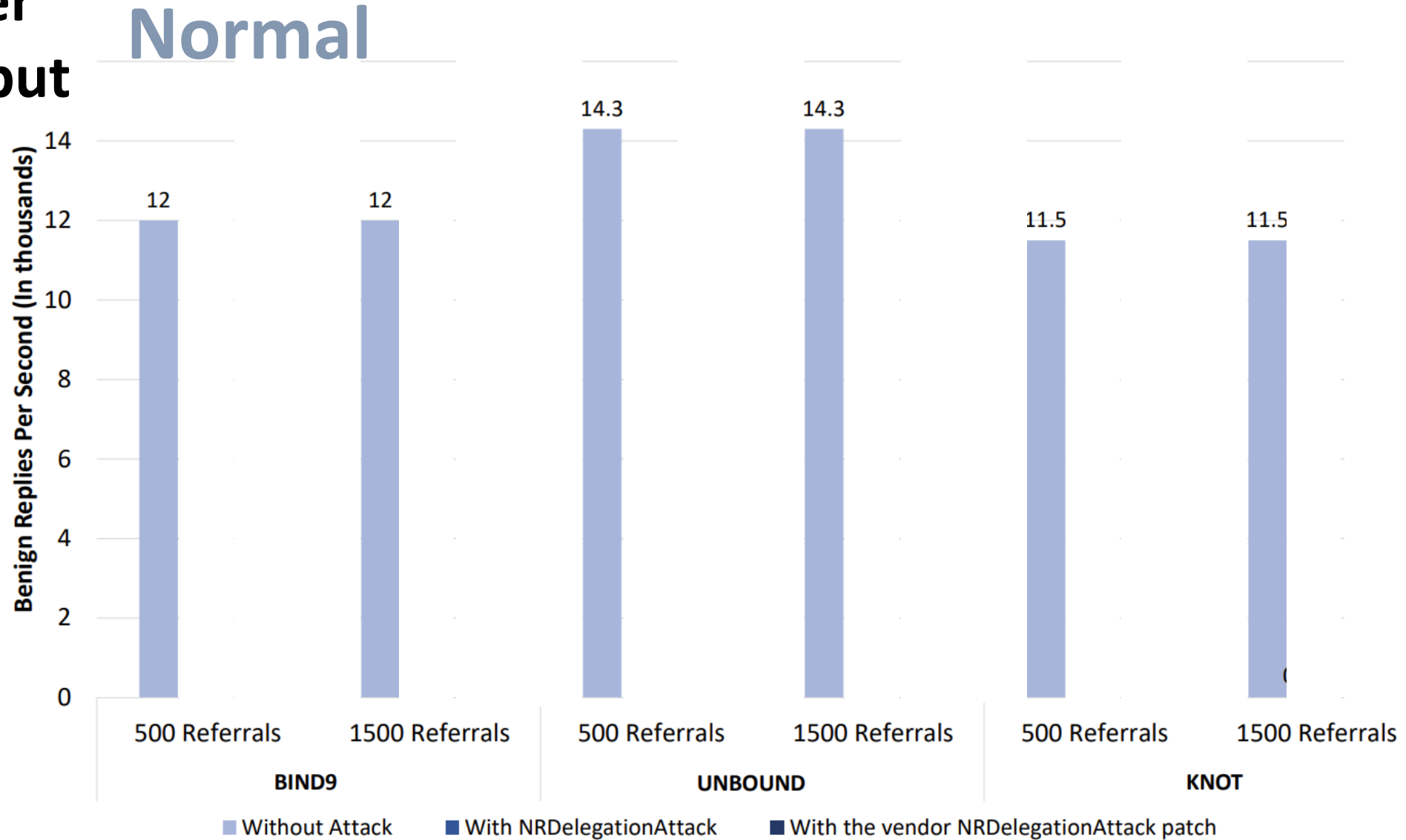
e.g., ~~$n = 1500$~~

20



Throughput Using NRDelegation Mitigation

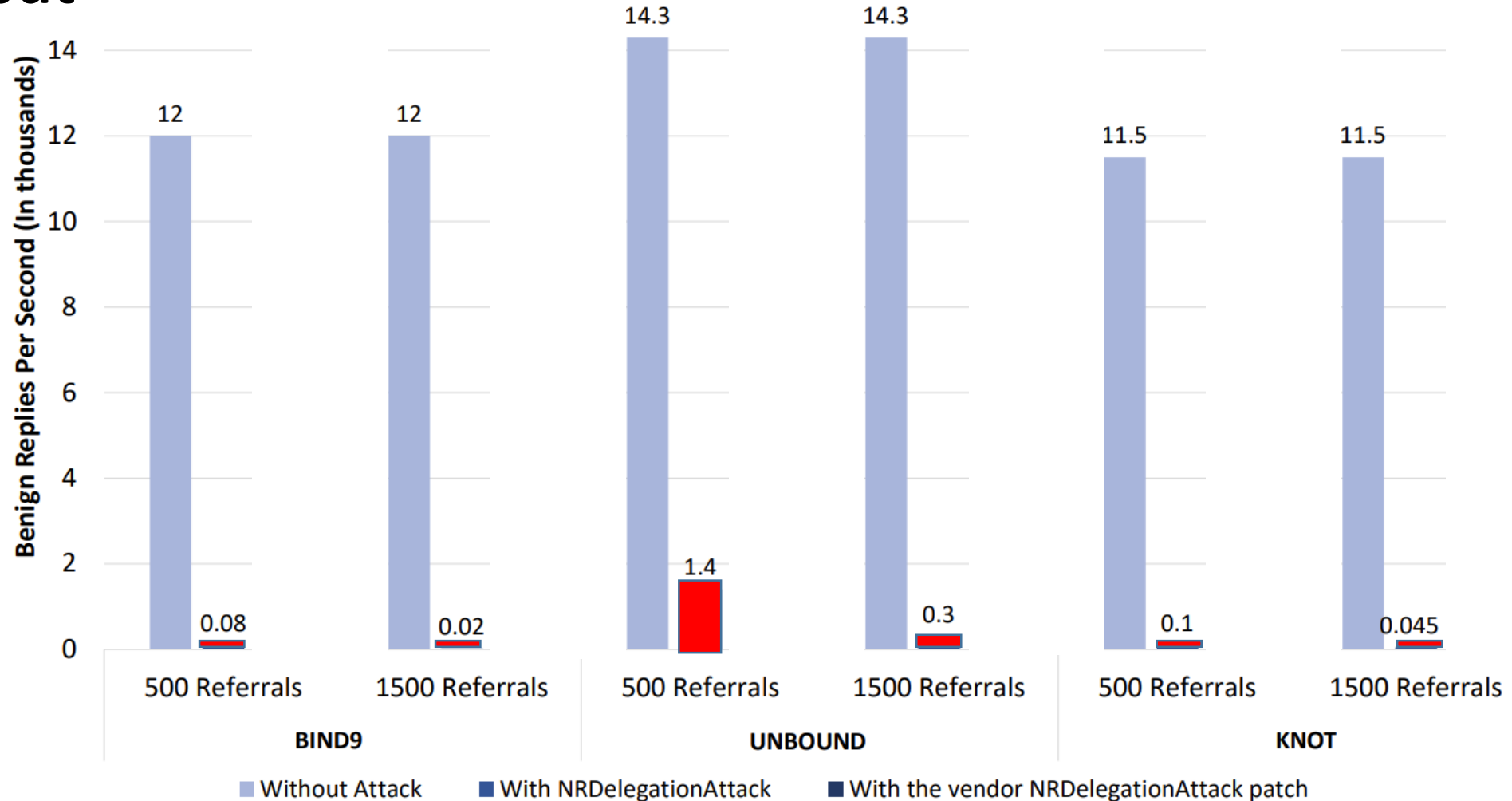
Resolver
throughput



Throughput Using NRDelegation Mitigation

Resolver
throughput

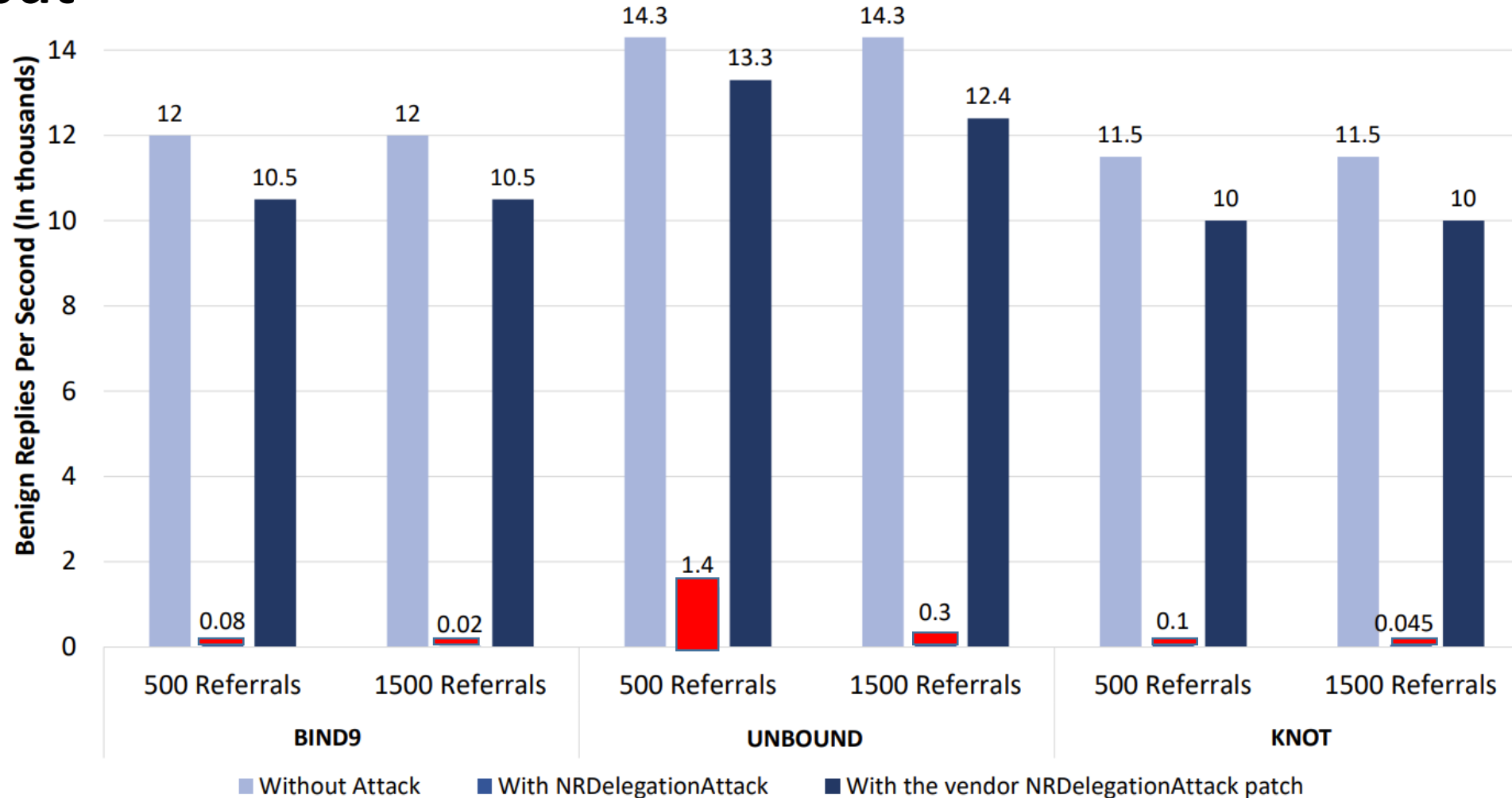
Normal Under NRDelegationAttack



Throughput Using NRDelegation Mitigation

Resolver
throughput

Normal Under NRDelegation w/Mitigation

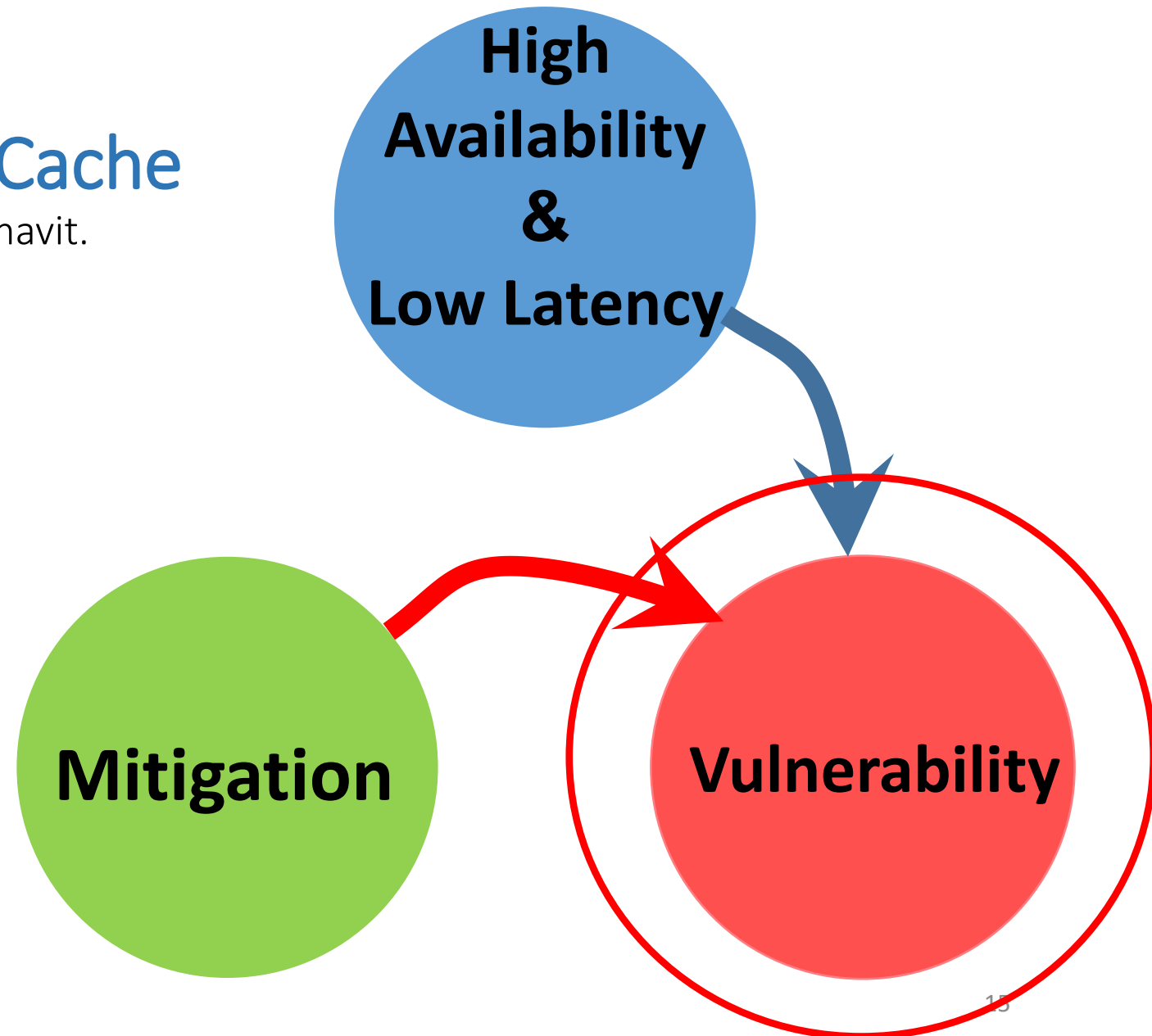


Third cycle example

CacheFlush: Flushing DNS Cache

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

Flush the resolver **cache**



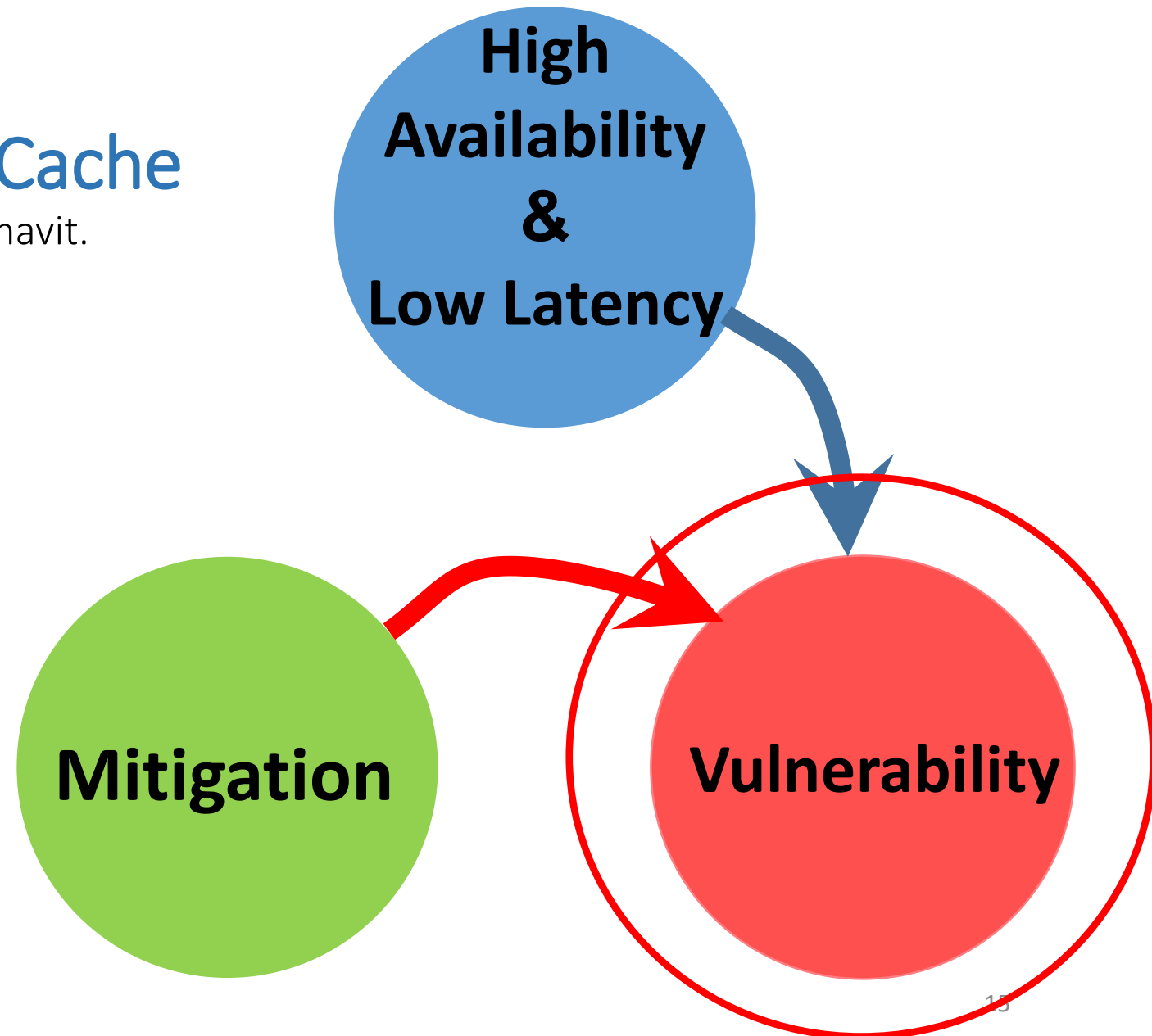
Third cycle example

CacheFlush: Flushing DNS Cache

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

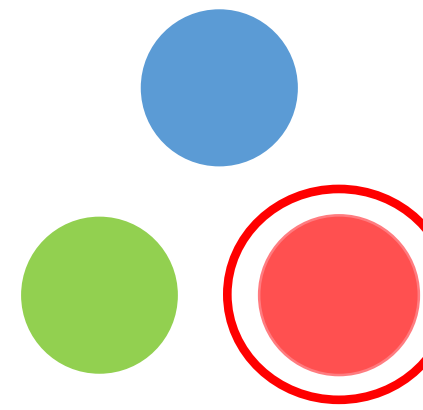
Flush the resolver **cache**

the **benign** cache !

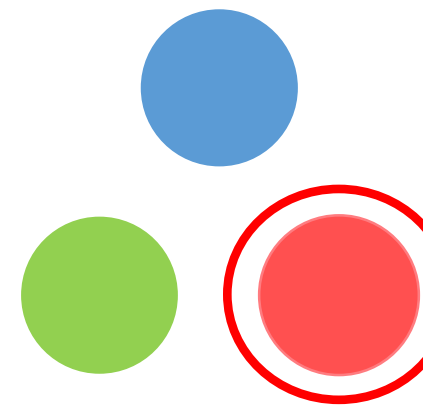


- NXNS → Resolve only 5 referrals
- NRD → Ignore beyond 20 referrals

- NXNS → Resolve only 5 referrals
- NRD → Ignore beyond 20 referrals
- **But keep all RR in Benign Cache !**

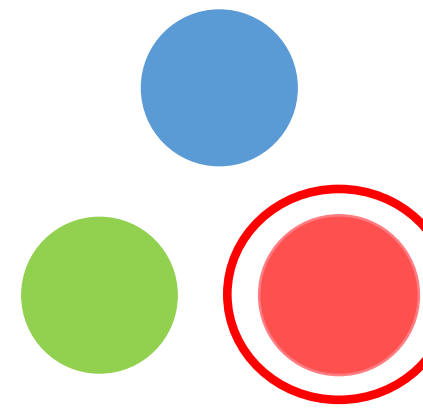


Flushing the DNS Cache (LRU)



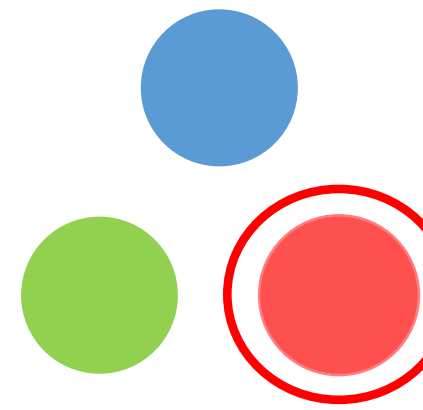
NS fake2.attack.com
NS fake3.attack.com
NS fake4.attack.com
NS fake5.attack.com
NS fake6.attack.com
NS fake7.attack.com
NS fake8.attack.com
NS fake9.attack.com
NS fake0.attack.com
NS fakea.attack.com
NS fakeb.attack.com
NS fakec.attack.com
NS faked.attack.com
NS fakef.attack.com
NS fakeg.attack.com
NS fakeh.attack.com
NS fakei.attack.com

Flushing the DNS Cache (LRU)



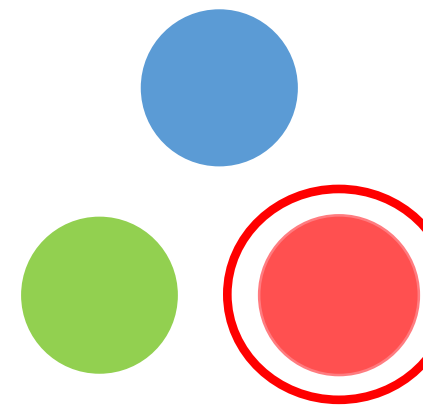
NS fake2.attack.com
NS fake3.attack.com
NS fake4.attack.com
www.google.com
NS fake6.attack.com
www.usenix.com
NS fake8.attack.com
www.univ1.edu
NS fake0.attack.com
www.yahoo.com
NS fakeb.attack.com
NS fakec.attack.com
NS faked.attack.com
www.netflix.com
NS fakeg.attack.com
NS fakeh.attack.com
NS fakei.attack.com

Flushing the DNS Cache (LRU)



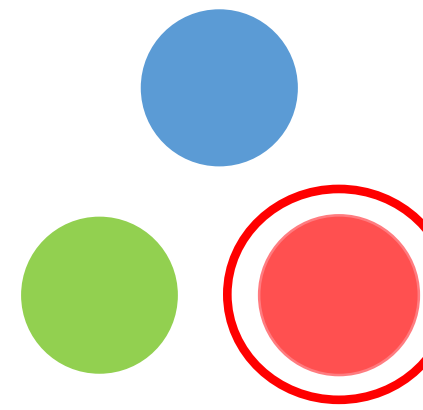
NS fakY.attack.com
NS fakx3.attack.com
NS fakx4.attack.com
NS fakx5.attack.com
NS fakQ6.attack.com
NS fakQ7.attack.com
NS fakQ8.attack.com
NS fakW9.attack.com
NS fakx0.attack.com
NS fakQa.attack.com
NS fakQb.attack.com
NS fakQc.attack.com
NS faked.attack.com
NS fakef.attack.com
NS fakXg.attack.com
NS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

Flushing the DNS Cache (LRU)



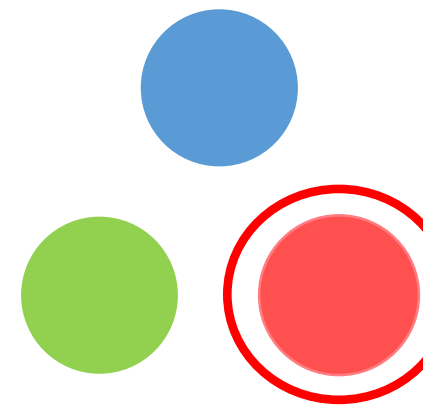
NS fakY.attack.com
NS fakx3.attack.com
NS fakx4.attack.com
www.google.com
NS fakQ6.attack.com
www.usenix.com
NS fakQ8.attack.com
www.univ1.edu
NS fakxU.attack.com
www.yahoo.com
NS fakQb.attack.com
NS fakQc.attack.com
NS faked.attack.com
www.netflix.com
NS fakXg.attack.com
NS fakXh.attack.com
NS fakXi.attack.com

Flushing the DNS Cache (LRU)



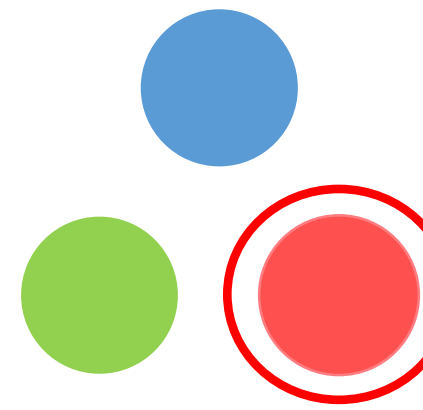
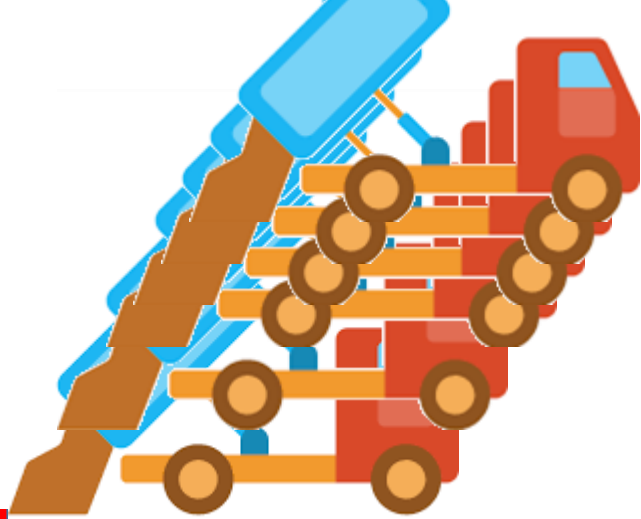
```
NS fake2.attack.com  
NS fake3.attack.com  
NS fake4.attack.com  
NS fake5.attack.com  
NS fake6.attack.com  
NS fake7.attack.com  
NS fake8.attack.com  
NS fake9.attack.com  
NS fake0.attack.com  
NS fakea.attack.com  
NS fakeb.attack.com  
NS fakec.attack.com  
NS faked.attack.com  
NS fakef.attack.com  
NS fakeg.attack.com  
NS fakeh.attack.com  
NS fakei.attack.com
```

Flushing the DNS Cache (LRU)



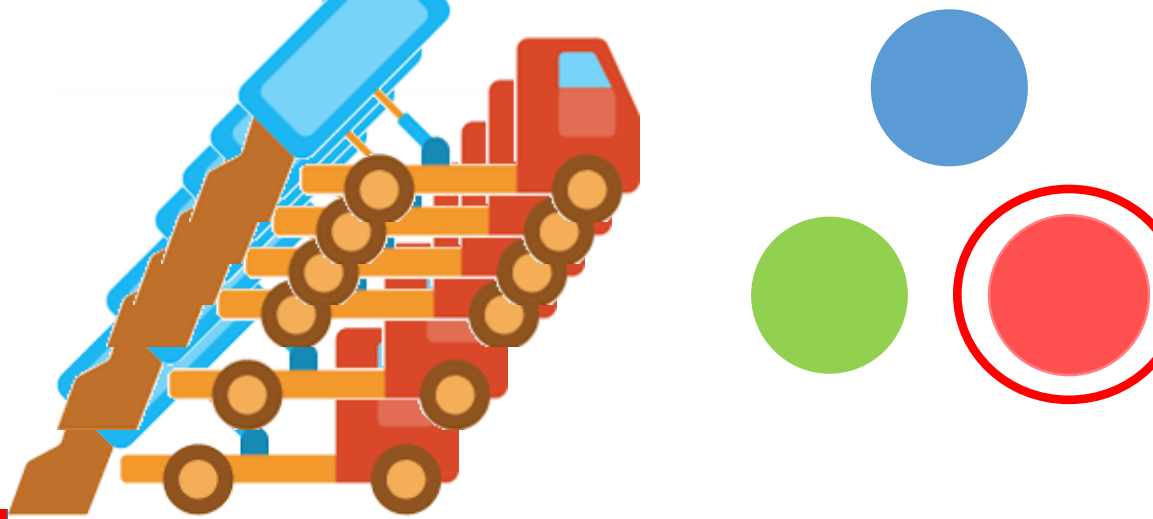
NS fake2.attack.com
NS fake3.attack.com
NS fake4.attack.com
NS fake5.attack.com
www.google.com
NS fake7.attack.com
www.usenix.com
NS fake8.attack.com
NS fake9.attack.com
www.univ1.edu
NS fakea.attack.com
www.yahoo.com
NS fakec.attack.com
NS faked.attack.com
NS fakef.attack.com
www.netflix.com
NS fakeh.attack.com
NS fakei.attack.com

Flushing the DNS Cache (LRU)



NS fake2.attack.com
NS fake3.attack.com
NS fake4.attack.com
NS fake5.attack.com
www.google.com
NS fake7.attack.com
www.usenix.com
NS fake8.attack.com
NS fake9.attack.com
www.univ1.edu
NS fakea.attack.com
www.yahoo.com
NS fakec.attack.com
NS faked.attack.com
NS fakef.attack.com
www.netflix.com
NS fakeh.attack.com
NS fakei.attack.com

Flushing the DNS Cache (LRU)

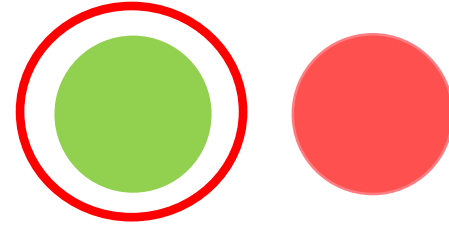


NS fake2.attack.com
NS fake3.attack.com
NS fake4.attack.com
NS fake5.attack.com
www.google.com
NS fake6.attack.com
www.usenix.com
NS fake8.attack.com
NS fake9.attack.com
www.univ1.edu
NS fakea.attack.com
www.yahoo.com
NS fakec.attack.com
NS faked.attack.com
NS fakef.attack.com
www.netflix.com
NS fakeh.attack.com
NS fakei.attack.com

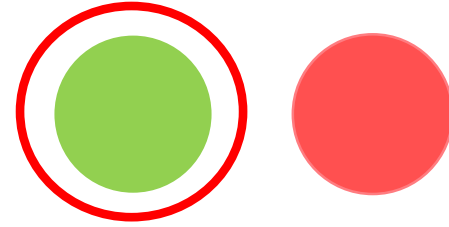
Knot not affected.
Why?

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

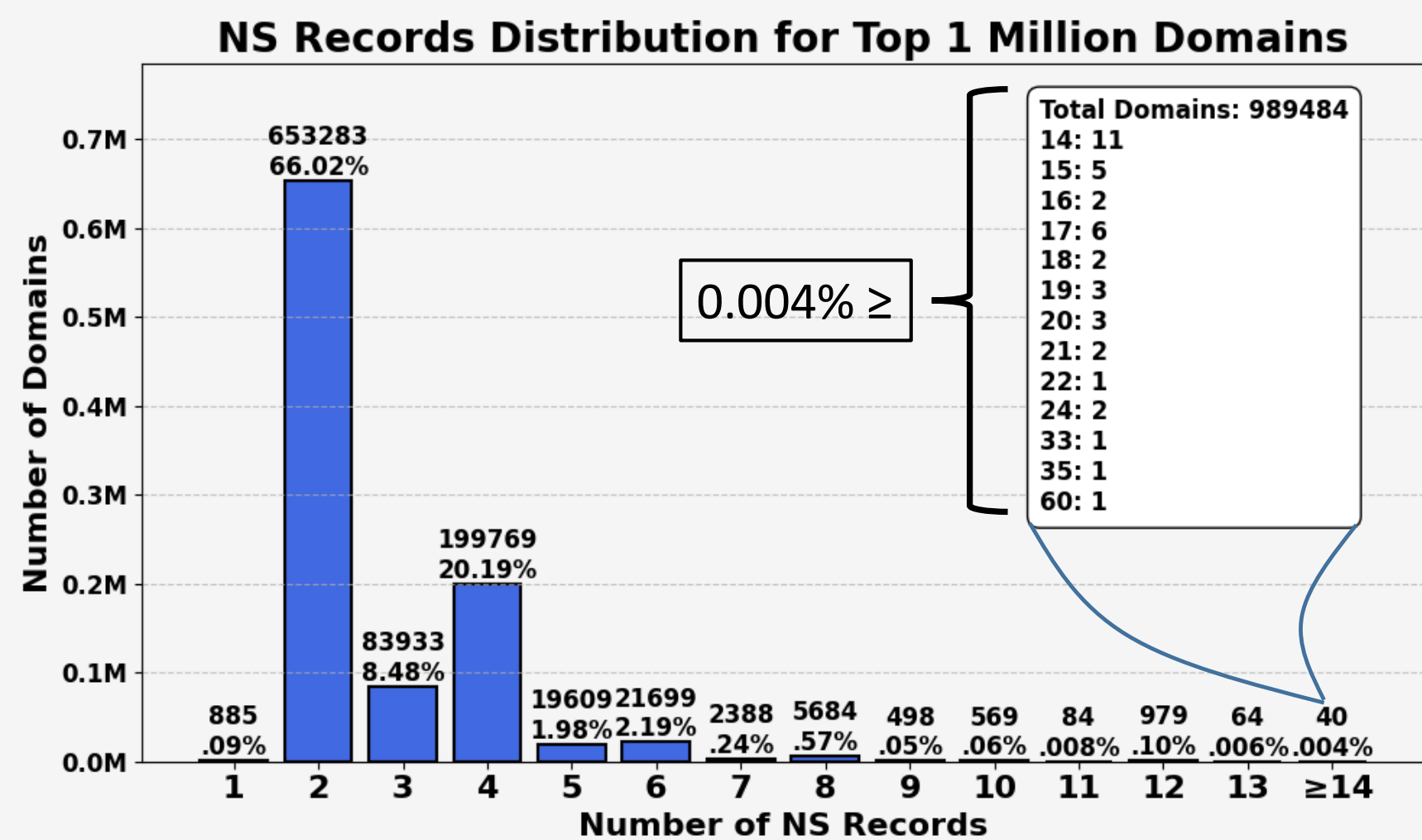
What should MAX #RR be?

- 100?

What should MAX #RR be?

Measured 17/10/2024

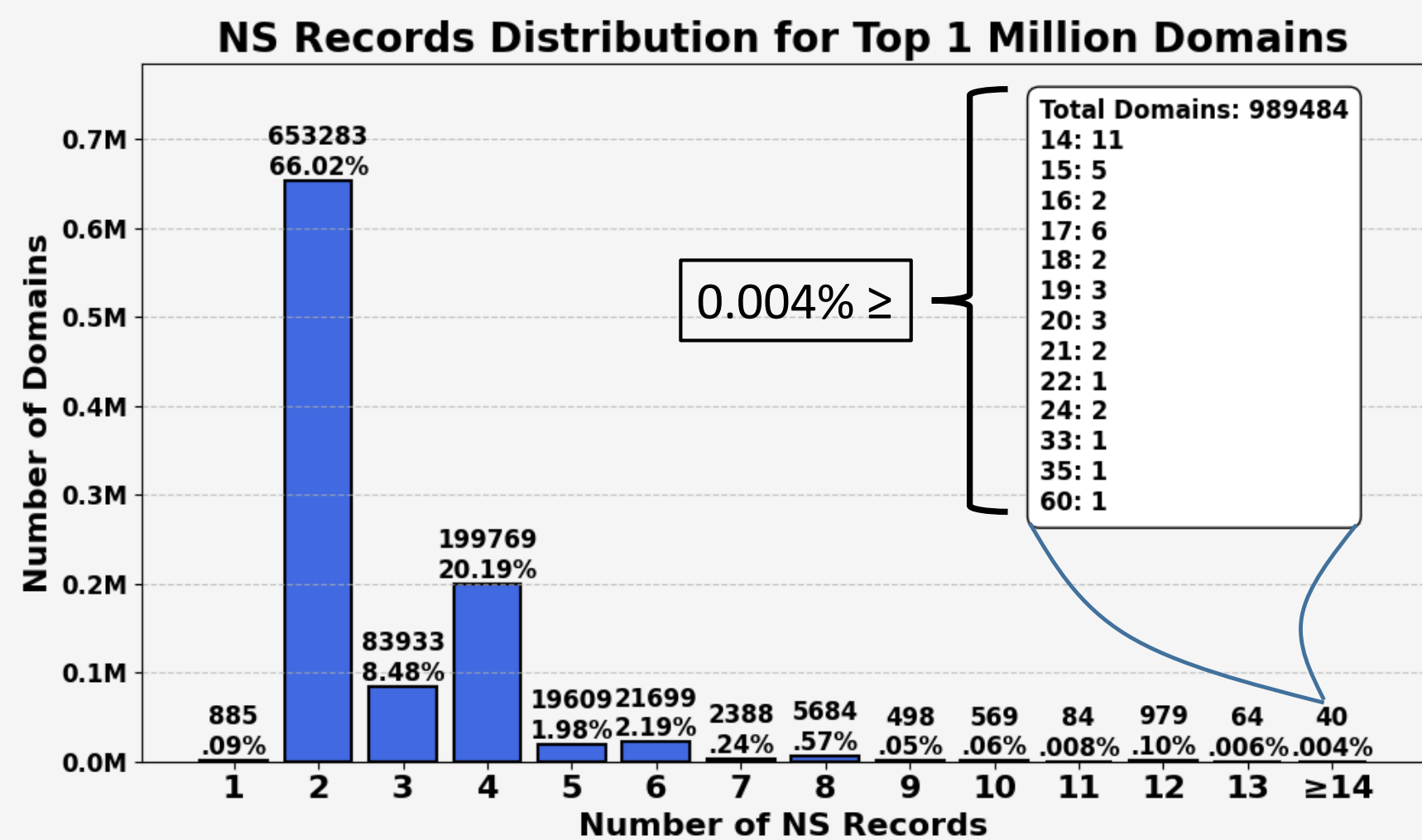
- 100?



What should MAX #RR be?

Measured 17/10/2024

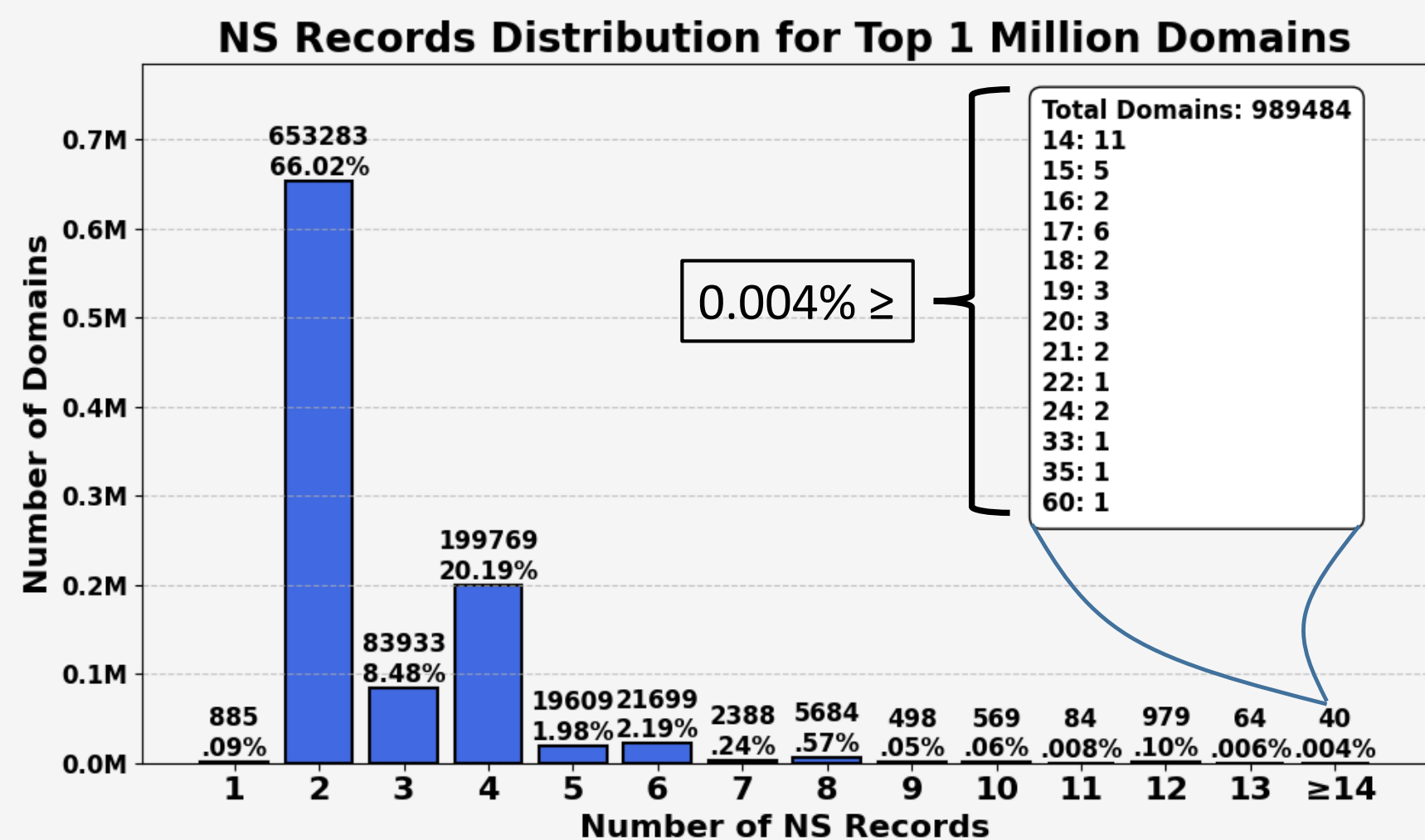
- 100? 20?



What should MAX #RR be?

Measured 17/10/2024

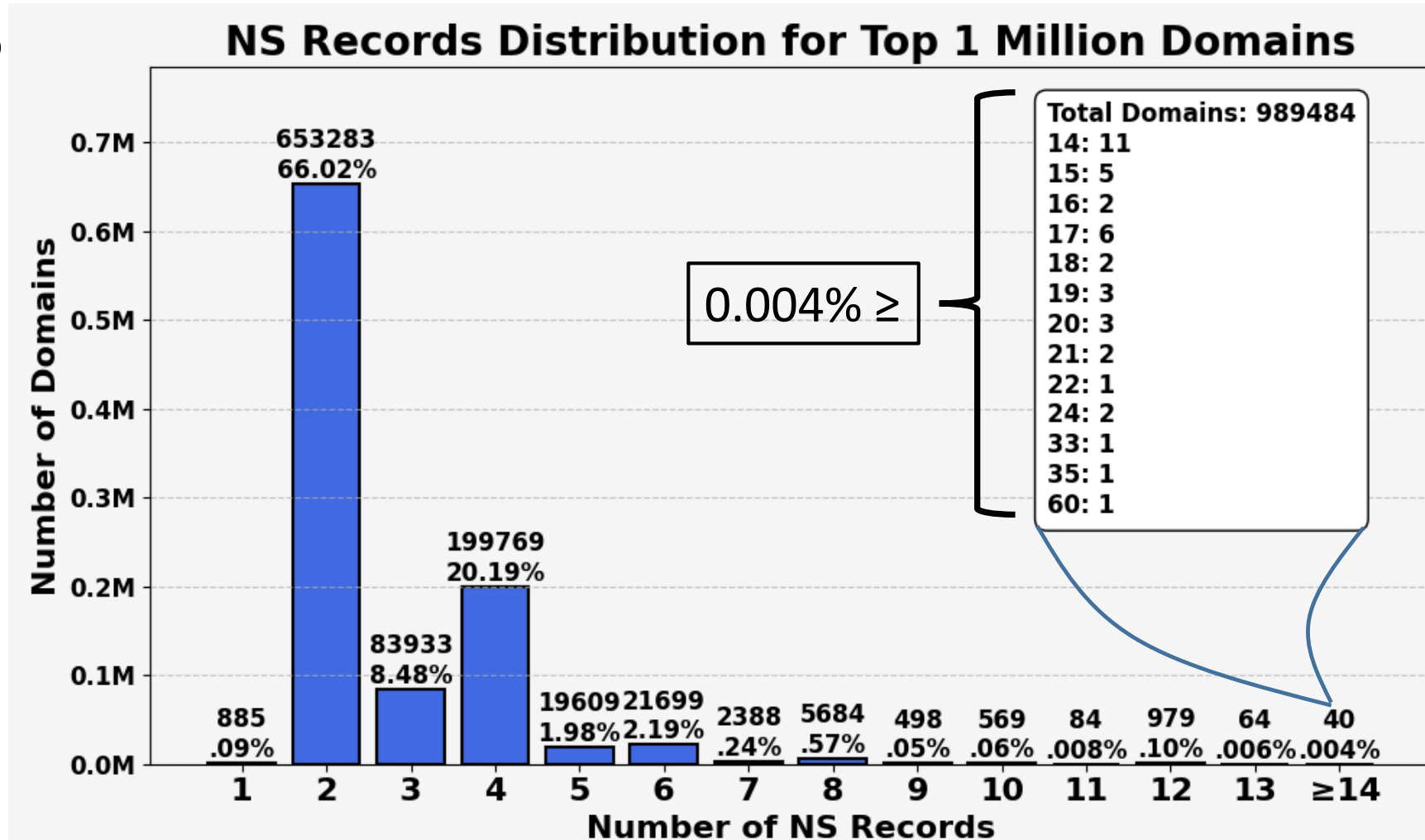
- 100? 20? 13?



What should MAX #RR be?

Measured 17/10/2024

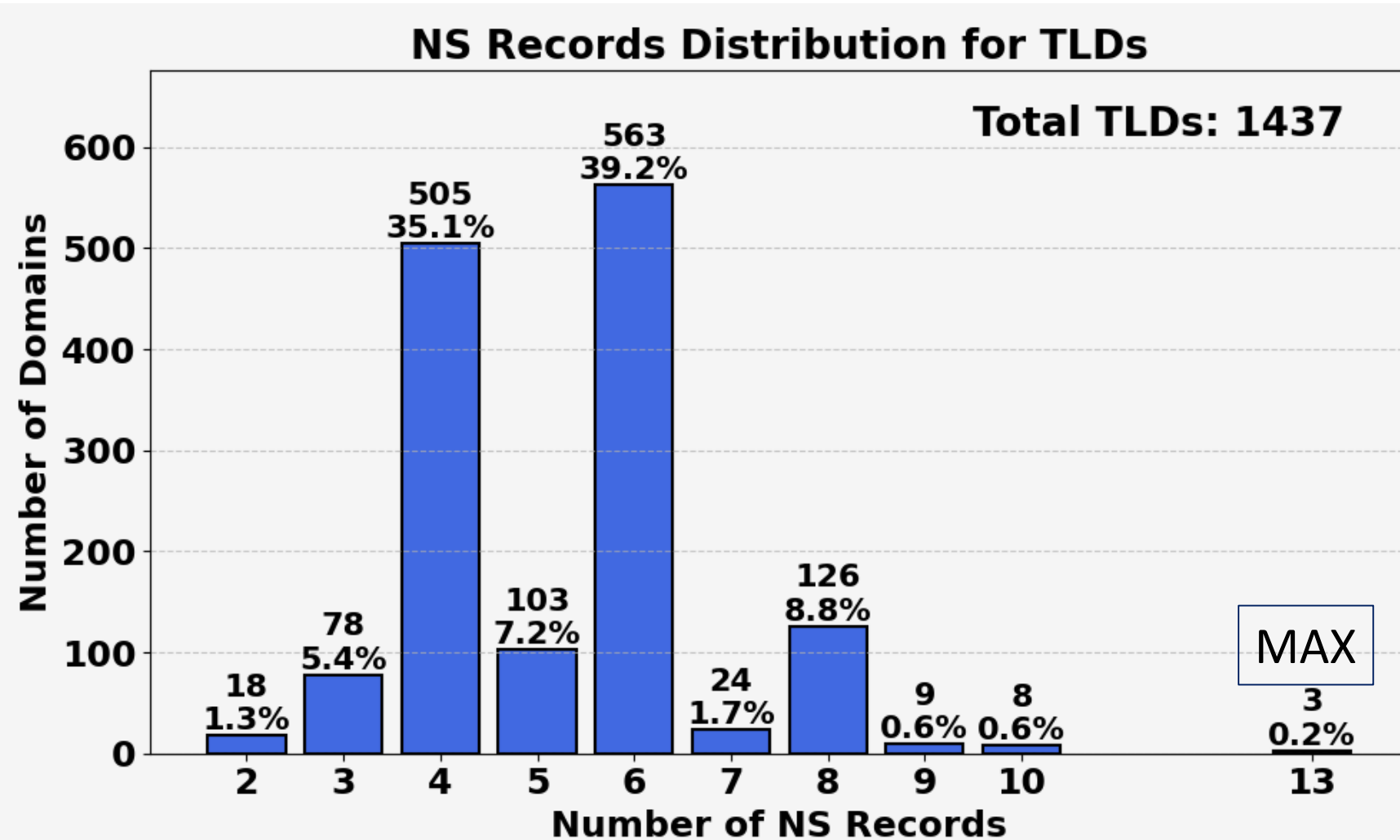
- 100? 20? 13? 10?



What should MAX #RR be?

Measured 17/10/2024

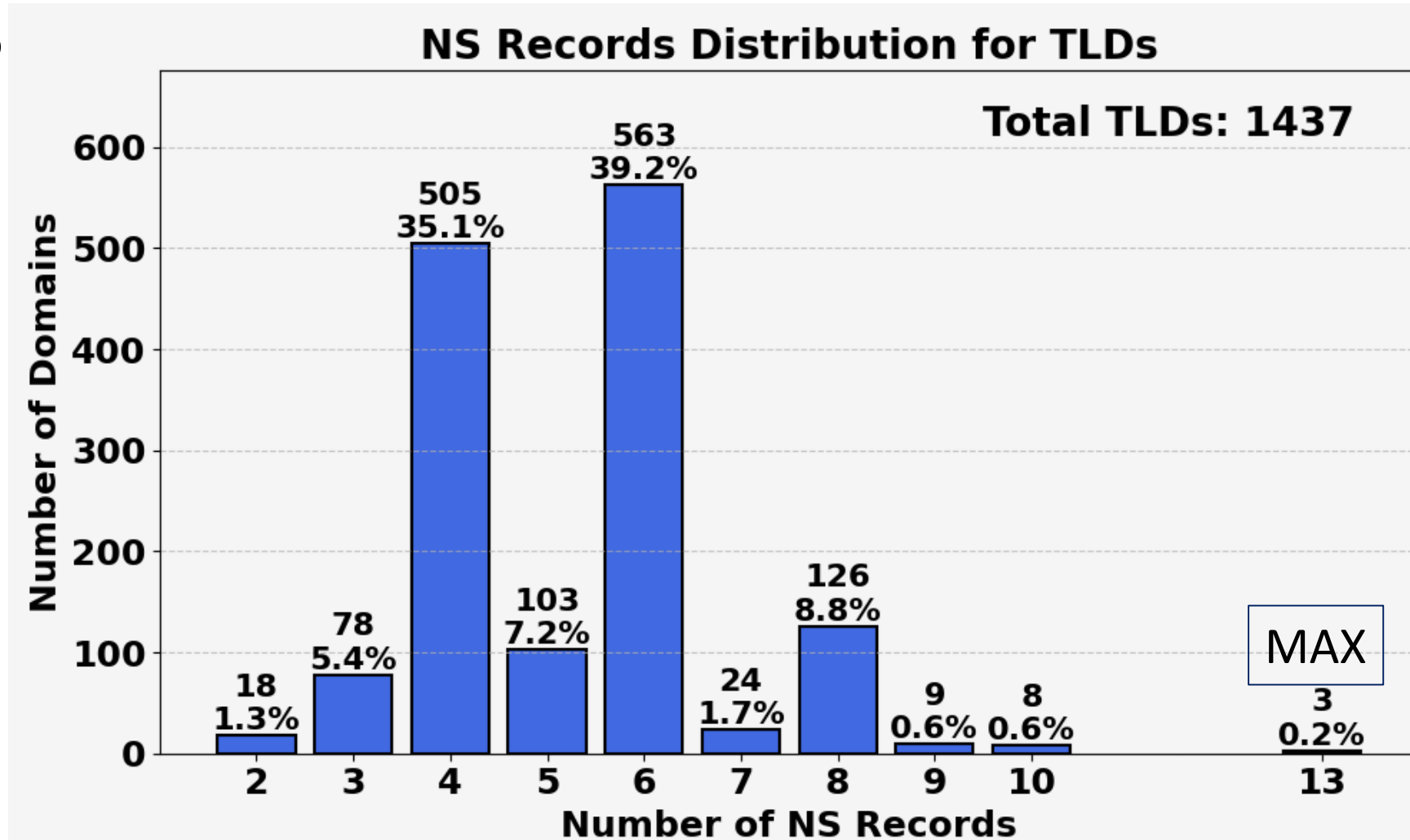
- 100? 20? 13? 10?



What should MAX #RR be?

- 100? 20? **13?** 10?

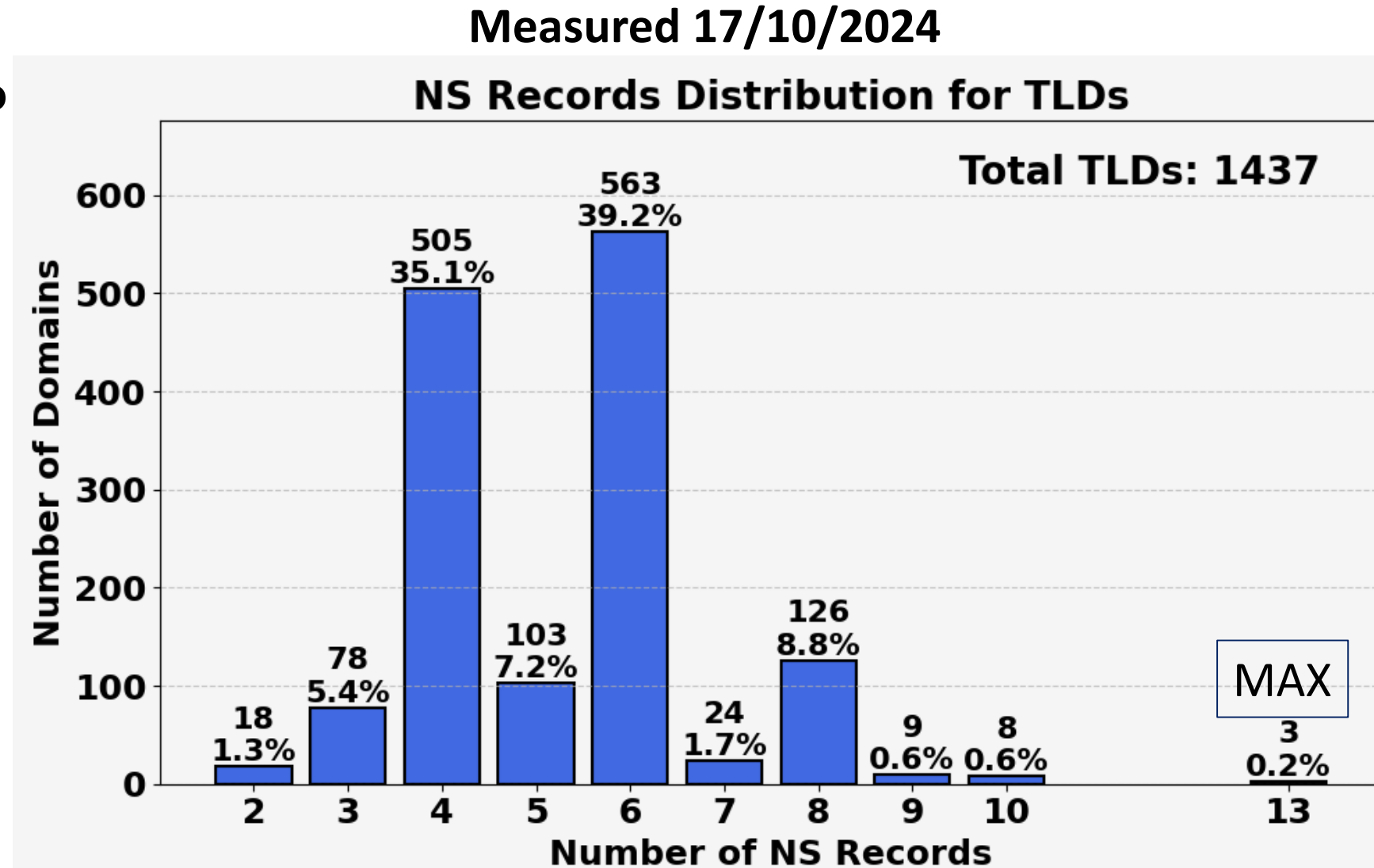
Measured 17/10/2024



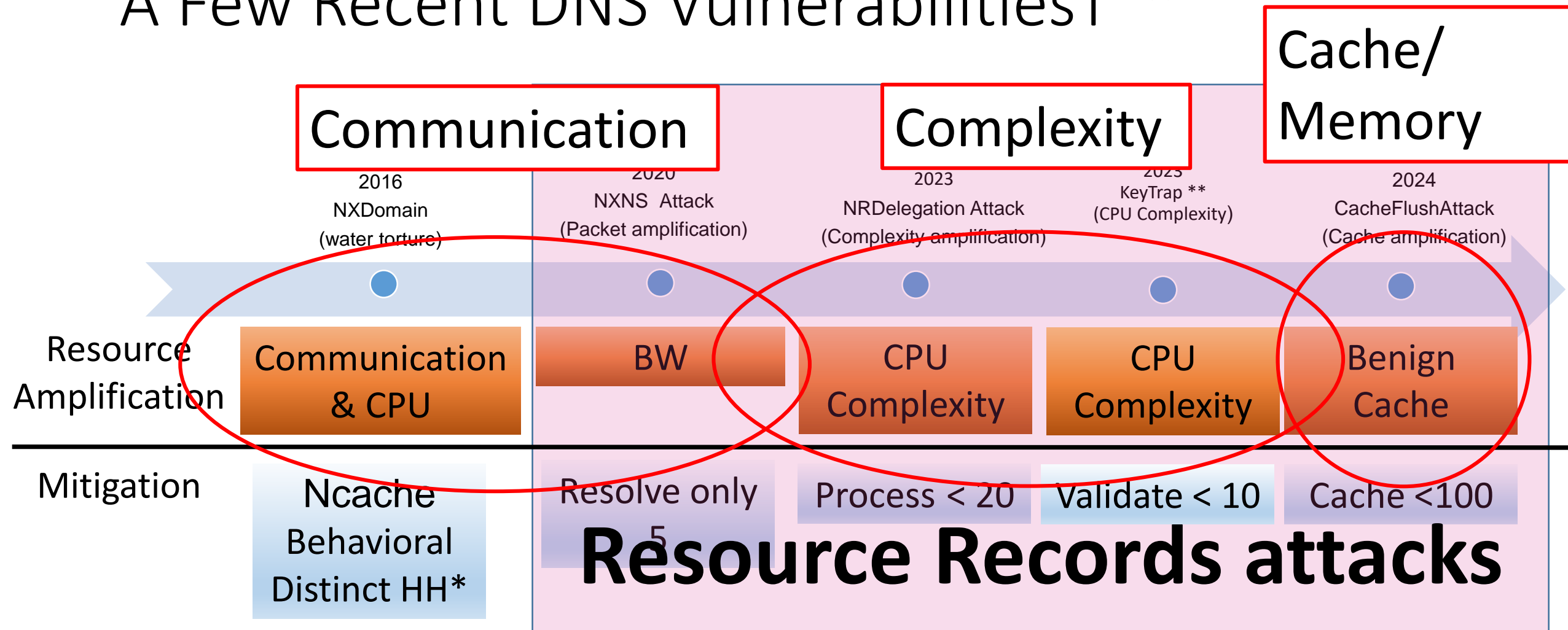
What should MAX #RR be?

- 100? 20? **13?** 10?

For all kinds & shape
of RR



A Few Recent DNS Vulnerabilities*



* [[Landau Feibish, A, Bremler-Barr, Cohen, Shaqam](#): 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?