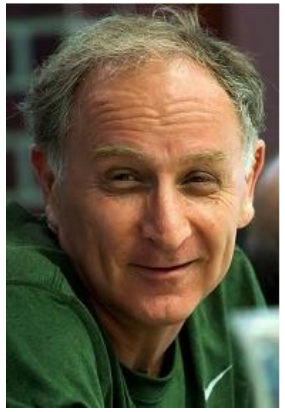
Adaptive DNSSEC



Yehuda Afek
Tel-Aviv University

Anat Bremler-Barr
Reichman University

Daniel Dubnikov
Tel-Aviv University

Supported By:

Motivation (1)

DNSSEC significantly degrades DNS performances

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989

NXDomain Attack

RANDOM DNS Request Flood

Rxy1xhggsgVCER.sony.com

XVBY$&HGDRxy2.sony.com

FJH*^DHGAKRxy3.sony.com

RxUYQVMNLKAy4.sony.com

Resolvers

RzHW$RE43CBJs$7.sony.com

.com

ns.sony.com

Motivation (1)

- DNSSEC significantly degrades DNS performances
  - ➔ DDoS amplification

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989

* All measurements are under NX queries flood attack

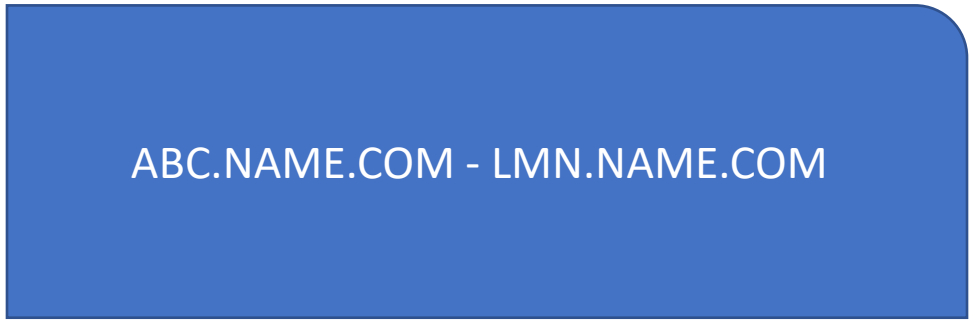
Motivation (1)

- Non Existent in DNSSEC?

Motivation (1)

- Non Existent in DNSSEC?

- NSEC Record:



Motivation (1)

- Non Existent in DNSSEC?

- NSEC Record:

Motivation (2)

- Aggressive Caching (RFC 8198) – stops NX Attack

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989
Aggressive Caching	**Max Queries Per Second**
DNSSEC: NSEC	96,226
DNSSEC: NSEC3	93,756

Motivation (2)

- Aggressive Caching (RFC 8198) – stops NX Attack

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989

Aggressive Caching	Max Queries Per Second
DNSSEC: NSEC	96,226
DNSSEC: NSEC3	93,756

Only Knot

Motivation (2)

- Aggressive Caching (RFC 8198) – stops NX Attack
- BUT: Enables Zone Walking

ABC.NAME.COM - LMN.NAME.COM

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989

Aggressive Caching	Max Queries Per Second
DNSSEC: NSEC	96,226
DNSSEC: NSEC3	93,756

Only Knot

Motivation (2)

- Aggressive Caching (RFC 8198) – stops NX Attack
- BUT: Enables Zone Walking
  Scalability Issues: Need to quickly find NSEC record

	Max Queries Per Second
Plain DNS	23,524
DNSSEC: NSEC	9,510
DNSSEC: NSEC3	8,989
Aggressive Caching	**Max Queries Per Second**
DNSSEC: NSEC	96,226
DNSSEC: NSEC3	93,756

Only Knot

Motivation (2)

- How to stop Zone Walking?

Motivation (2)

- How to stop Zone Walking?
  - You can’t (without online signing*, Goldberg et al.)

* Goldberg et al.
Nsec5: Provably preventing dnssec zone enumeration

Motivation (2)

- How to stop Zone Walking?
  - You can’t (without online signing*, Goldberg et al.)

- Black Lies, White Lies, NSEC5

* Goldberg et al.
Nsec5: Provably preventing dnssec zone enumeration

Motivation (2)

- How to stop Zone Walking?
  - You can’t (without online signing*, Goldberg et al.)

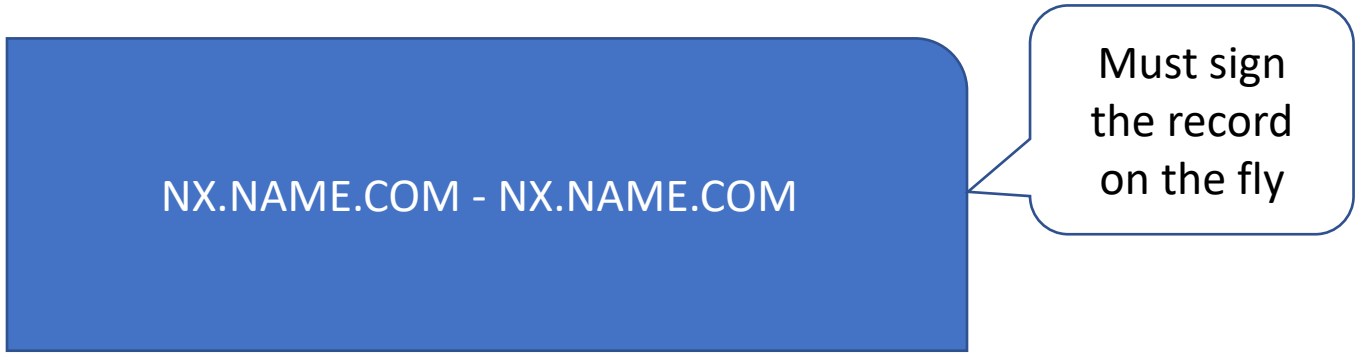
- Black Lies, White Lies, NSEC5

* Goldberg et al.
Nsec5: Provably preventing dnssec zone enumeration

Motivation (2)

- How to stop Zone Walking?
  - You can’t (without online signing*, Goldberg et al.)

- Black Lies, White Lies, NSEC5



* Goldberg et al.
Nsec5: Provably preventing dnssec zone enumeration

Motivation (3)

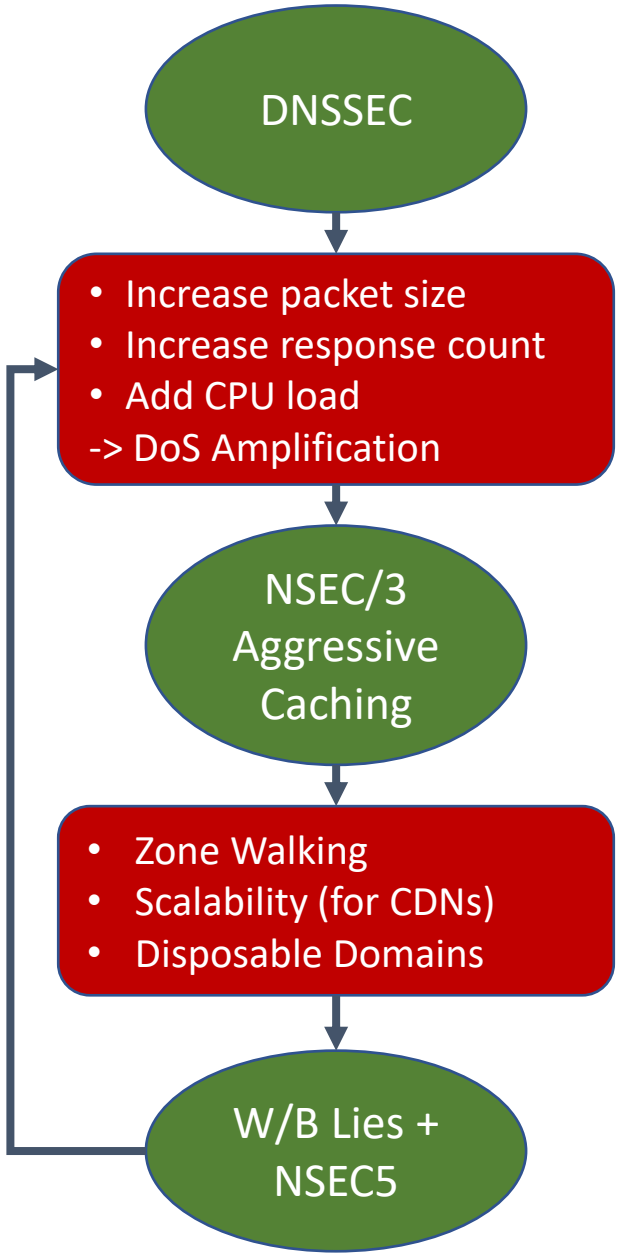
- Online Signing Algorithms – NX Attacks Amplified

	Max Queries Per Second	% of Plain DNS
Plain DNS	23,524	100%
DNSSEC: NSEC	9,510	40%
DNSSEC: NSEC3	8,989	38%
DNSSEC: White Lies	5,863	25%
DNSSEC: Black Lies	7,206	30%
DNSSEC: NSEC5	6,324	27%

Motivation (3)

- Online Signing Algorithms – NX Attacks Amplified
- **For security and scalability reasons online signing might be the only option**

	Max Queries Per Second	% of Plain DNS
Plain DNS	23,524	100%
DNSSEC: NSEC	9,510	40%
DNSSEC: NSEC3	8,989	38%
DNSSEC: White Lies	5,863	25%
DNSSEC: Black Lies	7,206	30%
DNSSEC: NSEC5	6,324	27%


DNSSEC
• Increase packet size
• Increase response count
• Add CPU load
-> DoS Amplification
NSEC/3
Aggressive
Caching
• Zone Walking
• Scalability (for CDNs)
• Disposable Domains
W/B Lies +
NSEC5

Proposed Solution

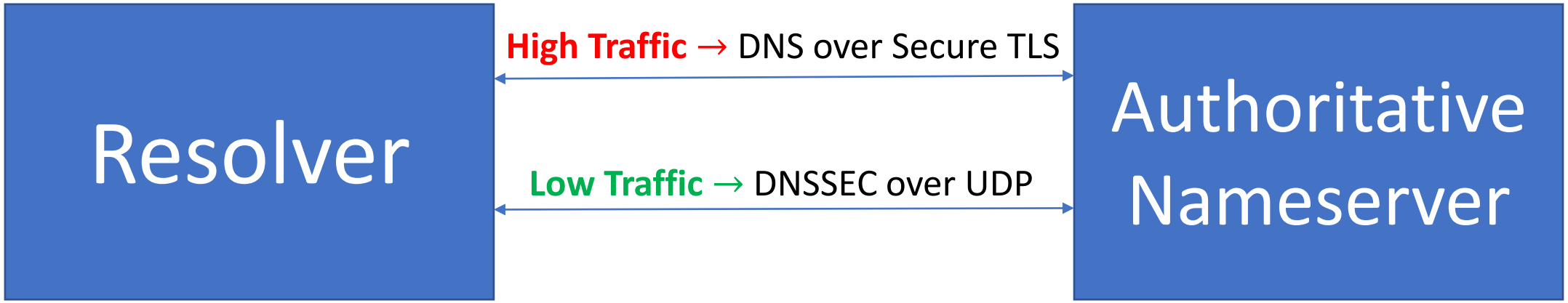
- Remove all DNSSEC overheads
  - Packet Size + Count
  - CPU Load

Proposed Solution

- Remove all DNSSEC overheads
  - Packet Size + Count
  - CPU Load

- Use TLS
  - Authoritative identifies once
  - Traffic sent with Plain DNS

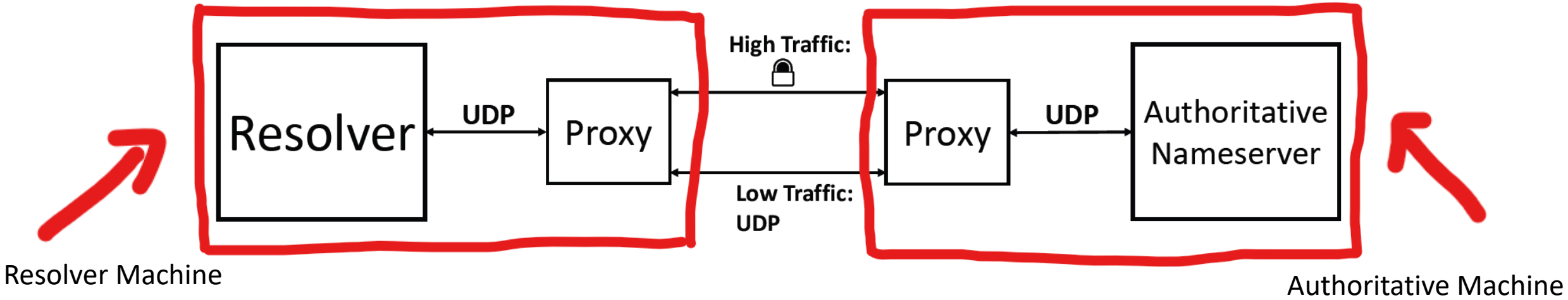
Proposed Solution - Hybrid



Proposed Solution – PoC (1)

- **Problem1:** Can't easily integrate with known resolver/auth implementations (Bind, Unbound, Knot, etc.)

- **Solution: proxy** interface resolver - authoritative servers



Proposed Solution – PoC (2)

- **Problem2**:
  - TLS overheads are high
  - TLS suffers from Head-of-Line blocking

Proposed Solution – PoC (2)

- **Problem2**:
  - TLS overheads are high
  - TLS suffers from Head-of-Line blocking
- **Solution:** Use QUIC (similar to HTTP3)
  - UDP Multiplexing (virtual connections)
  - 0-RTT: resume the connection with 0 round trip time
  - No need for TCP integration (firewalls/IPS)

Proposed Solution – PoC (3)

- **Problem3**: Teardown and restart QUIC connections

Proposed Solution – PoC (3)

- **Problem3**: Teardown and restart QUIC connections
- **Solution:** Keep connections alive
  - QUIC has low overhead – long lived connections
  - QUIC can resume quickly

Proposed Solution – PoC (4)

- **Problem4:** Resource limit

Proposed Solution – PoC (4)

- **Problem4:** Resource limit
- **Solution:** Score connection throughput with

$$F_i = \alpha \cdot F_{i-1} + (1 - \alpha) \cdot f_i$$

Terminate lowest scored connection (LRU)

Measurements

Knot	Max Queries Per Second	% of Plain DNS
Plain DNS	23,524	100%
DNSSEC: NSEC	9,510	40%
DNSSEC: NSEC3	8,989	38%
DNSSEC: White Lies	5,863	25%
DNSSEC: Black Lies	7,206	30%
DNSSEC: NSEC5	6,324	27%

Measurements

With QUIC, using the same experiment (NX flood), throughput is **87%** of the plain DNS

Knot	Max Queries Per Second	% of Plain DNS
Plain DNS	23,524	100%
DNSSEC: NSEC	9,510	40%
DNSSEC: NSEC3	8,989	38%
DNSSEC: White Lies	5,863	25%
DNSSEC: Black Lies	7,206	30%
DNSSEC: NSEC5	6,324	27%
AdaDoQ (Our Solution)	**20,558**	**87%**

Conclusions

- DNSSEC degrades DNS performance
  - Make NXDOMAIN attacks worse (DDoS amplification)
- AdaDoQ – Hybrid Solution
  - Light and fast connections
  - One time encryption overheads
  - Close to Plain DNS throughput
  - No Security Compromises
    - Zone Walking
  - No Scalability Issues

Questions?