A Security Evaluation of DNSSEC with NSEC3

Jason Bau, John Mitchell
Stanford University

Stanford Computer Security Lab

Motivation

- DNSSEC around for >10 years, adoption on the way

Motivation

- DNSSEC around for >10 years, adoption on the way

comcastvoices
a place for conversations with Comcast

Search this Blog GO
Advanced Search

Home | Archives | Media Gallery | About | Help

23 FEB

DNSSEC

Posted by Chris Griffiths, Manager, DNS Engineering, in Innovation

For the past couple years, Comcast has been testing and advocating for the widespread adoption of DNS Security extensions (also known as DNSSEC). If you don't know what DNSSEC is, you're probably not alone. Basically, it allows websites to secure their domain information so that ISPs can validate and make sure nothing has been tampered with. This prevents hackers from injecting false information (aka DNS 'poisoning') that re-directs you to a fake or nefarious site. The process needed to secure domains as well as validate them is very complex and that is why we are taking time over the next year to make sure everything works.

We plan to implement DNSSEC for the websites we manage, such as comcast.com, comcast.net and xfinity.com, by the first quarter of 2011, if not sooner. By the end of 2011, we plan to implement DNSSEC validation for all of our customers. You won't need to make any changes to start using DNSSEC; it will happen automatically if you are currently using our DNS.

Goal of Our Study

- Evaluate DNSSEC from perspective of enterprise considering adoption
- Scientific study of DNSSEC/NSEC3 protocol
  - Model-Checking methodology
  - Found violations of stated security conditions
    - Mostly due to design trade-off
    - Investigate potential resultant danger
  - Few observations
- Offer best-practice DNSSEC/NSEC3 configuration advice

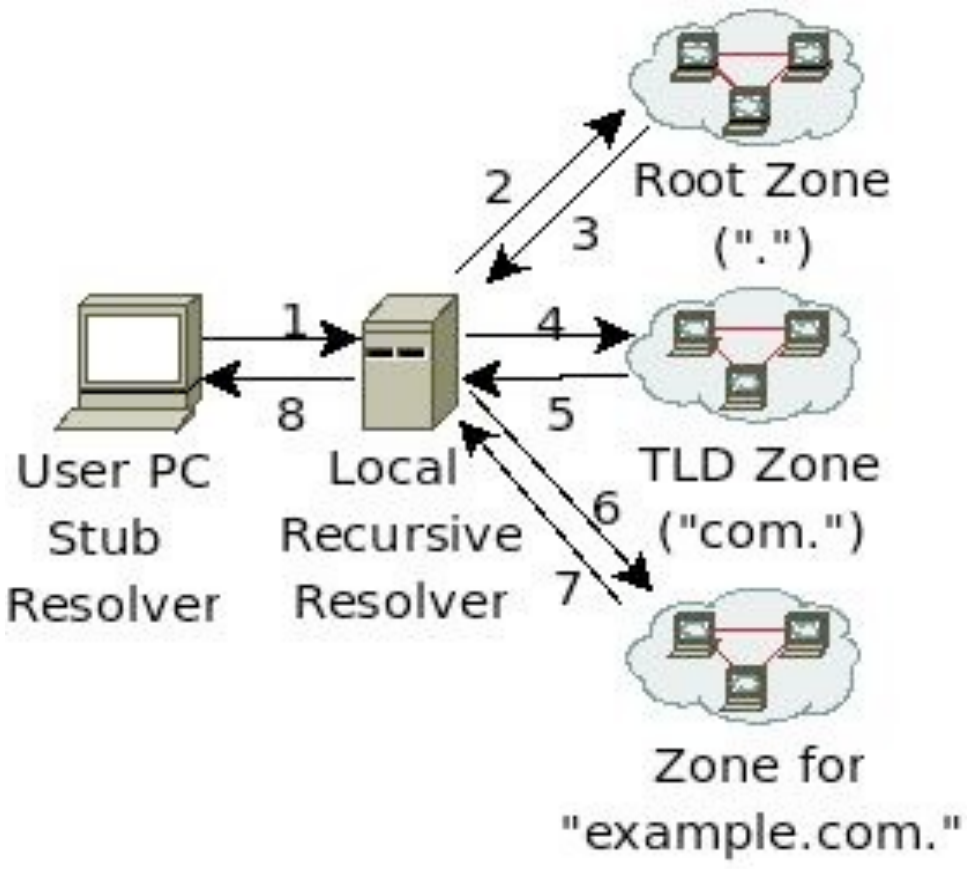
Outline

- Background
  - DNS
  - DNSSEC
- Finite State Enumerator (Murφ) analysis
  - Security Guarantees
  - Attested Cache Resolver Design
  - Cached Record Temporal Dependencies
  - Insecure Sub-Namespace of DNSSEC zone
    - Cookie-Theft
- DNSSEC Security Observations
- Configuration Advice and Conclusions

Background DNS Lookup

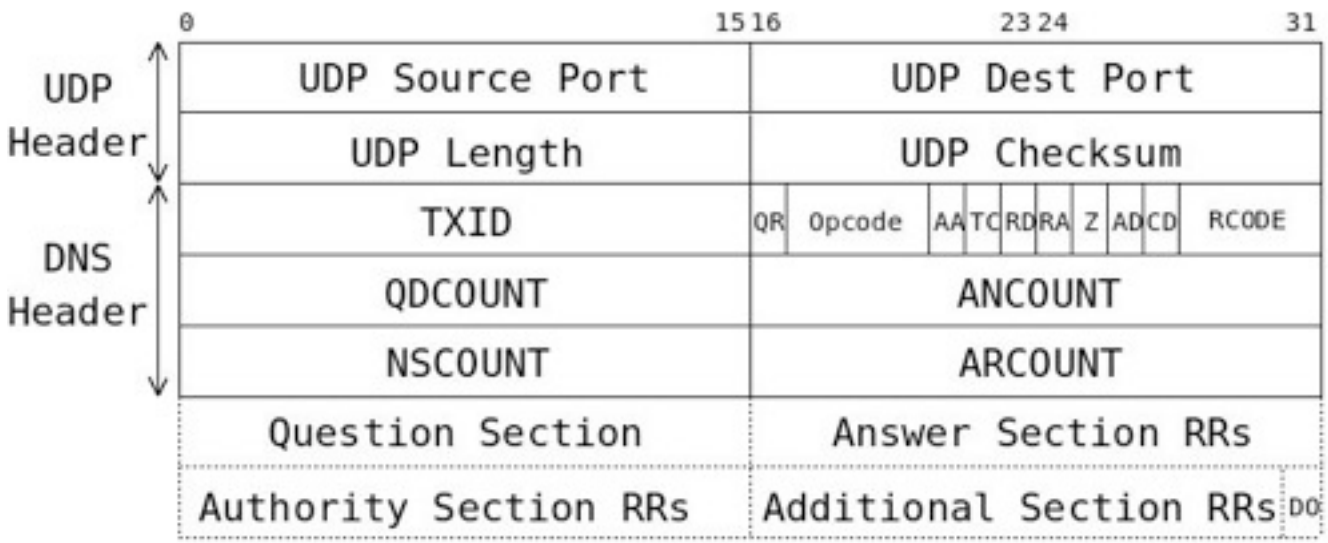
Query: "www.example.com A?"

Reply	Resource Records in Reply
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"
7	"www.example.com A 1.2.3.4"
8	"www.example.com A 1.2.3.4"



Local recursive resolver caches these for TTL specified by RR

DNS Packet Format



TXID = Transaction ID
QR = Query or Reply
Opcode = Typically 0 (QUERY)
AA = Authoritative Answer
TC = Truncated
RD = Recursion Desired
RA = Recursion Available
Z = Zero Bit
AD = Authenticated Data
CD = Checking Disabled
RCODE: 0 = No Error
2 = Server Failure or Bogus DNSSEC data
DO = DNSSEC OK (in EDNS0 header)

- Sent over UDP, < 512 Bytes
- TXID, UDP Source Port only "security" features

DNS is Insecure

- Packets over UDP, < 512 bytes
- 16-bit TXID, UDP Src port only “security”
- Resolver accepts packet if above match
- Packet from whom? Was it manipulated?

- Cache poisoning
  - Attacker forges record at resolver
  - Forged record cached, attacks future lookups
  - Kaminsky (BH USA08)
    - Attacks delegations with “birthday problem”

DNSSEC Goal

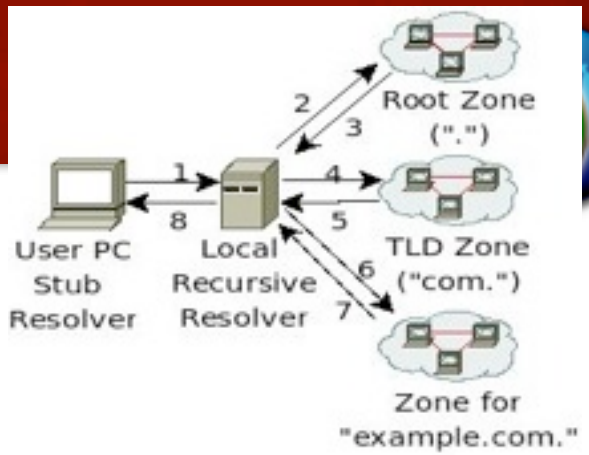
"The Domain Name System (DNS) security extensions provide origin authentication and integrity assurance services for DNS data, including mechanisms for authenticated denial of existence of DNS data."

-RFC 4033

DNSSEC

- Basically no change to packet format
  - Object security of DNS data, not channel security
- New Resource Records (RRs)
  - RRSIG : signature of RR by private zone key
  - DNSKEY : public zone key
  - DS : crypto digest of child zone key
  - NSEC / NSEC3 :authenticated denial of existence
- Lookup referral chain (unsigned)
- Origin attestation chain (PKI) (signed)
  - Start at pre-configured trust anchors
    - DS/DNSKEY of zone (should include root)
  - DS → DNSKEY → DS forms a link

DNSSEC Lookup

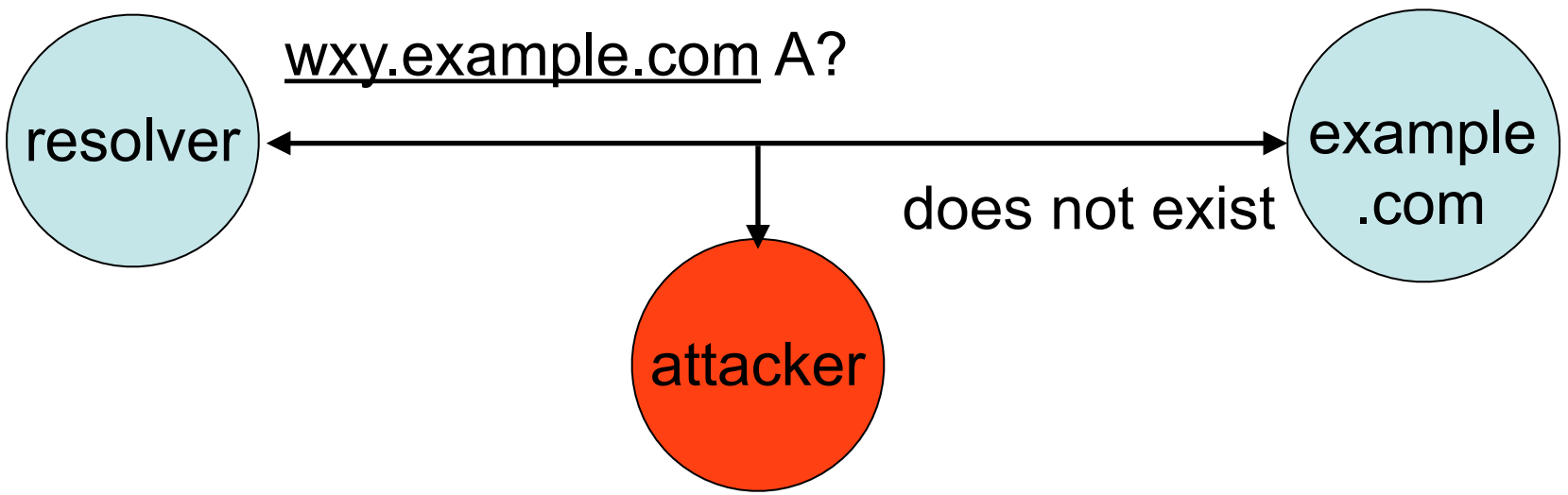


Query: "www.example.com A?"

Reply	RRs in DNS Reply	Added by DNSSEC
3	"com. NS a.gtld.net" "a.gtld.net A 192.5.6.30"	"com. DS" "RRSIG(DS) by ."
5	"example.com. NS a.iana.net" "a.iana.net A 192.0.34.43"	"com. DNSKEY" "RRSIG(DNSKEY) by com." "example.com. DS" "RRSIG(DS) by com."
7	"www.example.com A 1.2.3.4"	"example.com DNSKEY" "RRSIG(DNSKEY) by example.com." "RRSIG(A) by example.com."
8	"www.example.com A 1.2.3.4"	Last Hop?

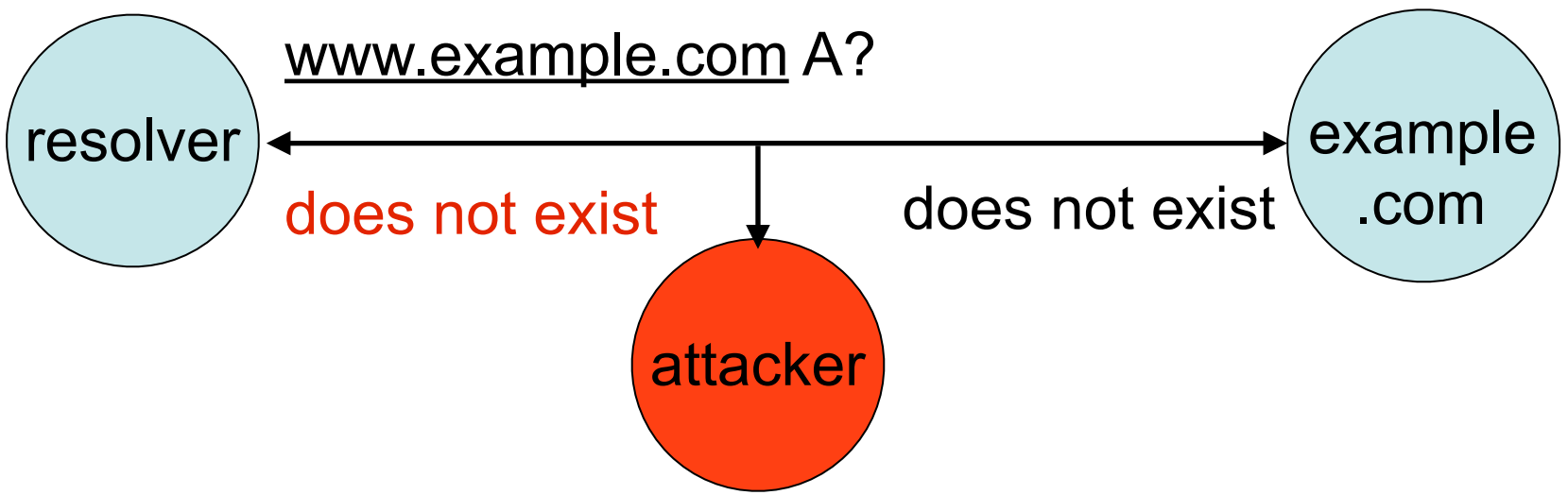
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence



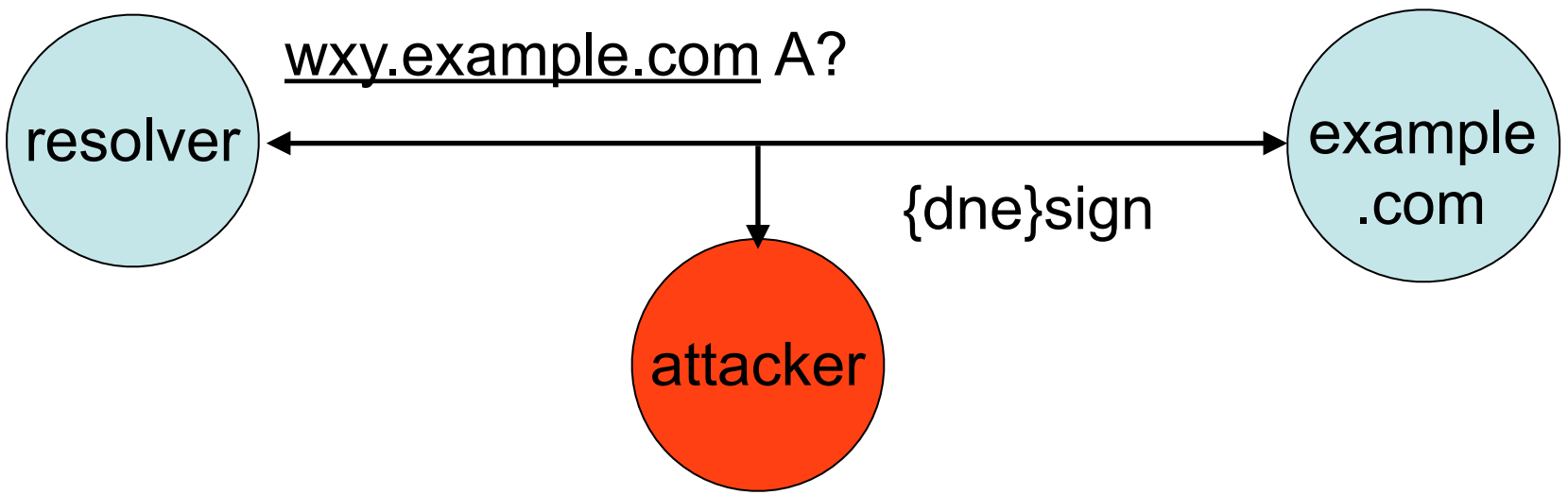
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence



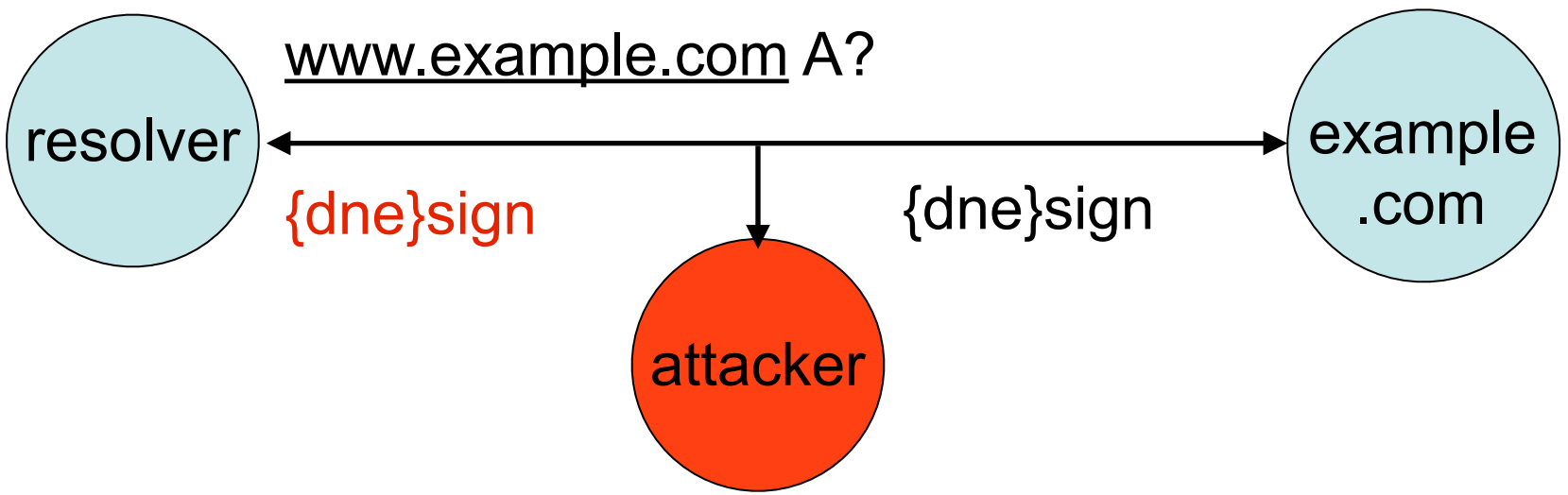
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence



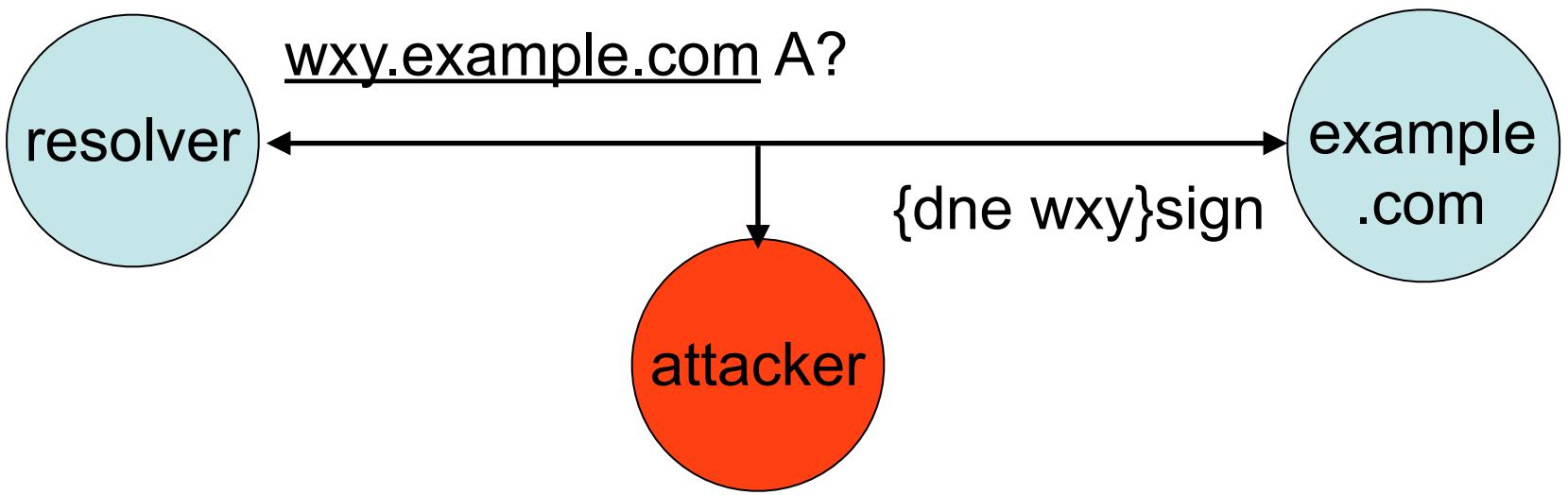
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence



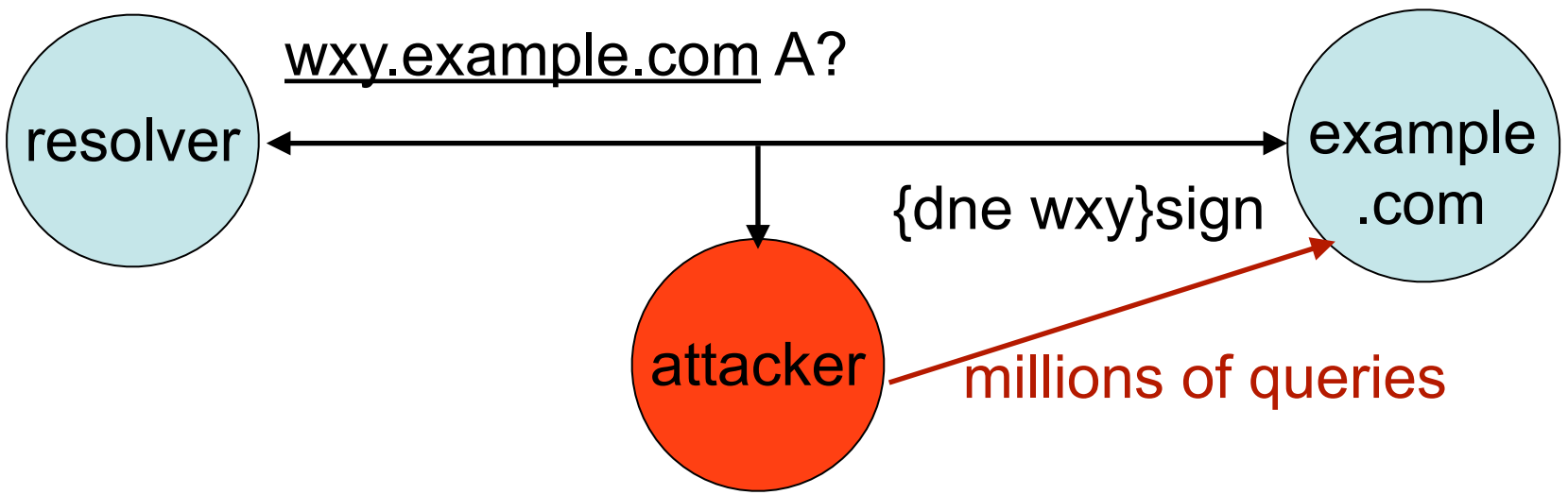
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence



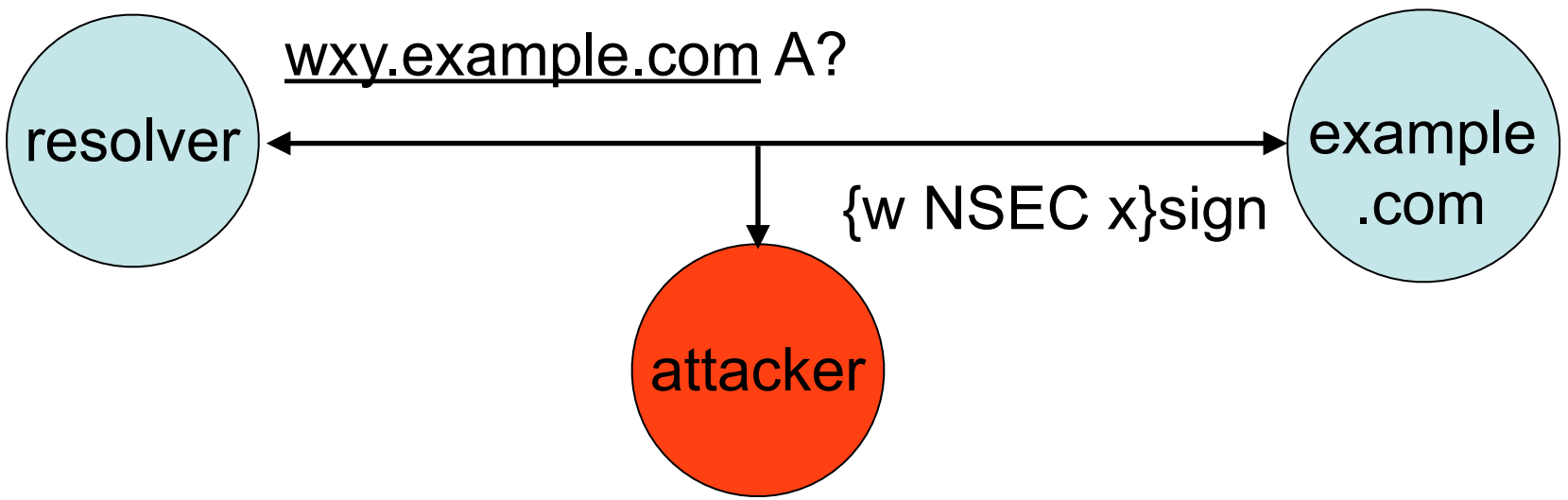
Authenticated Denial-of-Existence

- Most DNS lookups result in denial-of-existence
- Need for offline technique



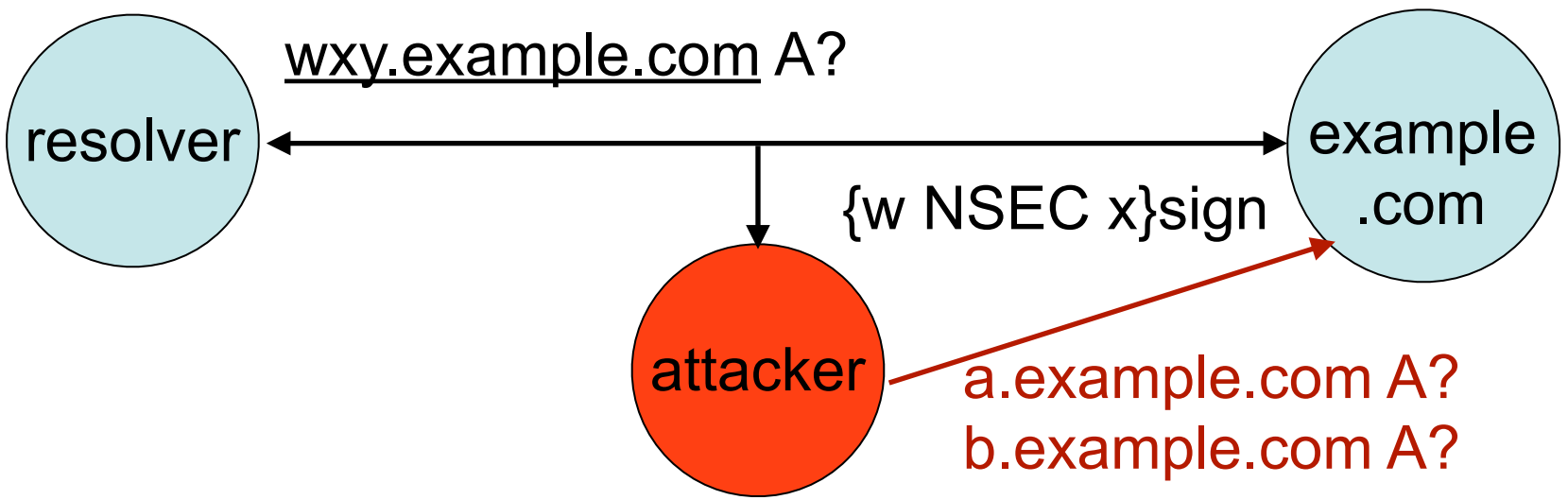
Authenticated Denial-of-Existence

- NSEC scheme



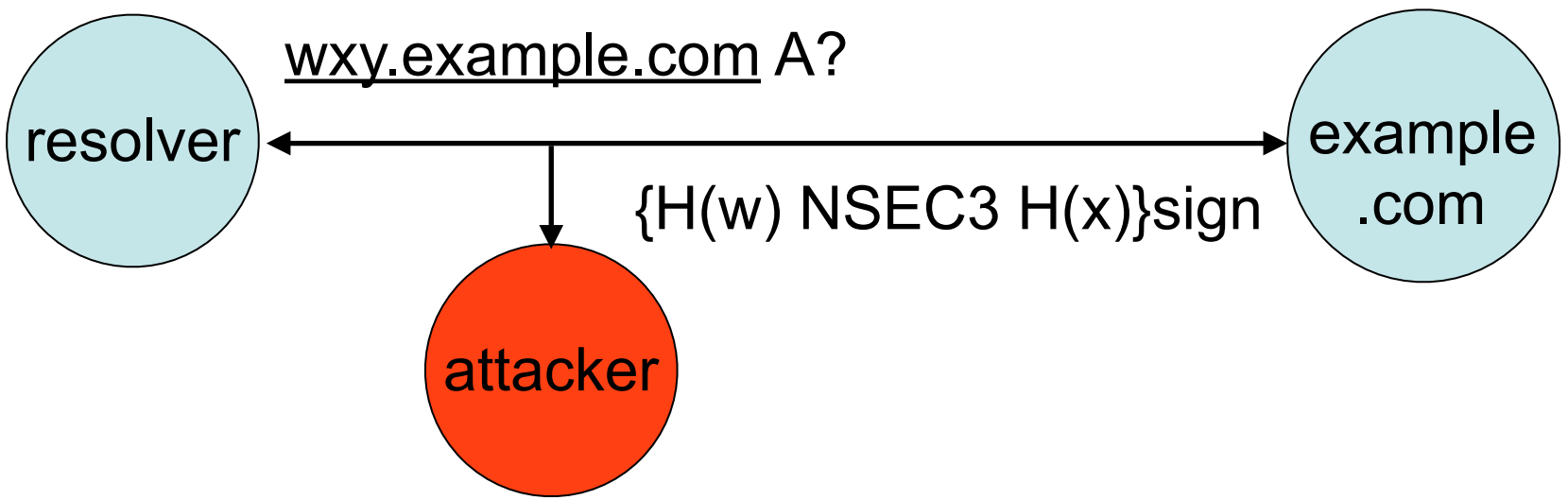
Authenticated Denial-of-Existence

- NSEC scheme



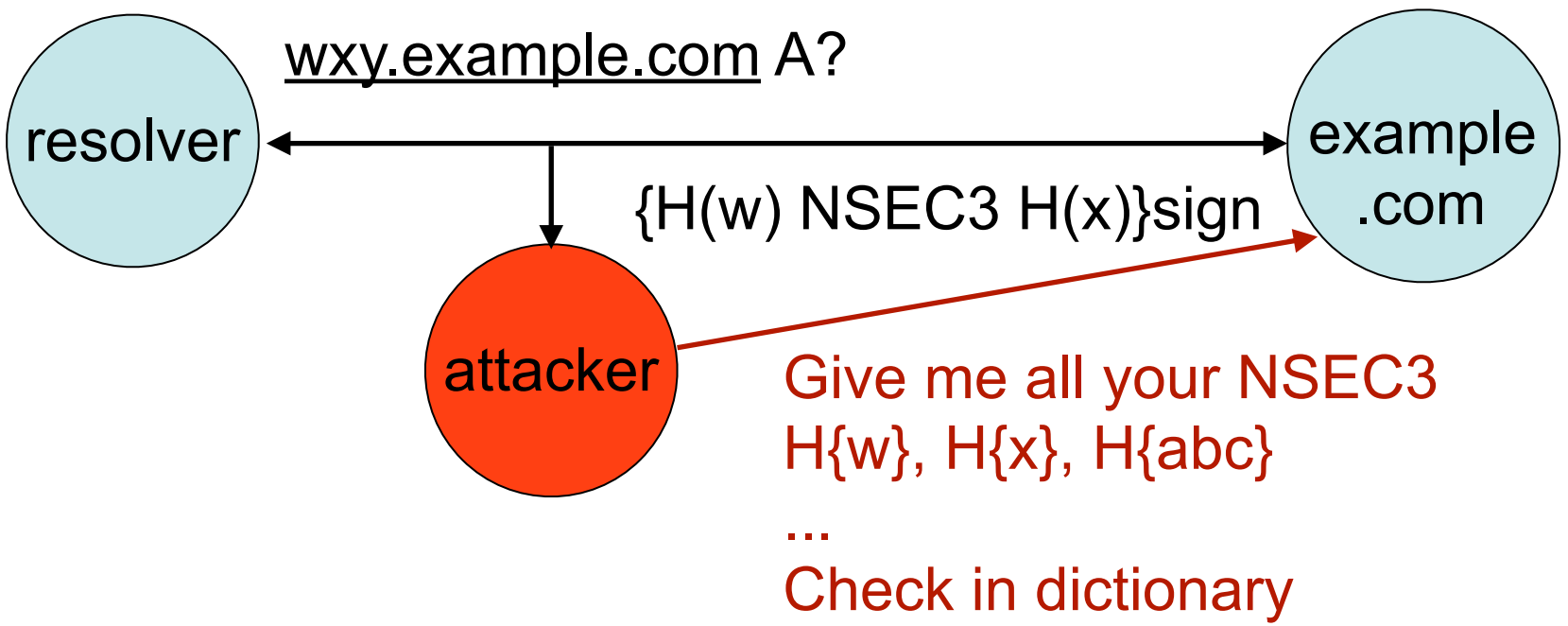
Authenticated Denial-of-Existence

- NSEC3 scheme



Authenticated Denial-of-Existence

- NSEC3 scheme



Authenticated Denial-of-Existence

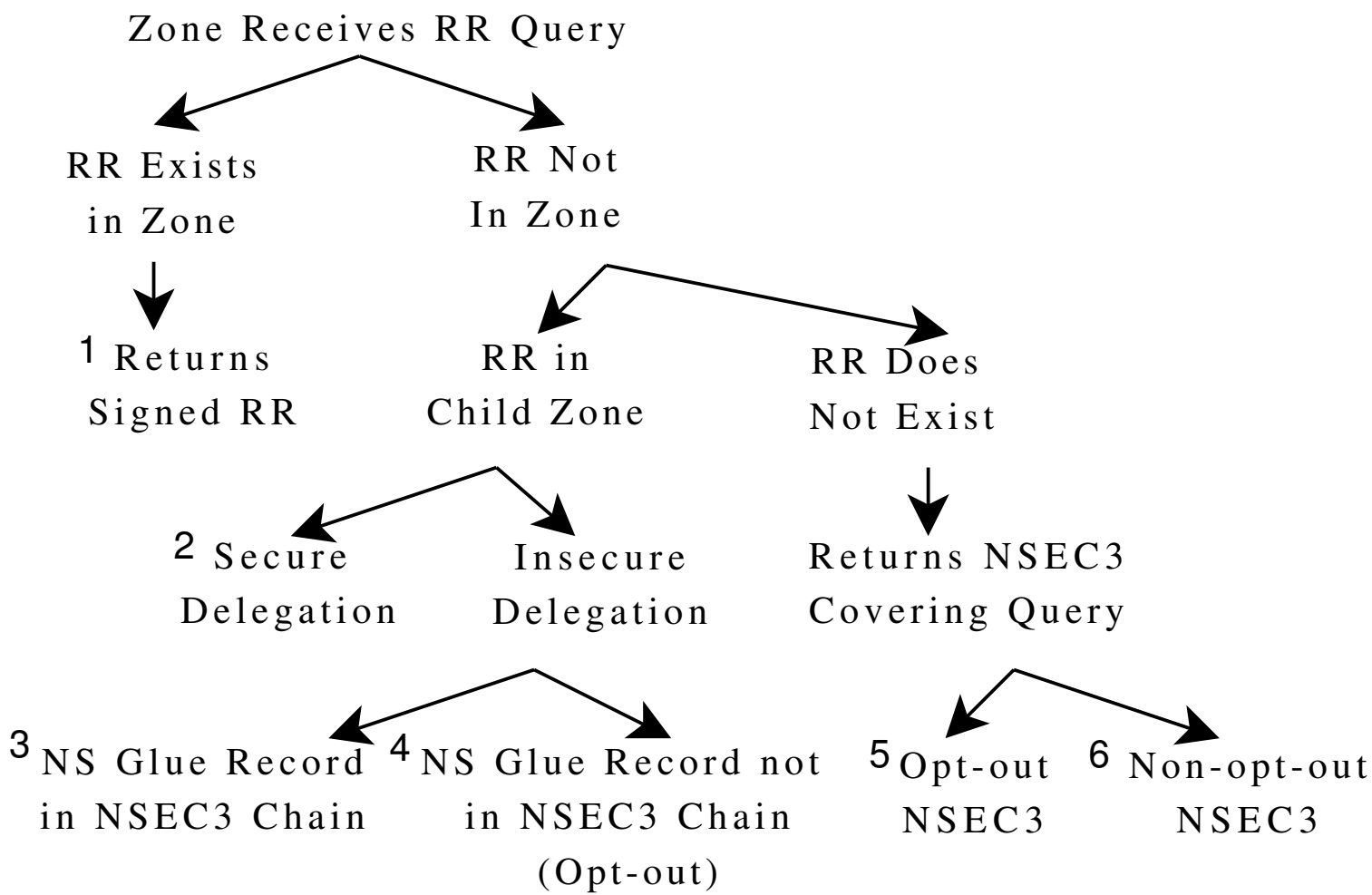
- Most DNS lookups result in denial-of-existence
- Understood mandate of offline-technique

- NSEC (Next SECure)
  - Lists all extant RRs associated with an owner name
  - Points to next owner name with extant RR
  - Easy zone enumeration

- NSEC3
  - Hashes owner names
    - Public salt to prevent pre-computed dictionaries
  - NSEC3 chain in hashed order
  - Opt-out bit for TLDs to support incremental adoption
    - Non-DNSSEC children not in NSEC3 chain

Murφ model

- Typical Usage (query for A RRs), 3 levels of DNSSEC zones
- Six responses from zone to record query
- Resolver queries for each



Attacker model

- All packet manipulation without key compromise
- Record signed RRs contained in packets
  - Add signed RR to packets
  - Delete signed RR from packets
- Create packets with its own signature
- Change unsigned parts
  - All headers
  - Unsigned glue records

Invariants

- “No spoof occurred in location of TLD/Auth server”
- “Attacker key is not valid key for TLD/Auth zone”
- “Accepted answer for [A-F] is correct”
- “Local record valid -> signature chain valid”

```
invariant "Local A or NS record ttl valid -> signature chain valid"
  forall i: LocalId do
    (loc[i].nameA_ttl = VALID | loc[i].nameB_ttl = VALID |
     loc[i].nameC_ttl = VALID | loc[i].nameD_ttl = VALID |
     loc[i].nameE_ttl = VALID | loc[i].nameF_ttl = VALID ) ->
       (!isundefined(loc[i].tld_key) & !isundefined(loc[i].auth_key))
  end;
```

Murφ Results

With full chain-of-trust, signed existent DNSSEC records and non-opt-out denial-of-existence are safe against forgery

- Signed A RR
- Signed DS RR (Secure Delegation)
- Signed Non-opt-out NSEC3

Security Property Violations

- Insecure delegation ↔ opt-out NSEC3
  - Difference is presence of unsigned “glue” RR
  - Denial-of-service
  - RR insertion (Name-prepend)
- Cached record still valid after expiration of attesting RRs
- Delegations can be redirected to attack server
  - Secure: Not exploitable with correct resolver due to DS
  - Insecure

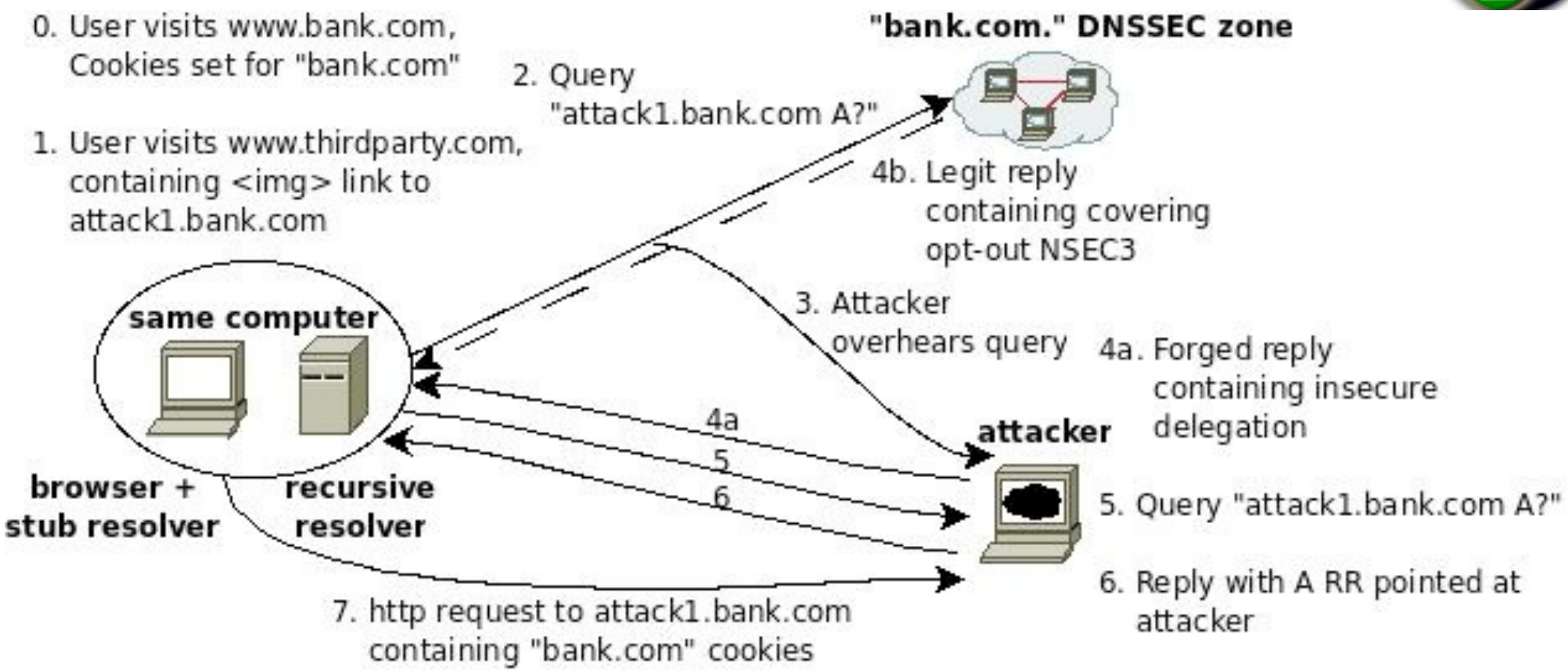
Insecure Sub-Namespace

- NSEC3 Opt-out
  - "Does not assert the existence or non-existence of the insecure delegations that it may cover" - RFC 5155
    - Only thing asserting this is insecure glue records
- Property: Possible to insert bogus pre-pended name into otherwise secure zone. (See RFC 5155)
- Insecure delegation from secure zone
  - Spoofs possible for resultant lookup results
- Acceptable for TLD, bad for enterprises

Cookie-Theft Experiment

- Break security policy dependent on “domain” membership
- Mimic enterprise-level DNSSEC zone
- Zone configured with insecure sub-namespace
  - Prepend false name with
    - NSEC3 opt-out
    - Insecure delegation
- Assume coarse-grain cookie 'domain' setting
  - Common usage: see paypal.com

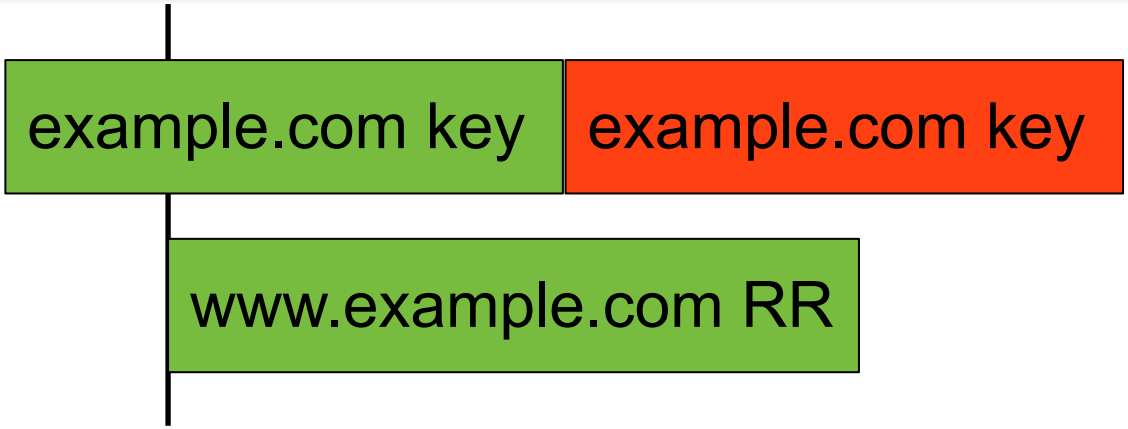
Cookie-Theft Experiment



4b (legit reply):
Opt-out NSEC3 covering "attack1.bank.com"

4a (attack reply):
Opt-out NSEC3 covering "attack1.bank.com" +
"attack1.bank.com. NS ns.atk.com" +
"ns.atk.com A 5.6.7.8"

Chain-of-Trust Expiration



- Chain-of-trust is complete at time of RR entry to cache
- RR can still be valid after an attesting signature expires

- Scenario:
  - "example.com." key compromised
  - Used to sign many RRs with long sig validity and TTL
  - Sig + Signed RR cached at recursive resolver
  - Key compromise discovered, remote zone key "roll-over"

- But signed poisoned records live on in resolver cache

Limiting Exposure Window

- Cap TTL of all cached RRs on lifetime of entire trust chain
- Reacquire expired records from chain
- But, TTL synchronicity may cause unacceptable traffic

- Resolvers cap all TTLs and Signature Validity periods
  - Limit period of exposure for their customers

Attested Cache

- Given network attacker capabilities
  - Change all DNSSEC packet header bits
  - Add recorded RRs / Delete RRs / Mangle bits within RRs
- Authenticating Resolvers must
  - Not trust any header bits
  - Build *attested cache* only with signed RRs with full chain-of-trust
    - Answer user queries only from attested cache
  - Use unsigned glue records only as indications of delegations and pointers to child-zone server addresses
    - These must not enter attested cache
    - Already: CVE-2009-4022

Securing DNSSEC Ecosystem

- For enterprises:
  - Eliminate insecure sub-space of DNSSEC namespace
    - No NSEC3 opt-out
    - No insecure delegations
  - Fine-grained cookie "domain" restriction
- For resolver software:
  - Do not trust any header bits in replies
  - Only provide user-answers from *attested cache*
  - Periodically re-check validity of cache contents?
- For resolver operators
  - Set artificial cap on TTL(< authoritative zone spec)
  - Provide secure last-hop channel
- For end-user software
  - Provide UI indicator of lookup security
  - Provide secure last-hop channel

Requirements for DNS Security

1. Authoritative zone: sign RRs with DNSSEC
2. Authoritative zone: do not use NSEC3 opt-out
3. Authoritative zone: no insecure delegations
4. High-level zones (root and TLD): sign and provide secure delegation
5. ISPs: Adopt DNSSEC in recursive resolver
6. ISPs+OS: Support secure channel in the last-hop between stub and recursive resolvers
7. Applications: Interface indicators of DNS lookup security

- Without all of these, no single party benefits from DNSSEC
- Perhaps explains long process of DNSSEC adoption
- Momentum is building, however

Conclusions

- DNSSEC / NSEC3 Model checking study
  - Some interesting security property violations
  - All can be mitigated by protocol/implementation config
  - Provided best-practice configuration