Network Security

AA 2020/2021

Security Protocols

Examples

- WLAN Security
- IPSec
- DNS

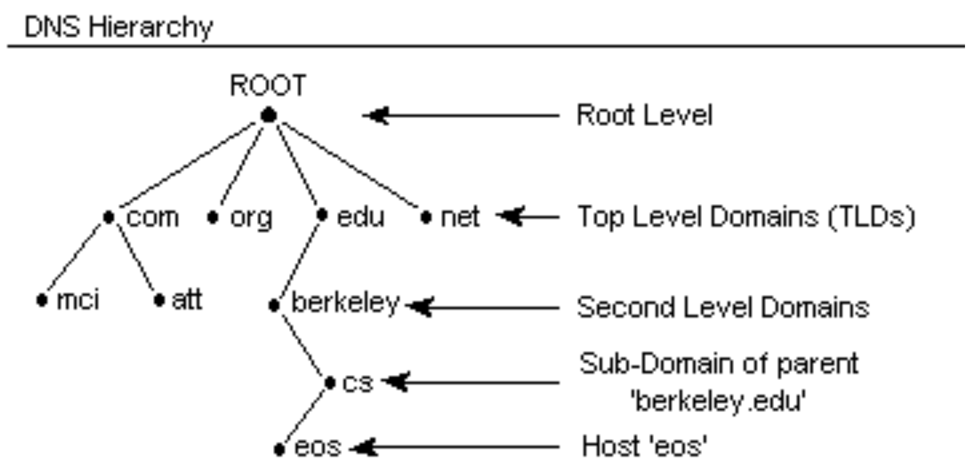
UNIVERSITÀ DEGLI STUDI
DI TRENTO

DNS: Domain Name Service

Domain Name Service (quick intro)

- DNS is a hierarchical system for domain name resolving
  - Translates human-readable addressed (google.com) to (a set of) IP addresses the domain is reachable at
  - UDP for fast answers (port 53)
- Each transaction identified by an ID (16 bits)
  - Query ID: “QID”
  - Original DNS implementation → incremental QID
- Several types of records. Of interest here
  - A (AAAA) → IPv4 (IPv6) of the requested domain
    - e.g. a.website.com A 65.61.198.201
  - NS → IP of the DNS server to ask
    - e.g. a.website.com NS ns.website.com
    - Followed by an A answer for the dns
      - ns.website.com A 2.2.2.2

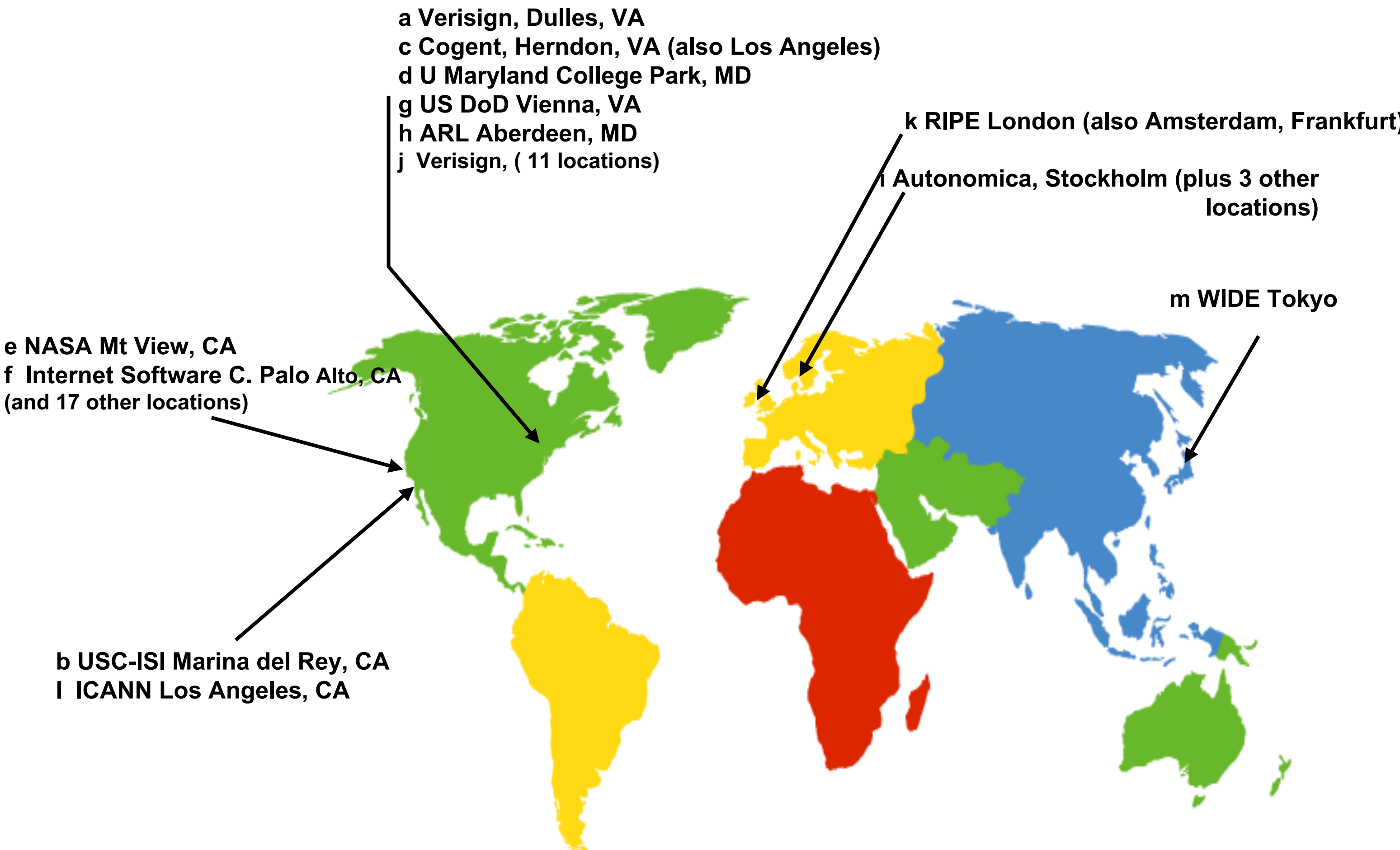
DNS hierarchy



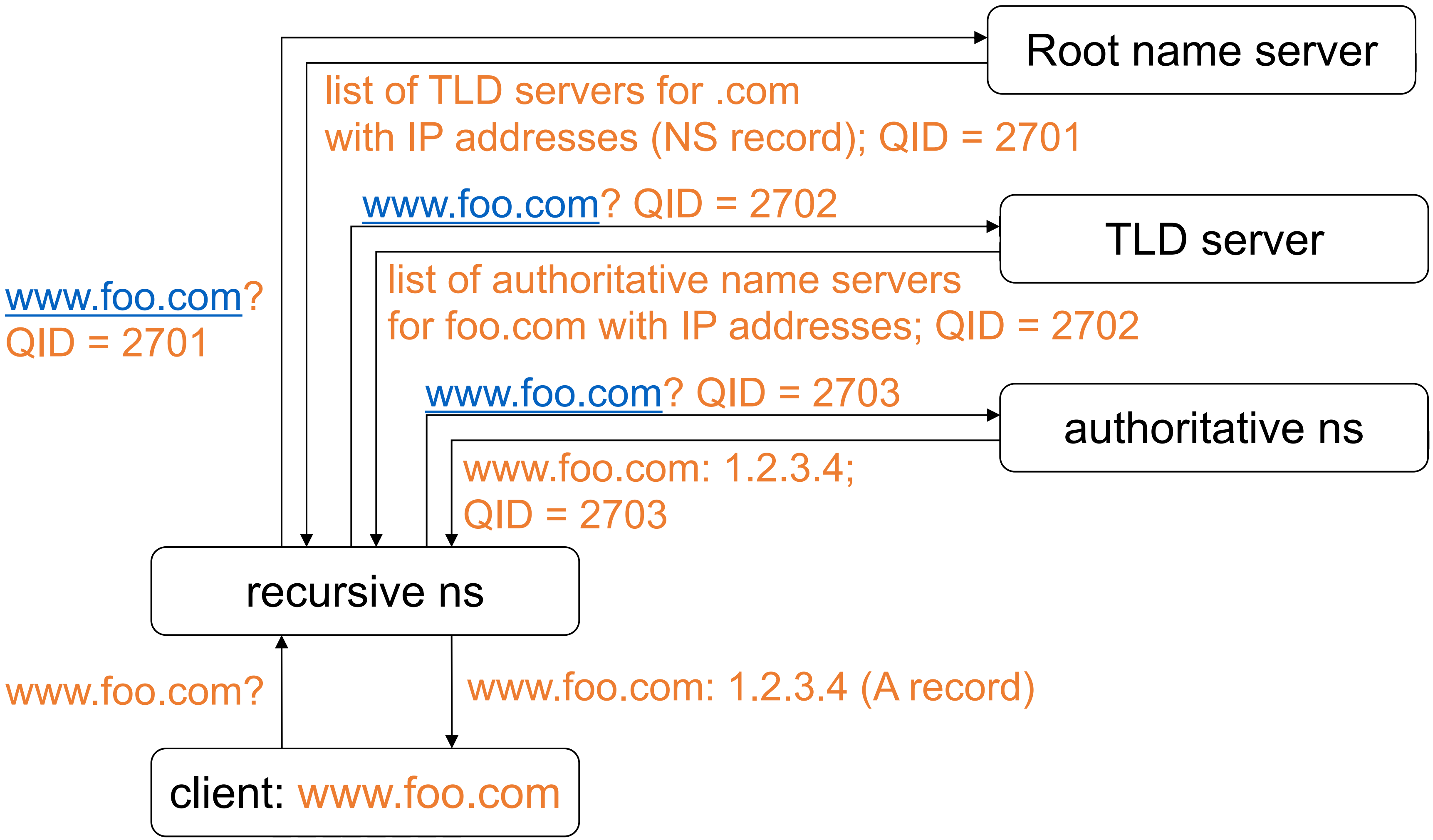
- Root DNSs → responsible for top level domain queries
  - e.g. .com NS ns.auth.net
- Authoritative DNS → a DNS server that answers queries whose answer it already knows
  - Does not ask to other DNSs
- Recursive DNS → a DNS server that forwards queries to Authoritative DNSs

DNS root servers location

http://root-servers.org



Name resolution



Queries

- recursive DNS query:

DNS client queries and DNS server respond with either the requested resource record, or an error message stating that the record or domain name does not exist.

- Iterative DNS query:

contacted server replies with name of server to contact “I don’t know this name, but ask this server XXX”

Cache & Time-to-live

- Simplified description left out an important aspect.
- Performance optimisation: when name server receives an answer, it stores answer in its cache.
- When receiving a request, name server first checks whether answer is already in its cache; if this is the case, the cached answer is given.
- Answer remains in cache until it expires; time-to-live (TTL) of answer is set by sender.
- Design question: reasons for setting TTL by sender, reasons for setting TTL by receiver?
- Long TTL = high security, low TTL = low security?

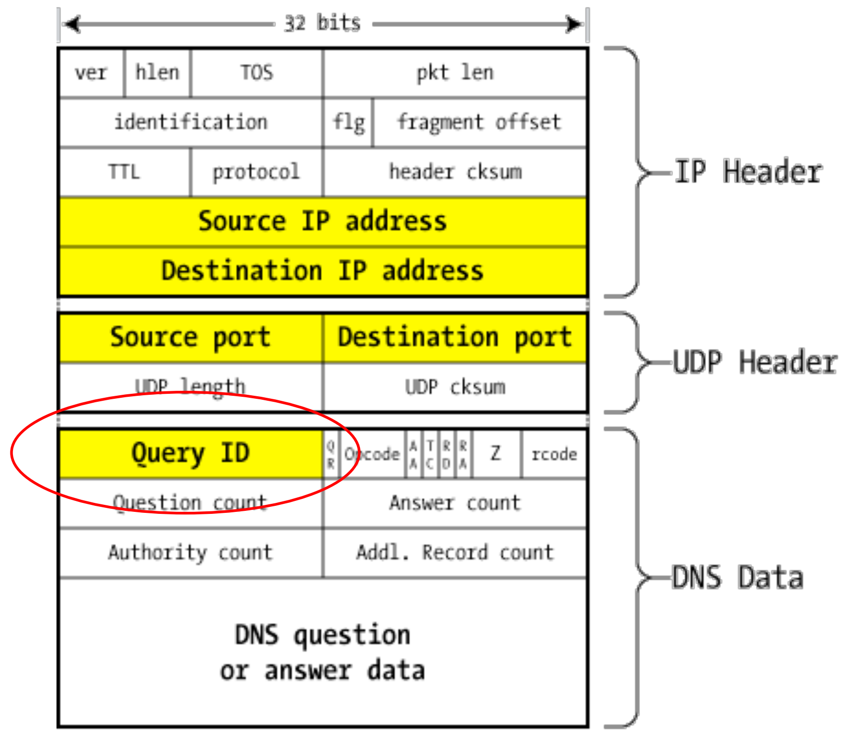
Light-weight Authentication

- Anybody can pretend to be an authoritative name server for any zone.
- How does a recursive name server know that it has received a reply from an authoritative name server?
- Recursive name server includes a 16-bit query ID (QID) in its requests.
- Responding name server copies QID into its answer; applies also to answer from authoritative name server.
- Recursive name server caches first answer for a given QID and host name; then discards this QID.
- Drops answers that do not match an active QID.

Compromising Authentication

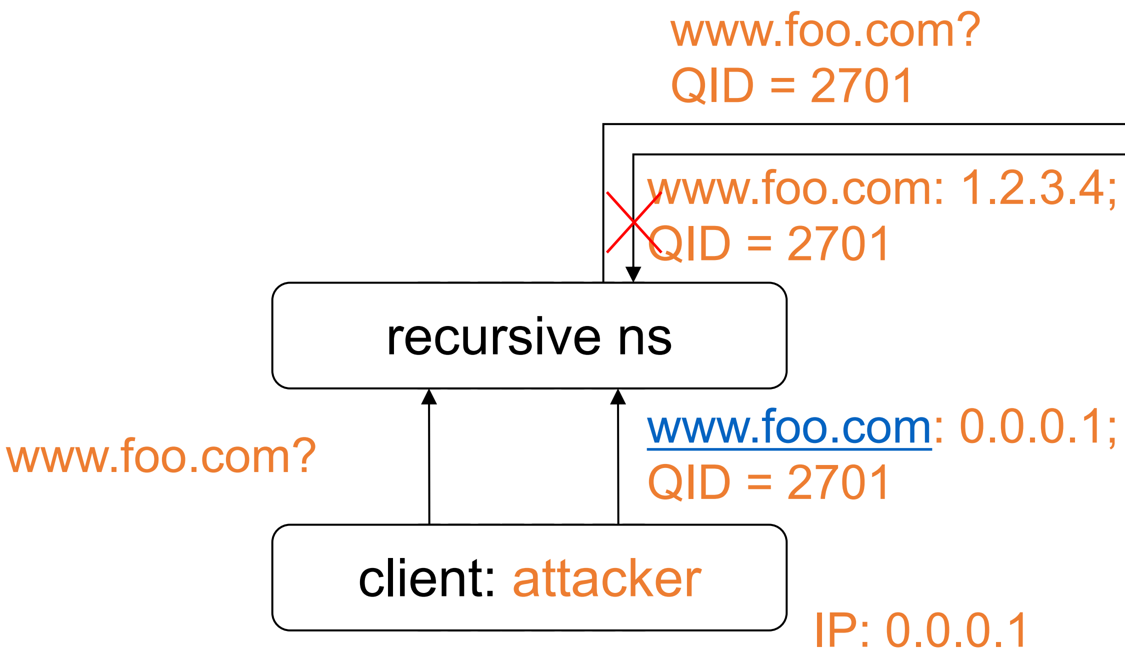
- If routing to and from root servers and TLD servers cannot be compromised, the attacker can only try to improve her chances of guessing a query ID.
- Some (earlier) versions of BIND used a counter to generate the QID.
- Cache poisoning attack:
  1. Ask recursive name server to resolve host name in attacker's domain.
  2. Request to attacker's name server contains current QID.
  3. Ask recursive name server to resolve host name you want to take over; send answer that includes next QID and maps host name to your chosen IP address.
  4. If your answer arrives before the authoritative answer, your value will be cached; the correct answer is dropped.

Cache poisoning attack



DNS packet on the wire

Cache poisoning attack



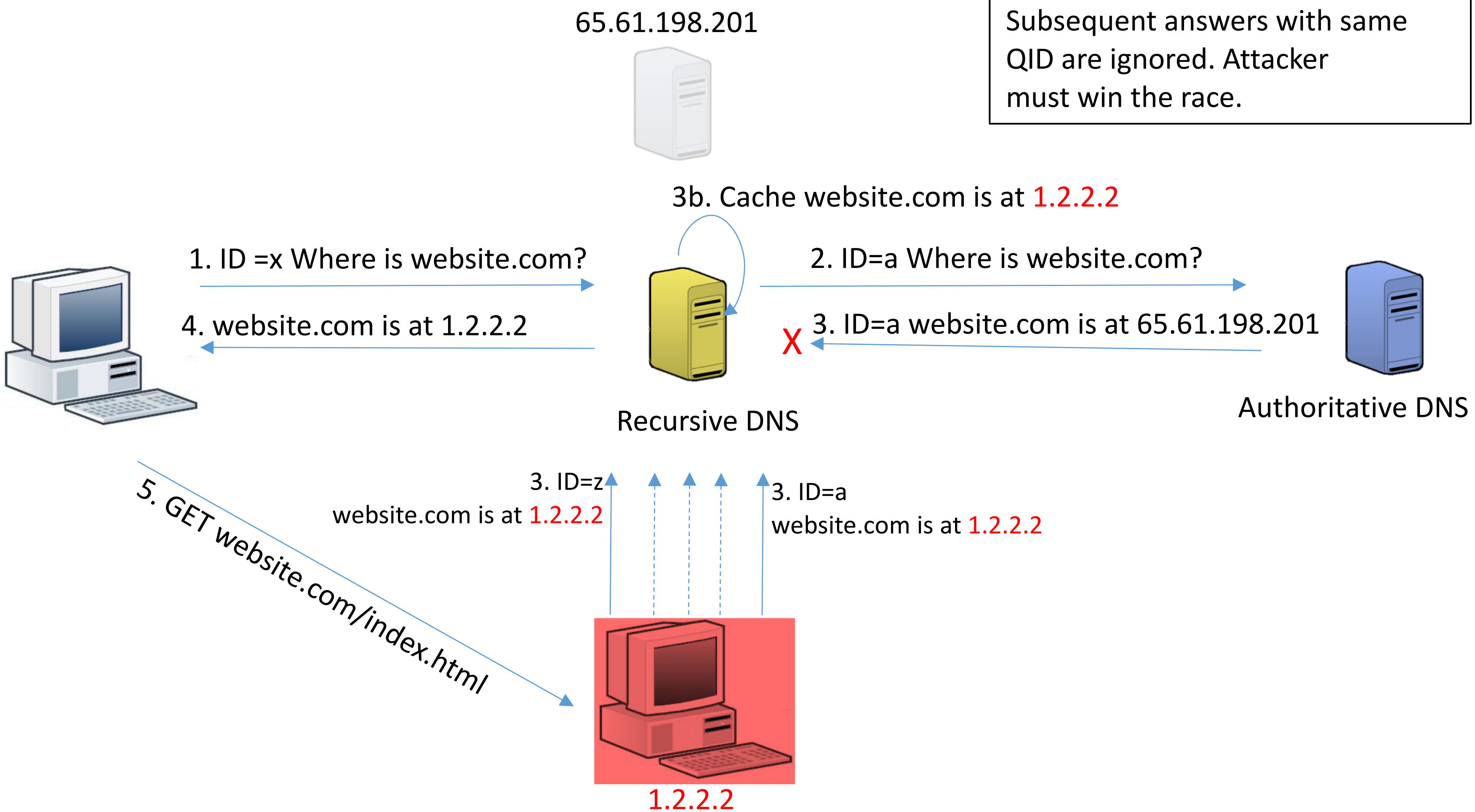
Predictable Challenges

- Lesson: If you want to perform authentication without cryptography, do not use predictable challenges.
- More ways of improving the attack's chances:
  - To account for other queries to the recursive name server concurrent to the attack, send answers with QIDs from a small window.
  - To increase the chance that fake answer arrives before authoritative answer, slow down authoritative name server with a DoS attack.
  - To prevent that a new query for the host name restores the correct binding, set a long time to live.

DNS cache poisoning

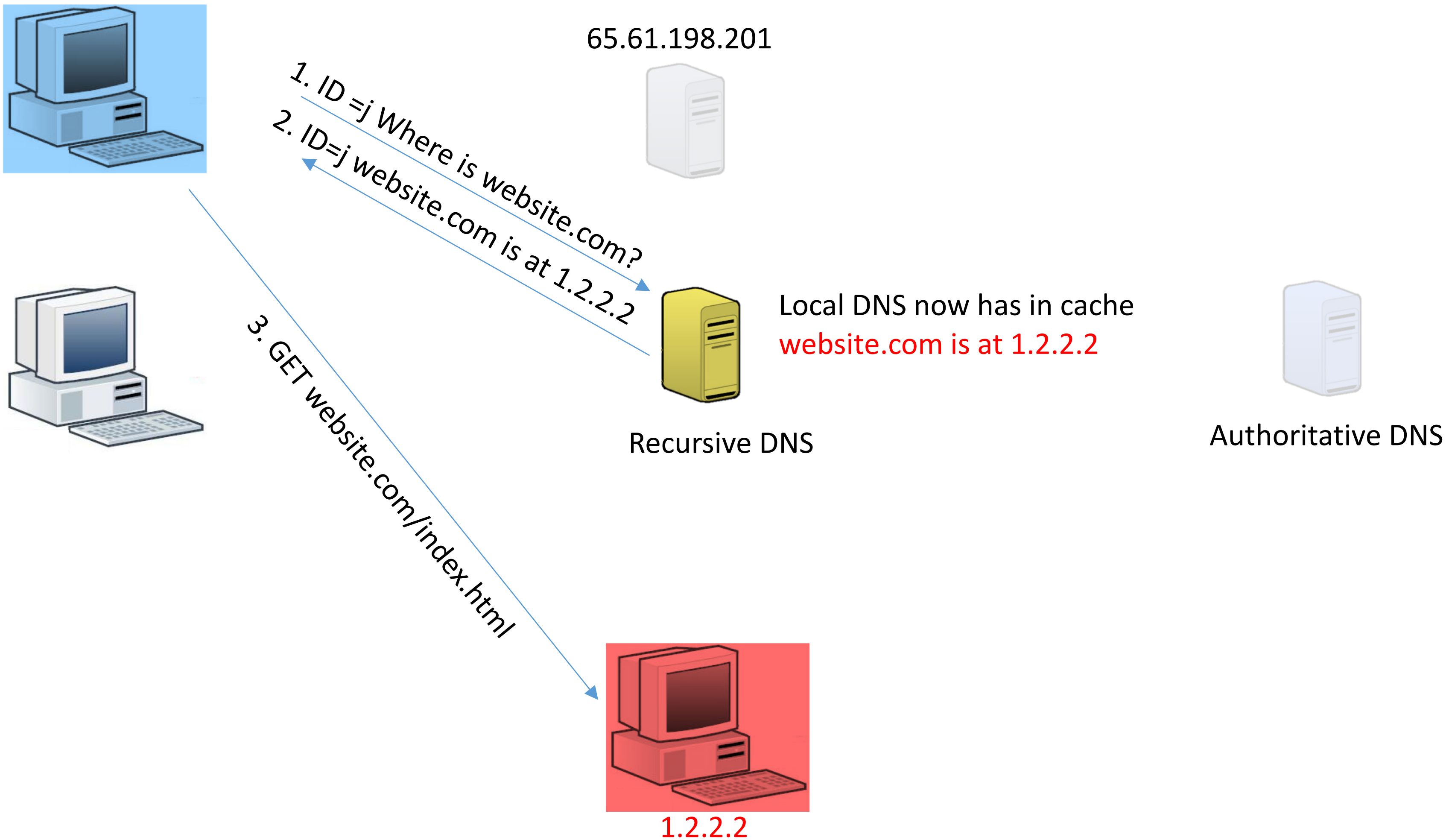
Recursive DNS' cache:
website.com A 1.2.2.2

The first received answer is cached
Subsequent answers with same
QID are ignored. Attacker
must win the race.



DNS cache poisoning

Recursive DNS' cache:
website.com A 1.2.2.2



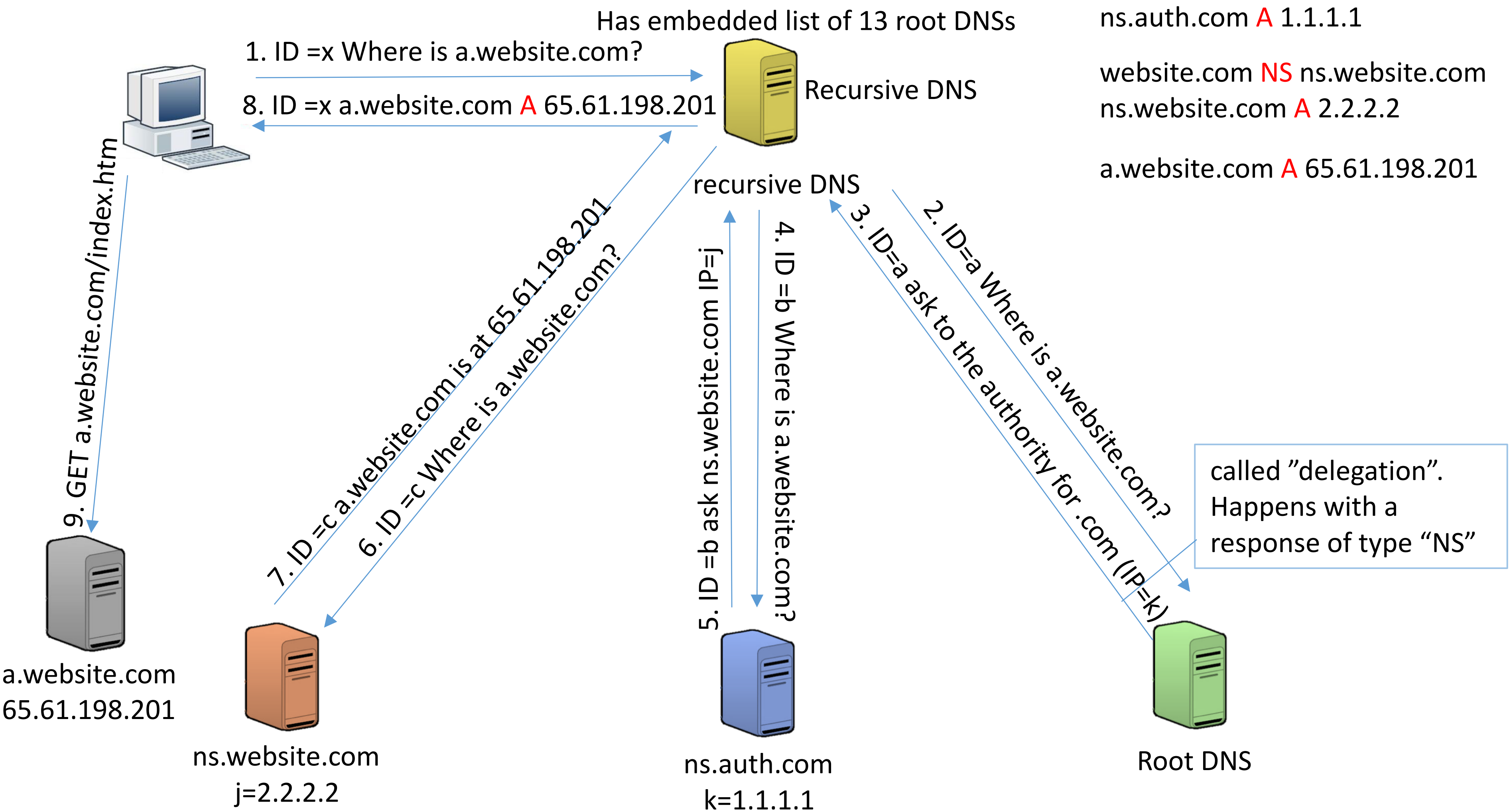
DNS, the full picture

Recursive DNS' cache:

.com NS ns.auth.com
ns.auth.com A 1.1.1.1

website.com NS ns.website.com
ns.website.com A 2.2.2.2

a.website.com A 65.61.198.201



Kaminsky vulnerability

- The Kaminsky vulnerability can lead to a cache poisoning attack
- The attacker rather than replacing an A record replaces an NS record
- This way the attacker can get control over any (sub)domain
  - b.a.website.com
  - a.website.com
  - website.com
  - .com

Kaminsky attack (cntd)

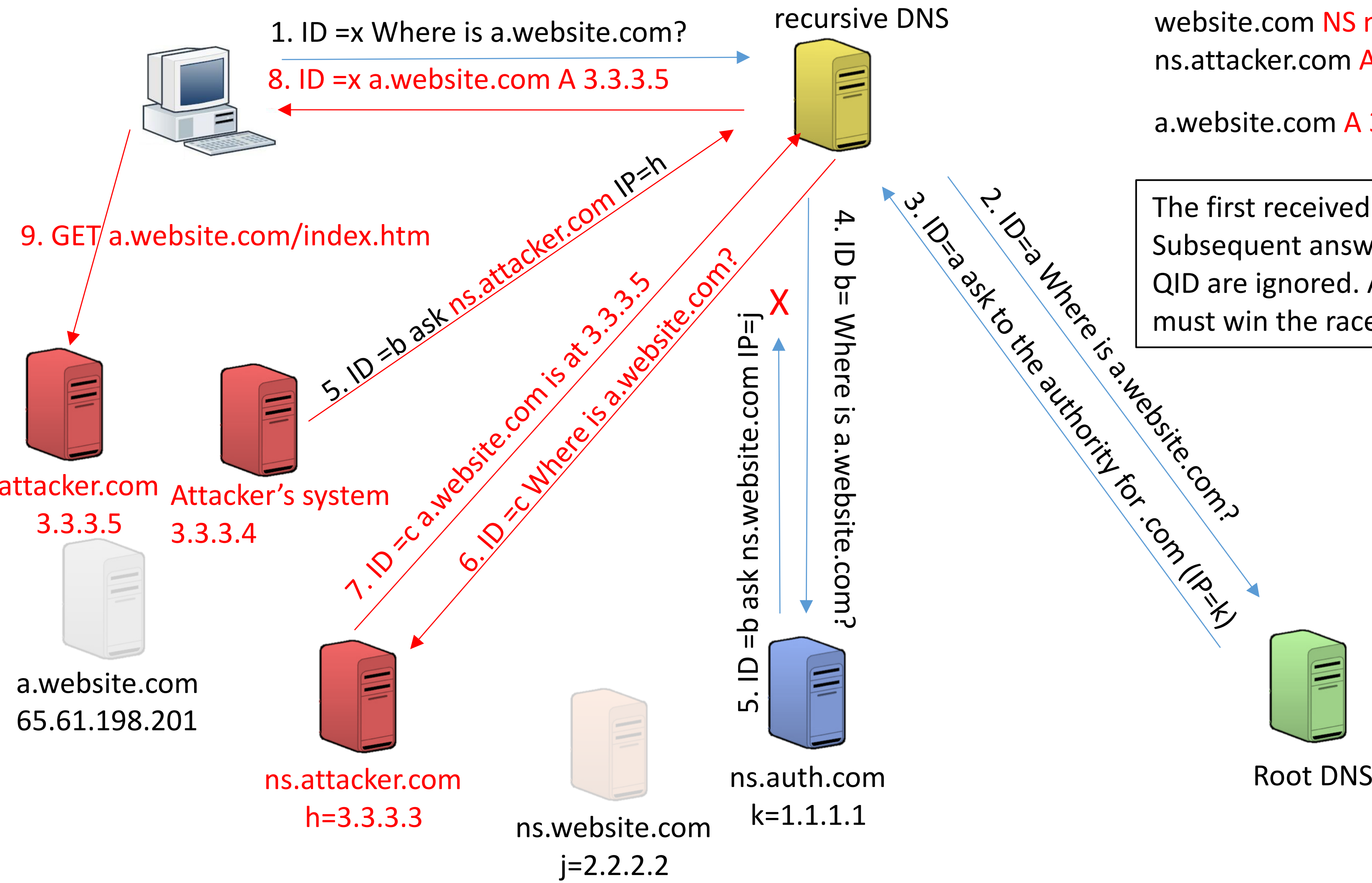
Recursive DNS' cache:

.com NS ns.auth.com
ns.auth.com A 1.1.1.1

website.com NS ns.attacker.com
ns.attacker.com A 3.3.3.3

a.website.com A 3.3.3.5

The first received answer is cached
Subsequent answers with same QID are ignored. Attacker must win the race.



Kaminsky attack (cntd)

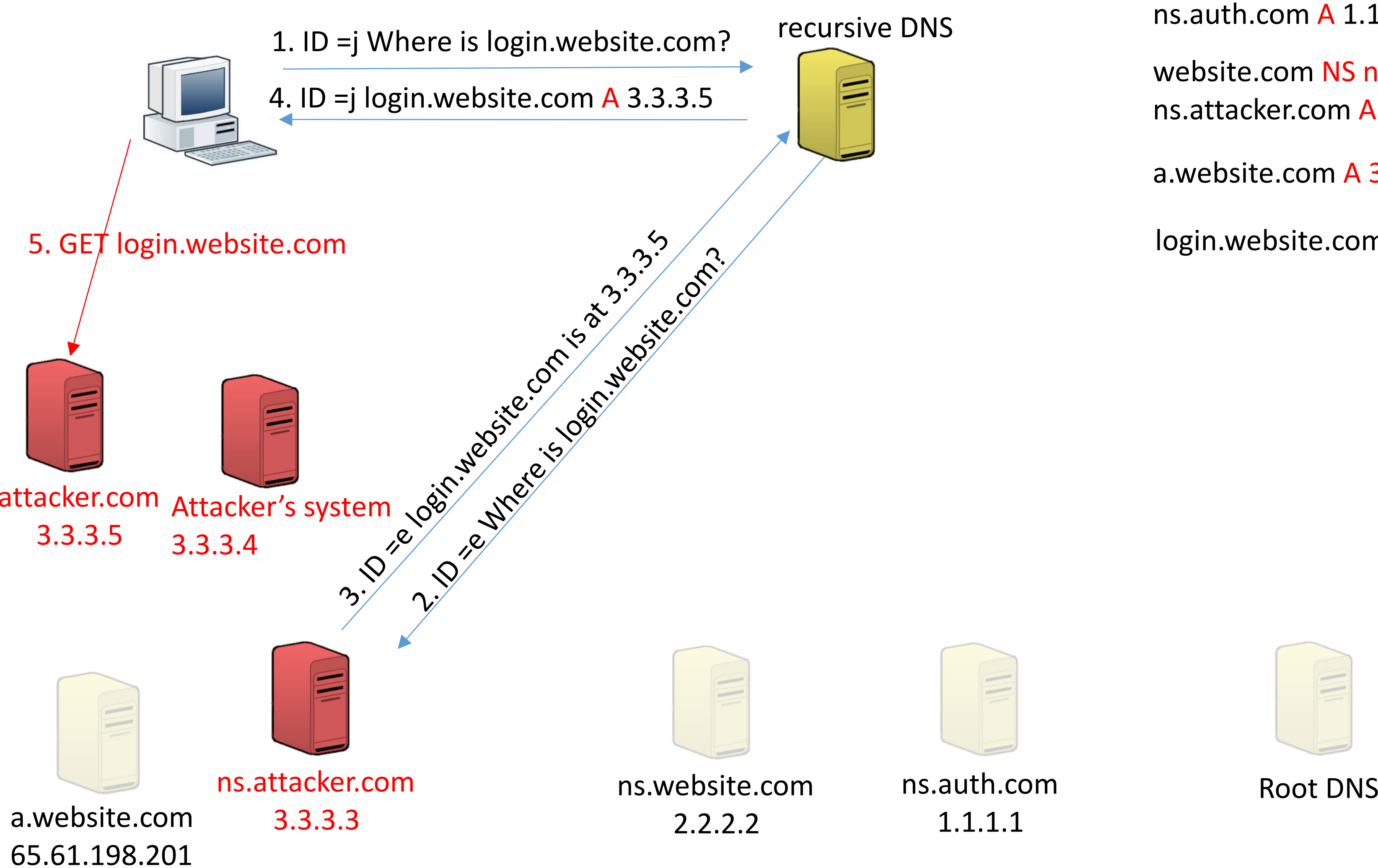
Recursive DNS' cache:

.com NS ns.auth.com
ns.auth.com A 1.1.1.1

website.com NS ns.attacker.com
ns.attacker.com A 3.3.3.3

a.website.com A 3.3.3.5

login.website.com A 3.3.3.5

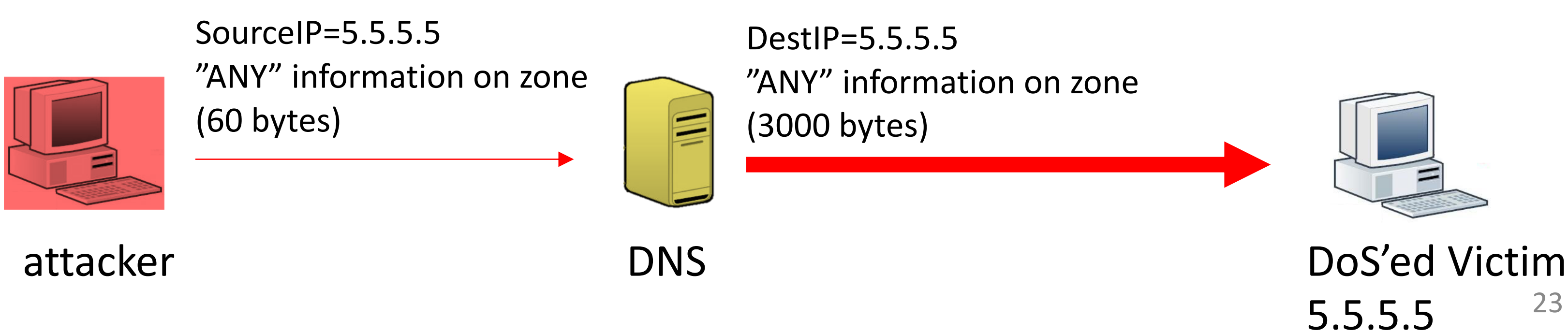


Mitigation of Kaminsky's vulnerability

- Source of attack is low entropy with a 16 bit ID
  - Randomness is not enough to represent a significant margin
  - Moving ID size to 32 bits is not feasible
    - Can not change the protocol
- Solution → randomize the source port (16 bits) to increase entropy
  - In reality can't use all 16 bits for the source port because of reserved values
  - Any answer that does not match **both** source port and transaction ID will be dropped

DNS amplification attack

- A type of DoS attack
- Exploits certain type of DNS answers that are much bigger in size than the requests
  - attack's throughput much bigger than attacker's input
- DNS works over UDP → source IP easy to spoof



DNS zone transfer

- A zone is a domain for which a server is authoritative
- “slave” servers can ask “authoritative” servers to copy their zone database
  - Over TCP
- An attacker pretends to be a slave server and dump the zone DB
  - Acquires knowledge of zone’s infrastructure
  - Can be used to facilitate further attacks (e.g. spoofing or more direct attacks)

DNSSec

- Secure implementation of the DNS protocol
- Implements DNS authentication on top of normal DNS exchange
  - Digitally signed over a *chain-of-trust* starting from the root server
  - Uses electronic certificates
    - Public-key crypto → authenticate by showing proof that you own a secret key
- Protects data integrity
  - No confidentiality protection
- Additional reading
  - Hao Yang ; Osterweil, E. ; Massey, D. ; Songwu Lu ; Lixia Zhang. Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC. *IEEE Transactions on Dependable and Secure Computing. Vol 8, Issue 5. 2010*

Common issues

- Most of the network attacks we've seen so far have at least one of two issues common among most network problems
  - Lack of authentication → the real sender/receiver of a packet/datagram can not be authenticated
    - It is possible to spoof its identity
  - Communication channel is in the clear → a clever or well-positioned (in the network) attacker can read and potentially modify the information exchanged over the channel
    - Confidentiality problem that becomes an authentication problem

→ Cryptography helps **mitigating** many of these problems

Suggested reading

- Bykova, Marina, and Shawn Ostermann. "Statistical analysis of malformed packets and their origins in the modern Internet." *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM, 2002.
- Hao Yang ; Osterweil, E. ; Massey, D. ; Songwu Lu ; Lixia Zhang. Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC. *IEEE Transactions on Dependable and Secure Computing. Vol 8, Issue 5. 2010*
- Internet Census 2012. Port scanning /0 using insecure embedded devices.
  - http://internetcensus2012.bitbucket.org/paper.html
- Blackert, W. J., et al. "Analyzing interaction between distributed denial of service attacks and mitigation technologies." *DARPA information survivability conference and exposition, 2003. Proceedings*. Vol. 1. IEEE, 2003.
- S. M. Bellovin. 1989. Security problems in the TCP/IP protocol suite. *SIGCOMM Comput. Commun. Rev.* 19, 2 (April 1989), 32-48. DOI=http://dx.doi.org/10.1145/378444.378449