Imperial College London

CO331 – Network and Web Security

9. DNS

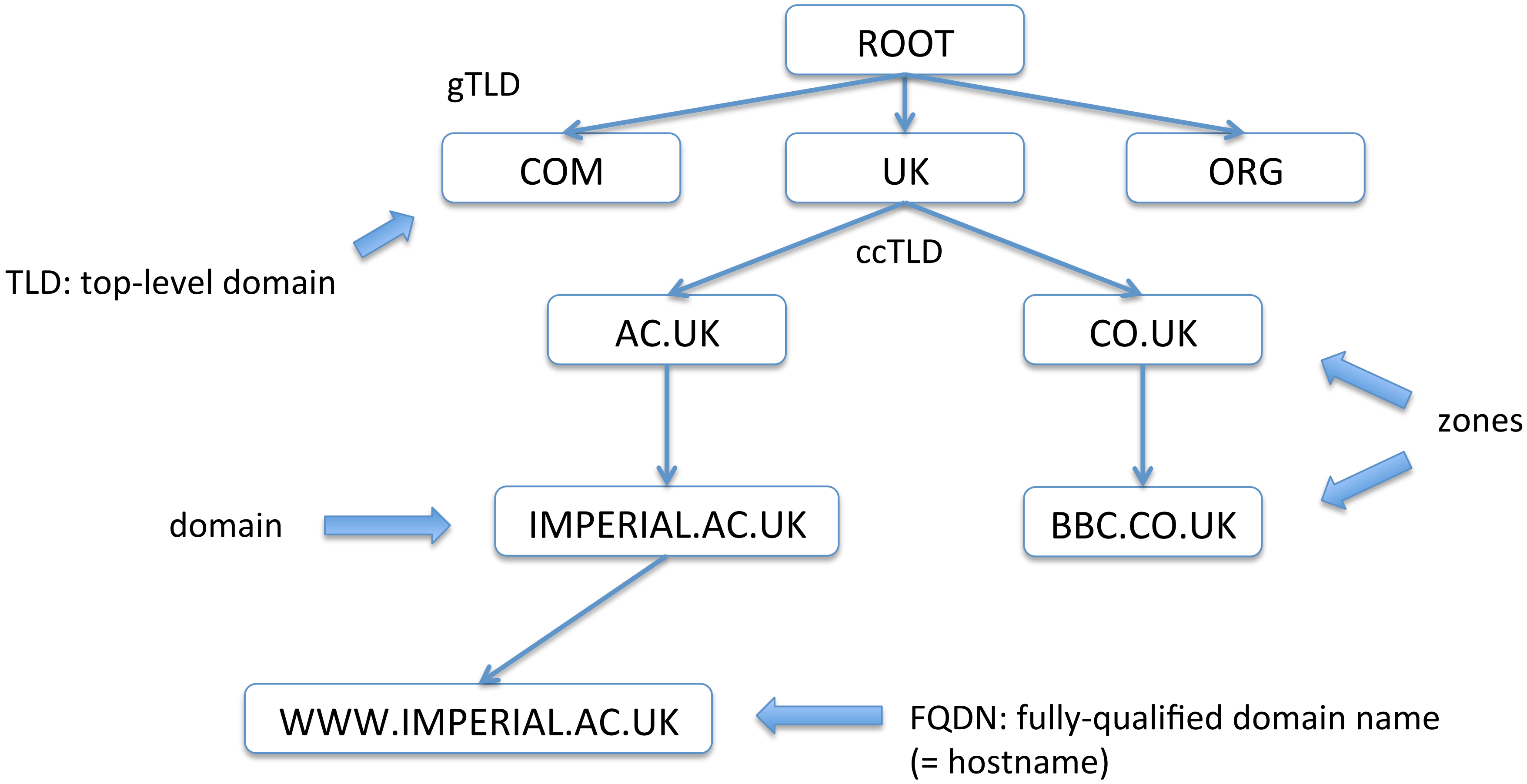
Dr Sergio Maffeis
Department of Computing
Course web page: http://www.doc.ic.ac.uk/~maffeis/331

Domain Name System

Imperial College London

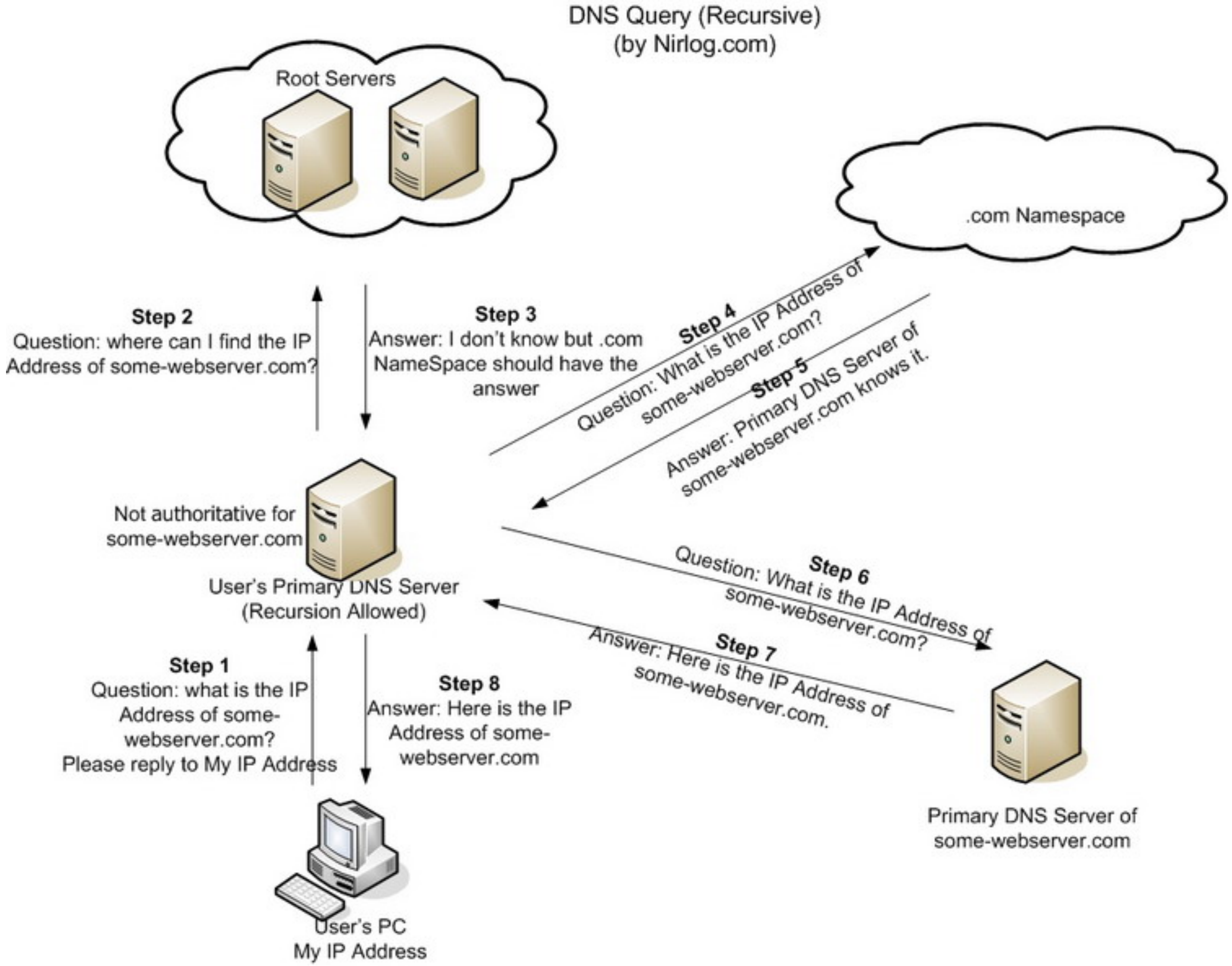
- The Domain Name System (DNS) lets us identify hosts via *hostname* instead of IP address
  - `www.imperial.ac.uk` instead of `155.198.140.14`
  - Hostnames are easy to remember, descriptive of service or brand
  - The DNS separates the logical address of a service from the physical address of the host running that service
    - Hostname does not need to change as we switch network provider
- *DNS Resolution*
  - Before creating an IP packet, a local *DNS client* (or *resolver*) looks up the IP address of the target hostname
    - Often the result is in the local cache
    - Otherwise, the resolver queries an external *primary* (or *recursive*) *DNS server*
  - Normal DNS traffic is sent over UDP
    - Typical queries and responses are small and fit in 1 UDP packet (512 bytes)
    - When more data needs to be exchanged, DNS falls back to TCP
- Domain names are organized hierarchically
  - DNS is managed by ICANN/IANA, which runs the root DNS servers

Domain Name System



DNS resolution

Imperial College London



Common DNS records

Imperial College London

Resource Record	Description
SOA (Start of Authority)	Indicates that the server is the best authoritative source for data concerning the zone. Each zone must have an SOA record, and only one SOA record can be in a zone.
NS (Name Server)	Identifies a DNS server functioning as an authority for the zone. Each DNS server in the zone (whether primary master or secondary) must be represented by an NS record.
A (Address)	Provides a name-to-address mapping that supplies an IPv4 address for a specific DNS name. This record type performs the primary function of the DNS: converting names to addresses
AAAA (Address)	Provides a name-to-address mapping that supplies an IPv6 address for a specific DNS name. This record type performs the primary function of the DNS: converting names to addresses.
PTR (Pointer)	Provides an address-to-name mapping that supplies a DNS name for a specific address in the in-addr.arpa domain. This is the functional opposite of an A record, used for reverse lookups only.
CNAME (Canonical Name)	Creates an alias that points to the canonical name (that is, the "real" name) of a host identified by an A record. Administrators use CNAME records to provide alternative names by which systems can be identified.
MX (Mail Exchange)	Identifies a system that will direct email traffic sent to an address in the domain to the individual recipient, a mail gateway, or another mail server.

DNS MITM attack

- Turkish government wanted to block Twitter access in March 2014
- Forced ISPs to respond to DNS queries for twitter.com with the IP of a government website
  – Effectively the ISP DNS resolvers launched a MITM attack on link between user and public DNS servers
- Once it became obvious, users got around restriction using Google's Public DNS

DNS security issues

Imperial College London

- DNS requests and responses are not authenticated
  - MITM or compromised DNS can map trusted domain names to malicious IPs
  - *DNS cache poisoning* (see recommended reading)
    - Off-path attacker can poison cache of honest DNS server
  - *DNS rebinding* (we'll see example later in the course)
- *DNSSEC* improves the security of DNS
  - Protects authenticity and integrity of DNS records
    - Each DNS zone has public/private key-pairs
      - Chain of trust starts at DNS root (https://www.iana.org/dnssec)
    - Private key is used to sign zone data
    - Public key is used by others to verify signature
  - DANE: DNSSEC data used to improve TLS certificate infrastructure
    - Domain owner can deploy trusted self-signed certificates
    - Possible to restrict acceptable CA or certificate for a domain
    - Trust moves from CAs to DNS operators
  - Weaknesses
    - Increased load on DNS servers (due to crypto)
    - Decreased network performance (longer records, over TCP)
    - *Zone enumeration* information leakage (see next slide)

DNSSEC zone enumeration

- If a domain does not exist, an NSEC record reveals alphabetically-closest neighbors
  - Failed query: “resolve bob.example.com”
  - Response: “no records exist between alice.example.com and charlie.example.com”
- NSEC is useful to *prove* that the domain does not exist
  - No further DSN queries are necessary
- Problem: this helps hacker’s intelligence gathering activities
  - Find out which domains don’t exist (bob) and discover “closest” ones (alice, charlie)
  - Target scanning activities reducing chance of detection
- NSEC3 extension mitigates problem by using (salted) hashes of domain names

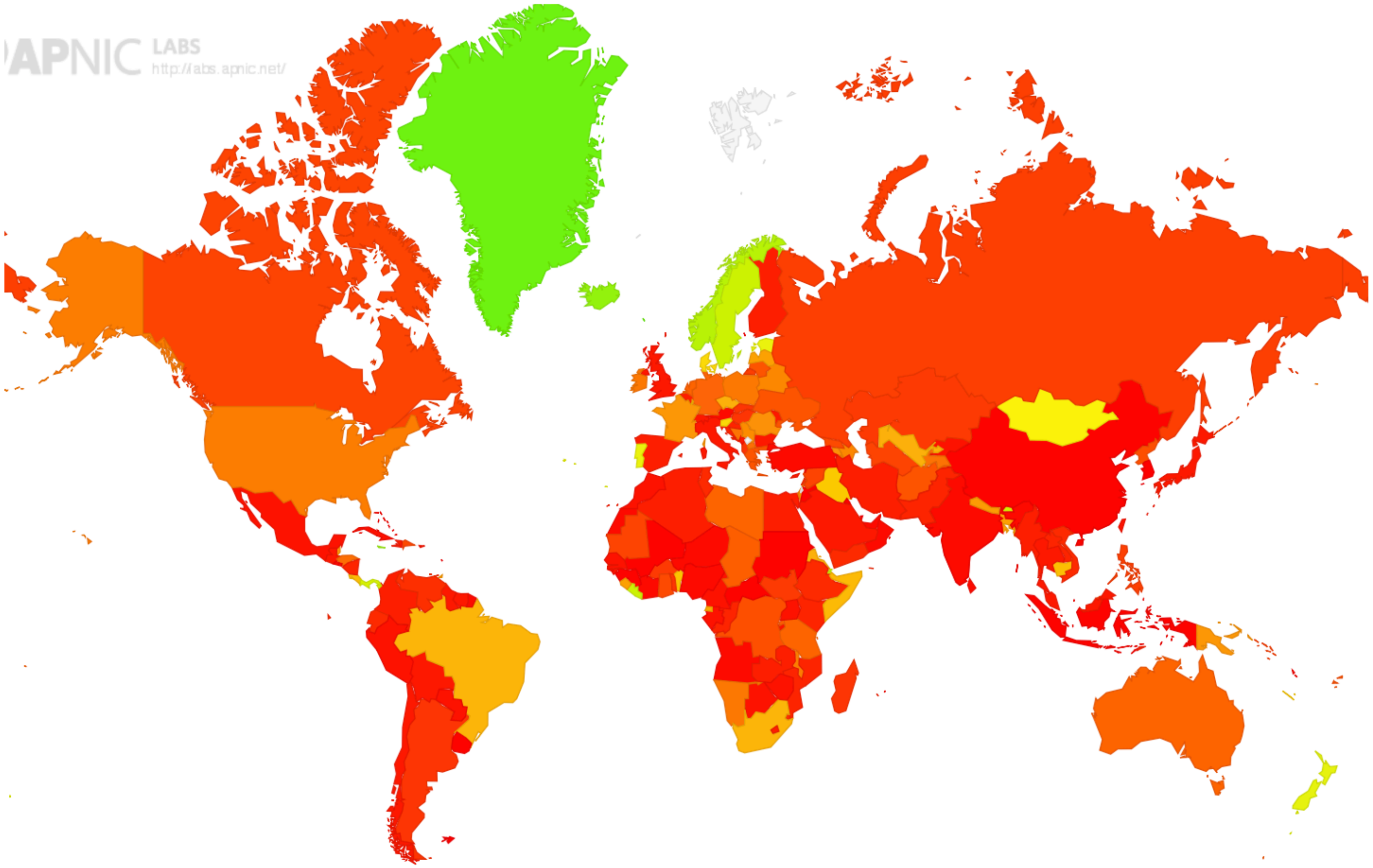
Hash(alice|65BF) = F34DDF56	sort by hash	4EE23198
Hash(bob|65BF) = 7B03235D		7B03235D
Hash(charlie|65BF) = 4EE23198		D14DEA64
Hash(zoey|65BF) = D14DEA64		F34DDF56

  - Failed query: “resolve bob.example.com”
  - Response: “no records exist between 4EE23198.example.com and D14DEA64.example.com, the salt is 65BF”
  - Still useful as a proof of non-existence
    - Given salt, check that 4EE23198 < Hash(bob|65BF) < D14DEA64
  - Salt hinders dictionary attacks: changes over time and across zones

DNSSEC adoption

Imperial College London

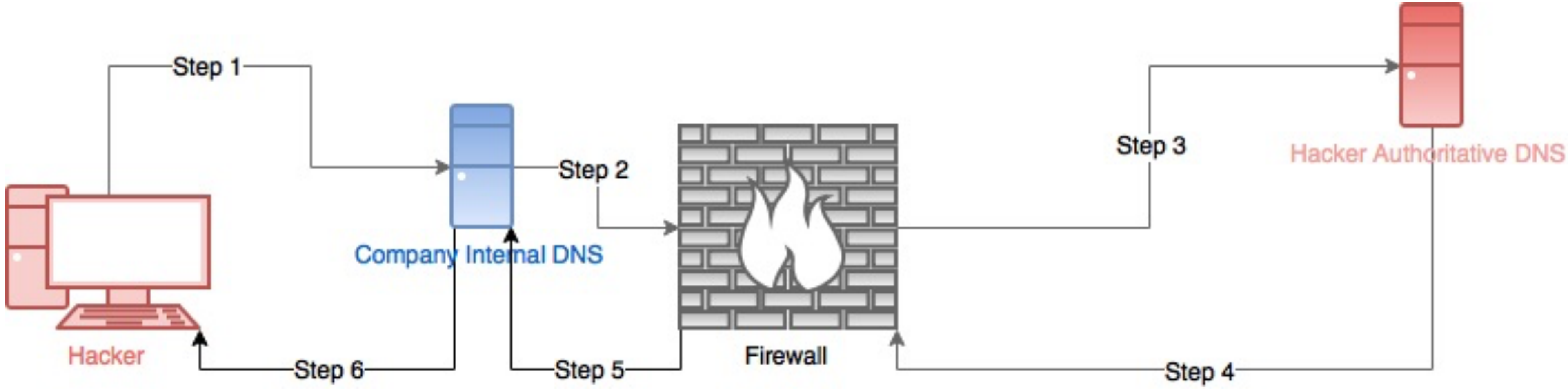
- Not widely adopted yet
  - Validation rate: USA 25%, UK 5%, CN 1%
- As more services support DNSSEC, it may become the standard
- Google's Public DNS uses DNSSEC by default
  - IPv4: 8.8.8.8 and 8.8.8.4
  - IPv6: 2001:4860:4860::8888 and 2001:4860:4860::8844



DNS tunneling

Imperial College London

- Goal: bypass a firewall or proxy that prevents HTTP communication with the target



1. Attacker encodes data to be sent in a DNS query for a domain for which he controls the authoritative DNS
2. Domain is not found locally, eventually authoritative server is contacted
3. DNS queries (and in particular to non-blacklisted domains) are not filtered
4. Server replies encoding data in DNS response
5. Firewall forwards innocent-looking response
6. Attacker receives and decodes the reply

- Vanilla version: exfiltrate data encoded as subdomain-names
- Advanced version: DNS SOCKS proxy to browse arbitrary websites (very slowly)