Computer Networks: Domain Name System
Domain Name System

- The **domain name system** (DNS) is an application-layer protocol.
- **Basic function of DNS**
  - Map domain names to IP addresses
  - The mapping is many to many
- **Examples:**
  - `www.cs.brown.edu` and `cs.brown.edu` map to `128.148.32.12`
  - `google.com` maps to `198.7.237.251`, `198.7.237.249`, and other addresses

- More generally, DNS is a distributed database that stores **resource records**
  - **Address (A) record:** IP address associated with a host name
  - **Mail exchange (MX) record:** mail server of a domain
  - **Name server (NS) record:** authoritative server for a domain

02/22/10 Networks: DNS
Domains

- **Domain name**
  - Two or more labels, separated by dots (e.g., `cs.brown.edu`)

- **Top-level domain (TLD)**
  - Generic (gTLD), e.g., `.com`, `.org`, `.net`
  - Country-code (ccTLD), e.g., `.ca`, `.it`

- **ICANN**
  - Internet Corporation for Assigned Names and Numbers
  - Nonprofit

- **ICANN**
  - Keeps database of registered gTLDs (InterNIC)
  - Accredits registrars for gTLDs

- **gTLDs**
  - Managed by ICANN

- **ccTLDs**
  - Managed by government organizations
DNS Tree

- com
  - google.com
  - microsoft.com
  - stanford.edu

- edu
  - brown.edu
  - cs.brown.edu
  - math.brown.edu

resource records

Networks: DNS
**Name Servers**

- **Name server**
  - Keeps local database of DNS records
  - Answers DNS queries
  - Can ask other name servers if record not in local database

- **Authoritative name server**
  - Stores reference version of DNS records for a zone (partial tree)

- **Examples**
  - `dns.cs.brown.edu` is authoritative for `cs.brown.edu`
  - `bru-ns2.brown.edu` is authoritative for `brown.edu`, except `cs.brown.edu`

- **Root servers**
  - Authoritative for the root zone (TLDs)
  - `[a-m].root-servers.net`
  - Supervised by ICANN
Name Resolution

• Resolver
  • Program that retrieves DNS records
  • E.g., dig in Linux and nslookup in Windows
  • Caches records received
  • Connects to a name server (default, root, or given)

• Iterative resolution
  • Name server refers client to authoritative server (e.g., a TLD server) via an NS record
  • Repeat

• Recursive resolution
  • Name server queries another server and forwards the final answer (e.g., A record) to client
Iterative Name Resolution

Local Machine

Application

Resolver

local name server

Resolver

(root)

Resolver

f.root-servers.net

com

Resolver

d.gtld-servers.net

google.com

Resolver

ns2.google.com

www.google.com A 74.125.226.116

query

www.google.com

(root) NS f.root-servers.net

com NS d.gtld-servers.net

google.com NS ns2.google.com
Recursive Name Resolution

- Application
- Resolver
- google.com server
  - query
  - answer: A 74.125.226.176

- local name server
  - Resolver
  - query
  - answer: A 74.125.226.176

- other name server
  - Resolver
  - ...
Glue Records

• Circular references
  • The authoritative name server for a domain may be within the same domain
  • E.g., dns.cs.brown.edu is authoritative for cs.brown.edu

• Glue record
  • Record of type A (IP address) for a name server referred to NS record
  • Essential to break circular references

• Example
  • brown.edu.  NS bru-ns1.brown.edu.
  • bru-ns1.brown.edu.  A  128.148.248.11  [glue record]
DNS Caching

• There would be too much network traffic if a path in the DNS tree would be traversed for each query
  – Root servers and TLD servers would be rapidly overloaded
• DNS servers **cache** records that are results of queries for a specified amount of time
  – Time-to-live field
• DNS queries with caching
  – First, resolver looks in cache for A record of query domain
  – Next, resolver looks in cache for NS record of longest suffix of query domain
Iterative Name Resolution with Caching

- Query: `www.google.com`
- Local Machine
  - Application
  - Resolver
  - DNS Cache
    - com NS d.gtld-servers.net
    - ... (other records)

- local name server
  - Resolver

- (root)
  - Resolver
    - f.root-servers.net

- Resolver
  - com
    - Resolver
      - d.gtld-servers.net

- Resolver
  - google.com
    - Resolver
      - ns2.google.com

- google.com NS ns2.google.com

- www.google.com A 74.125.226.116

Networks: DNS
Recursive Name Resolution with Caching

- **local machine**
  - Application
  - Resolver

- **local name server**
  - Resolver
  - DNS Cache
    - google.com A 74.125.226.176
    - google.com A 74.125.226.176
    - ...

- **other name server**
  - Resolver

- Application sends a query to the local resolver.
- The resolver looks in the DNS cache for the answer to google.com.
- If found, the answer (A 74.125.226.176) is returned.
- Otherwise, the resolver queries the local name server, and if the answer is cached, it is returned.
- If not cached, it is queried from another name server and cached for future use.
Local DNS Cache

• Operating system maintains DNS cache
  – Shared among all running applications
  – Can be displayed all users
  – View DNS cache in Windows with command `ipconfig /displaydns`
  – Clear DNS cache in Windows with command `ipconfig /flushdns`

• Privacy issues
  • Browsing by other users can be monitored
  • Note that private/incognito browsing does not clear DNS cache
DNS Cache Poisoning

• Basic idea
  • Give a DNS server a false address record and get it cached

• DNS query mechanism
  • Queries issued over UDP on port 53
  • 16-bit request identifier in payload to match answers with queries
  • No authentication

• Cache may be poisoned when a resolver
  • Disregards identifiers
  • Has predictable identifiers and return ports
  • Accepts unsolicited DNS records

• Early versions of BIND (popular DNS software) vulnerable to cache poisoning
DNS Cache Poisoning Defenses

- Query randomization
  - Random request identifier (16 bits)
  - Random return port (16 bits)
- Probability of guessing request ID or return port
  \[ \frac{1}{2^{16}} = 0.0015\% \]
- Probability of guessing request ID and return port is
  \[ \frac{1}{2^{32}} \text{ (less than one in four billion)} \]

- Check request identifier
- Use signed records
  - DNSSEC
Kaminsky’s Attack

- Attacker causes victim to send
  - Many DNS requests for nonexistent subdomains of target domain
- Attacker sends victim
  - Forged NS responses for the requests
- Format of forged response
  - Random ID
  - Correct NS record
  - Spoofed glue record pointing to the attacker’s name server IP

02/22/10

Networks: DNS
DNSSEC

- **Goals**
  - Authenticity of DNS answer origin
  - Integrity of reply
  - Authenticity of denial of existence

- **Implementation**
  - Signed DNS replies at each step
  - Public-key cryptography
  - Certificates in the OS

- **Slow deployment**
  - Root servers support since 2010

02/22/10 Networks: DNS
References

- **RFC 1034** (Domain Names – Concepts and Facilities)
- **CSCI 1680 slides on DNS**
- **Dan Kaminsky's 2008 Black Hat talk**