Today

• Identifiers and addressing

• Domain Name System
  – Telephone directory of the Internet
  – Protocol format
  – Caching: Load balancing
  – Security Challenges
DNS: Application Layer Protocol

• distributed database
  – implemented in hierarchy of many name servers.

• application-layer protocol:
  – hosts communicate to name servers
  – resolve names → addresses

• note: core Internet function, implemented as application-layer protocol
DNS: domain name system

- distributed database implemented in hierarchy of many name servers.
- application-layer protocol: hosts, name servers communicate to resolve names → addresses
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network’s “edge”
DNS: Hostname to IP translation

- Human-readable strings: www.example.com
- (Not much addressing here, ports to ID socket)
- IP addresses (32-bit IPv4, 128-bit IPv6)
- (Network dependent) Ethernet: 48-bit MAC address
DNS Services

- DNS is an application-layer protocol. E2E design!
- It provides:
  - Hostname to IP address translation
  - Host aliasing (canonical and alias names)
  - Mail server aliasing
  - Load distribution (one name may resolve to multiple IP addresses)
  - Lots of other stuff that you might use a directory service to find. (Wikipedia: List of DNS record types)
DNS: a distributed, hierarchical database

- com DNS servers
  - yahoo.com DNS servers
  - amazon.com DNS servers

- org DNS servers
  - pbs.org DNS servers

- edu DNS servers
  - swarthmore.edu DNS servers
  - umass.edu DNS servers
  - cs.swarthmore.edu DNS servers
  - allspice.cs.swarthmore.edu (other cs hosts)
  - Host

• allspice.cs.swarthmore.edu.

Nameless root, Usually implied.
Domain Name System (DNS)

• Distributed administrative control
  – Hierarchical name space divided into zones
  – Distributed over a collection of DNS servers

• Hierarchy of DNS servers
  – Root servers
  – Top-level domain (TLD) servers
  – Authoritative DNS servers

• Performing the translations
  – Local DNS servers
  – Resolver software
Resolution Process: As an end host if you want to look up a hostname (swarthmore.edu) who do you contact?

A. Contact the swarthmore DNS servers
B. Contact edu DNS servers
C. Contact the Root DNS servers
D. Someone else should do this job.
Resolution Process

• End host wants to look up a name, who should it contact?
  – It could traverse the hierarchy, starting at a root
  – More efficient for ISP to provide a local server

• *ISP’s local server for handling queries not necessarily a part of the pictured hierarchy*
Local DNS Name Server

• Each ISP
  – (residential ISP, company, university) ...
  – has (at least) one

• also called “default name server”
DNS query host -> local DNS server

• Local DNS server
  – acts as proxy, forwards query into hierarchy
  – has local cache of recent name-to-address translation pairs (but may be out of date!)
DNS name resolution example #1

- allspice wants IP address for gaia.cs.umass.edu

iterative query:
- contacted server replies with name of server to contact
  - “I don’t know this name, but ask this server”
Inserting (or changing) records

1. Example: new startup “Network Utopia”
2. Register networkuptopia.com at DNS registrar
   a) provide names, IP addresses of authoritative name server (primary and secondary)
   b) registrar inserts two RRs into .com TLD server
      (networkutopia.com, dns1.networkutopia.com, NS)
      (dns1.networkutopia.com, 212.212.212.1, A)
3. Set up authoritative server at that name/address: Create records for the services:
   a) type A record for www.networkuptopia.com
   b) type MX record for @networkutopia.com email
Inserting (or changing) records

1 Example: new startup “Network Utopia”
   • hierarchical picture of where we want our new web and mail servers to be
Any client end host should be able to reach our new server.

- let’s say allspice.cs.swarthmore.edu wants to reach networkutopia.com
- what information does it need?
Inserting (or changing) records

Root DNS Servers

- com DNS servers
  - yahoo.com
    - Auth. DNS Server
  - dns1.networkutopia.com
    - Authoritative DNS server for Network Utopia
    - Website host name: networkutopia.com
    - Email Server: @networkutopia.com
- org DNS servers
  - pbs.org
    - Auth. DNS Server
- edu DNS servers
  - swarthmore.edu
    - Auth. DNS Server
  - umass.edu
    - Auth. DNS Server

- allspice.cs.swarthmore.edu
  - Client Host Name

3. allspice contacts Root servers
   - allspice has a cached copy of the IP addresses of the root servers since they are unchanging.
   - allspice queries the root servers “what’s the IP address of networkutopia.com”
   - root server response: I can get you to the .com servers: here are the DNS server names of the .com servers, and their IP addresses.
allspice contacts .com servers

- allspice uses the root server response to query the .com servers.
- \textit{query: “what’s the IP address of networkutopia.com”}
- \textit{.com server response: I need to get you to the authoritative DNS server for networkutopia.com}
Inserting (or changing) records

Root DNS Servers

.com DNS servers
- yahoo.com
  Auth. DNS Server

.org DNS servers
- dns1.networkutopia.com
  Authoritative DNS server for Network Utopia
- pbs.org
  Auth. DNS Server
- networkutopia.com
  Website host name
- @networkutopia.com
  Email Server

.edu DNS servers
- swarthmore.edu
  Auth. DNS Server
- umass.edu
  Auth. DNS Server
- allspice.cs.swarthmore.edu
  Client Host Name

3a allspice contacts .com servers
- .com server response: I need to get you to the authoritative DNS server for networkutopia.com

As the owners of networkutopia.com we need to let .com know how to reach our DNS server
Inserting (or changing) records

To do so, we add two entries in the .com server:
1. NS record to redirect: networkutopia.com can be reached via our DNS server dns1.networkutopia.com
2. the IP address of our DNS server: dns1.networkutopoa.com is 212.212.212.1 (made up IP)
allspice contacts .com servers

- .com server response (like root response): I can get you to the .com server. Authoritative DNS server for networkutopia.com: here is the DNS server name (dns1.networkutopia.com) and its IP addresses (212.212.212.1).
allspice contacts dns1.networkutopia.com

- allspice uses the .com server response to query the DNS server of network utopia.
  - query: "what’s the IP address of networkutopia.com"
  - dns1.networkutopia.com: I will give you the IP address of networkutopia.com
Inserting (or changing) records

Root DNS Servers

.com DNS servers
  yahoo.com
  com Auth. DNS Server

.org DNS servers
  pbs.org
  org Auth. DNS Server

.edu DNS servers
  swarthmore.edu
  edu Auth. DNS Server

.university.edu

Authoritative DNS server for Network Utopia

dns1.networkutopia.com
  networkutopia.com
  Website host name

@networkutopia.com
  Email Server

umass.edu
  Auth. DNS Server

allspice contacts dns1.networkutopia.com
  • dns1.networkutopia.com: I will give you the IP address of networkutopia.com

To do so, we need to add an entry into our authoritative DNS server for network utopia:
1. the IP address of networkutopia.com: 212.1.2.3 (made up)
2. if we have a mail server, we add the address for this too:
   mail.networkutopia.com has the IP address: 212.1.2.4
Inserting (or changing) records

Root DNS Servers

.com DNS servers
- yahoo.com
  Auth. DNS Server
- .com DNS server
  Authoritative DNS server for Network Utopia

.org DNS servers
- pbs.org
  Auth. DNS Server
- .com DNS server
  Authoritative DNS server for Network Utopia

.edu DNS servers
- swarthmore.edu
  Auth. DNS Server
- umass.edu
  Auth. DNS Server

networkutopia.com
  Website host name

@networkutopia.com
  Email Server

allspice.cs.swarthmore.edu
  Client Host Name

Insert NS, A records to reach DNS Server for network Utopia
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)

Insert A, MX records to reach Web server and mail server for network Utopia
(networkutopia.com, 212.212.212.1, A)
(networkutopia.com, mail.networkutopia.com, MX)
(mail.networkutopia.com, 212.212.212.2, A)
Caching

- Once (any) name server learns a mapping, it caches mapping
  - cache entries timeout (disappear) after some time (TTL: time to live)
  - TLD servers typically cached in local name servers
  - Thus root name servers not often (legitimately) visited
The TTL value should be…

A. Short, to make sure that changes are accurately reflected

B. Long, to avoid re-queries of higher-level DNS servers

C. Something else
Caching

• Once (any) name server learns a mapping, it caches mapping
  – cache entries timeout (disappear) after some time (TTL: time to live)
  – TLD servers typically cached in local name servers.
  – Root name servers not often (legitimately) visited
• (+) Subsequent requests need not burden DNS
• (-) Cached entries may be out-of-date (best effort!)
  – If host’s name or IP address changes, it may not be known Internet-wide until all TTLs expire
DNS as Indirection Service

• DNS gives us very powerful capabilities
  – Not only easier for humans to reference machines!

• Changing the IPs of machines becomes trivial
  – e.g. you want to move your web server to a new host
  – Just change the DNS record!
Load Balancing

One domain can map to multiple machines

www.google.com
Aliasing

One machine can have many aliases

www.reddit.com

www.foursquare.com

www.huffingtonpost.com

eastcoast.akamai.net
Content Delivery Networks

DNS responses may vary based on geography, ISP, etc.
DNS: distributed DB storing resource records (RR)

RR format: (name, value, type, ttl)

type=A
- name is hostname
- value is IP address

type=NS
- name is domain (e.g., foo.com)
- value is hostname of authoritative name server for this domain

type=CNAME
- name is alias name for some “canonical” (the real) name
- www.ibm.com is really servereast.backup2.ibm.com
- value is canonical name

type=MX
- value is name of mailserver associated with name
DNS Types

RR format: (name, value, type, ttl)

• Type = A / AAAA
  – Name = domain name
  – Value = IP address
  – A is IPv4, AAAA is IPv6

• Type = NS
  – Name = partial domain
  – Value = name of DNS server for this domain
  – “Go send your query to this other server”

Query
Name: cs.swarthmore.edu
Type: A

Resp.
Name: cs.swarthmore.edu
Value: 130.58.68.9

Query
Name: cs.swarthmore.edu
Type: NS

Resp.
Name: cs.swarthmore.edu
Value: 130.58.68.9
**DNS Types, Continued**

**RR format:** `(name, value, type, ttl)`

- **Type = CNAME**
  - Name = hostname
  - Value = canonical hostname
  - Useful for aliasing
  - CDNs use this

- **Type = MX**
  - Name = domain in email address
  - Value = canonical name of mail server

---

**Query**

- Name: `foo.mysite.com`
- Type: CNAME

**Resp.**

- Name: `foo.mysite.com`
- Value: `bar.mysite.com`

**Query**

- Name: `cs.umass.edu`
- Type: MX

**Resp.**

- Name: `cs.umass.edu`
- Value: `barramail.cs.umass.edu.`
DNS protocol, messages

- query and reply messages, both with same message format

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td># questions</td>
<td># answer RRs</td>
</tr>
<tr>
<td># authority RRs</td>
<td># additional RRs</td>
</tr>
<tr>
<td>Questions</td>
<td></td>
</tr>
<tr>
<td>Answers</td>
<td></td>
</tr>
<tr>
<td>Authority</td>
<td></td>
</tr>
<tr>
<td>Additional info</td>
<td></td>
</tr>
</tbody>
</table>
DNS protocol, messages

- query and reply messages, both with same message format

Message header
- **identification**: 16 bit # for query, reply to query uses same #
- **flags**:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative

Sent via UDP!
- No connection established
- Not reliable
DNS name resolution example

- **Requesting host**: allspice.cs.swarthmore.edu
- **Local DNS server**: dns.cs.swarthmore.edu
- **Authoritative DNS server**: dns.cs.umass.edu
- **TLD DNS server**:
- **Root DNS server**:

**Steps of the resolution process**:
1. Requesting host queries local DNS server.
2. Local DNS server queries root DNS server.
3. Root DNS server returns TLD DNS server address.
4. TLD DNS server returns authoritative DNS server address.
5. Authoritative DNS server returns IP address.
6. IP address is returned to local DNS server.
7. Local DNS server returns IP address to requesting host.
8. Requesting host receives IP address.
Example: iterative query using dig()
dig demo.cs.swarthmore.edu

;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39007
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp 4096

;; QUESTION SECTION:
demo.cs.swarthmore.edu. IN A

;; ANSWER SECTION:
demo.cs.swarthmore.edu. 86400 IN A 130.58.68.26

;; Query time: 41 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Sep 28 09:38:19 EDT 2020
;; MSG SIZE  rcvd: 67
```
;; DiG 9.10.6 <<>> facebook.com
;; global options: +cmd
;; Got answer:
;; Header: QUERY, status: NOERROR, id: 13938
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
facebook.com. IN A

;; ANSWER SECTION:
facebook.com. 300 IN A 157.240.220.35

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:
a.ns.facebook.com. 160673 IN A 129.134.30.12
b.ns.facebook.com. 160673 IN A 129.134.31.12
c.ns.facebook.com. 160673 IN A 185.89.218.12
d.ns.facebook.com. 160673 IN A 185.89.219.12
a.ns.facebook.com. 160673 IN AAAA 2a03:2880:f0fc:c:face:b00c:0:35
b.ns.facebook.com. 160673 IN AAAA 2a03:2880:f0fd:c:face:b00c:0:35
c.ns.facebook.com. 160673 IN AAAA 2a03:2880:f1fc:c:face:b00c:0:35
d.ns.facebook.com. 160673 IN AAAA 2a03:2880:f1fd:c:face:b00c:0:35

;; Query time: 40 msec
;; SERVER: 128.119.240.1#53(128.119.240.1)
;; WHEN: Mon Sep 28 10:09:13 EDT 2020
;; MSG SIZE rcvd: 300
```
DiG 9.16.1–Ubuntu 

global options: +cmd 

; Got answer:
; -->>HEADER<<-- opcode: QUERY, status: NOERROR, id: 12267
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: fcfd45ce63936062afea660657f71f2e022337d09cde845d2 (good)
; QUESTION SECTION:
facebook.com. IN A

; ANSWER SECTION:
facebook.com. 300 IN A 31.13.71.36

; AUTHORITY SECTION:
facebook.com. 125611 IN NS d.ns.facebook.com.

; ADDITIONAL SECTION:
a.ns.facebook.com. 125611 IN A 129.134.30.12
b.ns.facebook.com. 125611 IN A 129.134.31.12
c.ns.facebook.com. 125611 IN A 185.89.218.12
d.ns.facebook.com. 125611 IN A 185.89.219.12
a.ns.facebook.com. 125611 IN AAAA 2a03:2880:f0fc:c:face:b000:c0:35
b.ns.facebook.com. 125611 IN AAAA 2a03:2880:f0fd:c:face:b000:c0:35
c.ns.facebook.com. 125611 IN AAAA 2a03:2880:f1fc:c:face:b000:c0:35
d.ns.facebook.com. 125611 IN AAAA 2a03:2880:f1fd:c:face:b000:c0:35

; Query time: 7 msec
; SERVER: 130.58.68.10#53(130.58.68.10)
; WHEN: Mon Sep 28 10:27:44 EDT 2020
; MSG SIZE rcvd: 328
DiG 9.10.6

GLOBAL OPTIONS: +cmd

Got answer:

HEADER<-- opcode: QUERY, status: NOERROR, id: 39016
flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

OPT PSEUDOSECTION:
EDNS: version: 0, flags; udp: 4096
QUESTION SECTION:
35.220.240.157.in-addr.arpa. IN PTR

ANSWER SECTION:

Query time: 41 msec
SERVER: 192.168.1.1#53(192.168.1.1)
WHEN: Mon Sep 28 10:22:01 EDT 2020
MSG SIZE rcvd: 109

DiG 9.10.6

GLOBAL OPTIONS: +cmd

Got answer:

HEADER<-- opcode: QUERY, status: NOERROR, id: 32758
flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

OPT PSEUDOSECTION:
EDNS: version: 0, flags; udp: 4096
QUESTION SECTION:
35.66.13.31.in-addr.arpa. IN PTR

ANSWER SECTION:

Query time: 42 msec
SERVER: 192.168.1.1#53(192.168.1.1)
WHEN: Mon Sep 28 10:22:18 EDT 2020
MSG SIZE rcvd: 106
DiG 9.16.1-Ubuntu <><> nytimes.com

; global options: +cmd
; Got answer:
; HEADER<<- opcode: QUERY, status: NOERROR, id: 5595
; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 7, ADDITIONAL: 11

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 828eae4b2592b9d098ca488e5f71f46c360bb78b40b0038a (good)
; QUESTION SECTION:
;    nytimes.com.

; ANSWER SECTION:
    nytimes.com. 120 IN A 151.101.193.164
    nytimes.com. 120 IN A 151.101.129.164
    nytimes.com. 120 IN A 151.101.1.164
    nytimes.com. 120 IN A 151.101.65.164

; AUTHORITY SECTION:
    nytimes.com. 172800 IN NS dns4.p06.nsone.net.
    nytimes.com. 172800 IN NS dns3.p06.nsone.net.
    nytimes.com. 172800 IN NS dns1.p06.nsone.net.
    nytimes.com. 172800 IN NS dns2.p06.nsone.net.

; ADDITIONAL SECTION:
    ns5.dnsmadeeasy.com. 38834 IN A 208.94.148.13
    ns6.dnsmadeeasy.com. 38834 IN A 208.80.124.13
    ns7.dnsmadeeasy.com. 38834 IN A 208.80.126.13
    dns1.p06.nsone.net. 18528 IN A 198.51.44.6
    dns2.p06.nsone.net. 18528 IN A 198.51.45.6
    dns3.p06.nsone.net. 18528 IN A 198.51.44.70
    dns4.p06.nsone.net. 18528 IN A 198.51.45.70
    ns5.dnsmadeeasy.com. 38834 IN AAAA 2600:1800:5::1
    ns6.dnsmadeeasy.com. 38834 IN AAAA 2600:1801:6::1
    ns7.dnsmadeeasy.com. 38834 IN AAAA 2600:1802:7::1

; Query time: 19 msec
; SERVER: 130.58.68.10#53(130.58.68.10)
; WHEN: Mon Sep 28 10:34:20 EDT 2020
; MSG SIZE  rcvd: 483
DNS security

DNS Vulnerabilities:
- No authentication
- Connectionless transport layer protocol (UDP)

DNS Attacks:
- Amplification Attack
- Cache Poisoning
- Man-in-the-middle
- DNS Redirection
- DDoS
- DNS Injection

Requesting host: allspice.cs.swarthmore.edu

Authoritative DNS server: dns.cs.umass.edu

TLD DNS server: dns.cs.swarthmore.edu

Local DNS server: dns.cs.swarthmore.edu

Root DNS server: gaia.cs.umass.edu
Attacking DNS

DDoS attacks
- Bombard root servers with traffic
  - Not successful to date
  - Traffic Filtering
  - Local DNS servers cache IPs of TLD servers, bypassing root
- Bombard TLD servers
  - Potentially more dangerous

Redirect attacks
- Man-in-middle
  - Intercept queries
- DNS poisoning
  - Send bogus replies to DNS server that caches

Exploit DNS for DDoS
- Send queries with spoofed source address: target IP
- Requires amplification
DNSSEC Hierarchy of Trust

Where is bankofamerica.com?

Root Zone (ICANN)

.com (Verisign)

dns.evil.com
dns.bofa.com

IP: 66.66.66.93
Key: <     >
SIG: 9na8x7040a3

Slide 48
Solution: DNSSEC

• Cryptographically sign critical resource records
  – Resolver can verify the cryptographic signature
• Two new resource *types*
  – Type = DNSKEY
    • Name = Zone domain name
    • Value = Public key for the zone
  – Type = RRSIG
    • Name = (type, name) tuple, i.e. the query itself
    • Value = Cryptographic signature of the query results

---

**Prevents hijacking and spoofing**

**Creates a hierarchy of trust within each zone**
DNSSEC Deployment

- On the roots since July 2010
- Verisign enabled it on .com and .net in January 2011
- Comcast is the first major ISP to support it (January 2012)
- Continues to be weak

DNS Cache Poisioning/Spoofing

- Respond with bogus information for a DNS query
- UDP: forge header data to masquerade as legitimate response
  - No TCP connection to “handshake” and verify identity.
- Attacker can:
  - point to malicious website
  - phishing attacks: fake version of genuine website
  - poison victim with computer worms/viruses
DNS Cache Poisoning/Spoofing

Poisoned DNS Cache:

Attacker

Courtesy: Cloudflare, DNS Cache Poisoning
DNS Cache Poisoning: Prevention

• Attacker needs to figure out:
  – no cached entry
  – use the same request ID
  – use the same UDP port #
  – figure out the authoritative name server

• DNSSEC
  – source port randomization (not always successful)
  – request ID randomization
  – 0x20 encoding for queries: DNS is case insensitive
    • resolver/server agree on a shared key
    • www.gOoGIE.com
DNSSEC offers authentication of known DNS servers using a chain-of-trust starting from the root server to an authoritative name server. How do you think the root server establishes its authenticity?

A. That’s a single point of failure for DNSSEC
B. Another service establishes root server authenticity
C. A group of people ratify the root server authenticity
D. Some other way
E. Some combination of the above
DNS: Root Signing Ceremony

• Trusting the integrity, authenticity of the root zone

• Root-signing key:
  – El Segundo, CA,
  – Culpeper, VA

• Ceremony participants: Distributed control
  – no one person can modify the private key
What kinds of attacks do you think are mitigated by using DNSSEC?

A. DNS Redirection
   • Cache Poisoning
   • Man-in-the-middle
   • DNS Injection

B. DDoS
   • Amplification/Reflection Attack
Summary

• DNS maps human readable names to IP addresses

• DNS arranged into a hierarchy
  – Scalability / distributed responsibility
  – Autonomous control of local name servers

• Caching is crucial for performance

• DNSSEC provides security improvements to DNS.