Network Applications: DNS

Y. Richard Yang

http://zoo.cs.yale.edu/classes/cs433/

2/1/2016
Outline

- Admin and recap
- DNS
Admin

- 72 discretionary late hours for assignments across the semester
Recap: The Big Picture of the Internet

- **Hosts and routers:**
  - ~ 1 billion hosts (2015)
  - Organized into ~50K networks
  - Backbone links 100 Gbps

- **Software:**
  - Datagram switching with virtual circuit support
  - Layered network architecture
    - Use end-to-end arguments to determine the services provided by each layer
  - The hourglass architecture of the Internet
Protocol Formats

- 16-bit source port number
- 16-bit destination port number
- 16-bit UDP length
- 16-bit UDP checksum
- Data (if any)

TCP header:
- 4-bit version
- 4-bit header length
- 8-bit type of service (TOS)
- 16-bit total length (in bytes)
- 16-bit identification
- 13-bit fragment offset
- 8-bit time to live (TTL)
- 8-bit protocol
- 16-bit header checksum
- 32-bit source IP address
- 32-bit destination IP address
- Options (if any)
- Data

Ethernet frame:
- DA: 6
- SA: 6
- Type: 2
- Data: 46-1500
- CRC: 4

Minimum size = 64 bytes
Multiplexing/Demultiplexing

TCP sockets

UDP sockets

Applications

Socket references

Sockets bound to ports

TCP ports 1 2 ... 65535

TCP

UDP

UDP ports

IP
Recap: Client-Server Paradigm

- The basic paradigm of network applications is the client-server (C-S) paradigm

- Some key design questions to ask about a C-S application:
  - extensibility
  - scalability
  - robustness
  - security
Recap: Email App

Some nice protocol extensibility design features

- separate protocols for different functions
- simple/basic (smtp) requests to implement basic control; fine-grain control through ASCII header and message body
- status code in response makes message easy to parse
Email: Challenge

- A large percentage of spam/phish

Global spam volume as percentage of total e-mail traffic from 2007 to 2014

Recap: Spam Detection Methods by GMail

- Known phishing scams
- Message from unconfirmed sender identity
- Message you sent to Spam/similarity to suspicious messages
- Administrator-set policies
- Empty message content

https://support.google.com/mail/answer/1366858?hl=en
Current Email Authentication Approaches

Sender Policy Frame (SPF)  DomainKeys Identified Mail (DKIM)
Sender Policy Framework (SPF RFC7208)

SPF Exercise

- Test 1
  - Send real email by gmail
  - POP retr

- Test 2
  - Send using telnet
  - POP retr
Key Remaining Question for SPF?

- How does SPF know if its neighbor MTA is a permitted sender of the domain?
DomainKeys Identified Mail (DKIM; RFC 5585)

- A domain-level digital signature authentication framework for email, using public key crypto
  - E.g., gmail.com signs that the message is sent by gmail server

- Basic idea of public key signature
  - Owner has both public and private keys
  - Owner uses private key to sign a message to generate a signature
  - Others with public key can verify signature
Example: RSA

1. Choose two large prime numbers \( p, q \).
   (e.g., 1024 bits each)

2. Compute \( n = pq, \ z = (p-1)(q-1) \)

3. Choose \( e \) (with \( e < n \)) that has no common factors
   with \( z \). (\( e, z \) are “relatively prime”).

4. Choose \( d \) such that \( ed-1 \) is exactly divisible by \( z \).
   (in other words: \( ed \mod z = 1 \)).

5. Public key is \((n,e)\). Private key is \((n,d)\).
RSA: Signing/Verification

0. Given \((n,e)\) and \((n,d)\) as computed above

1. To sign message, \(m\), compute \(h = \text{hash}(m)\), then sign with private key
   \[ s = h^d \mod n \] (i.e., remainder when \(h^d\) is divided by \(n\))

2. To verify signature \(s\), compute
   \[ h' = s^e \mod n \] (i.e., remainder when \(s^e\) is divided by \(n\))

   \[ h = (h^d \mod n)^e \mod n \]

   Magic happens!

The magic is a simple application of Euler's generalization of Fermat's little theorem
DomainKeys Identified Mail (DKIM)

Is the message signed by the private key of the signing domain?

Diagram:
- MUA
- Signing MTA
- MTA
- Verifying MTA
- MUA

Connections:
- MUA to Signing MTA: smtp/submission
- Signing MTA to MTA: smtp
- MTA to Verifying MTA: smtp
- Verifying MTA to MUA: pop/imap
Key Remaining Question about DKIM?

- How does DKIM retrieve the public key of the author domain?
Summary: Client-Server Paradigm

- The basic paradigm of network applications is the client-server (C-S) paradigm

- Some key design questions to ask about a C-S application:
  - extensibility
  - scalability
  - robustness
  - security
High scalability and robustness fundamentally require that multiple email servers serve the same email address.
Mapping Functions Design Alternatives

- Map from an email address server name to IP address of email server

name (e.g., yale.edu)

mapping

1 IP

mapping

multiple IPs

name (e.g., yale.edu)

mapping

multiple IPs
Mapping Functions Design Alternatives

name (e.g., yale.edu)

mapping

1 IP

load balancer (routing)

switch

mapping

name (e.g., yale.edu)

mapping

1 IP

1 IP
Summary: Some Key Remaining Issues about Email

- Basic: How to find the email server of a domain?

- Scalability/robustness: how to find multiple servers for the email domain?

- Security
  - SPF: How does SPF know if its neighbor MTA is a permitted sender of the domain?
  - DKIM: How does DKIM retrieve the public key of the author domain?
Outline

- Recap
- Email security (authentication)
  - DNS
Function

- map between (domain name, service) to value, e.g.,
  - (www.cs.yale.edu, Addr) -> 128.36.229.30
  - (cs.yale.edu, Email) -> netra.cs.yale.edu
DNS Records

DNS: stores resource records (RR)

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

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

- Type=NS
  - name is domain (e.g. yale.edu)
  - value is the name of the authoritative name server for this domain

- Type=CNAME
  - name is an alias name for some “canonical” (the real) name
  - value is canonical name

- Type=MX
  - value is hostname of mail server associated with name

- Type=SRV
  - general extension for services

- Type=TXT
  - general txt

http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml
Try DNS: Examples

- `dig <type> <domain>`
  - `type=MX`
    - `gmail.com`
  - `type=A`
  - `type=TXT`
    - `gmail.com`
    - `20120113._domainkey.gmail.com`
DNS Design: Dummy Design

- DNS itself can be considered as a client-server system as well
- How about a dummy design: introducing one super Internet DNS server?

**THE DNS server of the Internet**
Problems of a Single DNS Server

- Scalability and robustness bottleneck
- Administrative bottleneck
DNS: Distributed Management of the Domain Name Space

- A distributed database managed by authoritative name servers
  - divided into zones, where each zone is a sub-tree of the global tree
  - each zone has its own authoritative name servers
  - an authoritative name server of a zone may delegate a subset (i.e. a sub-tree) of its zone to another name server

```
int  com  edu  gov  mil  org  net  jp  us  nl ...

sun  eng  cs   yale  eng  acm  ieee  jack  jill

ai   linda

robot
```
called a zone
Email Architecture + DNS

Mail server

SMTP

User agent

User agent

User agent

SMTP

SMTP

User agent

User agent

SMTP

POP3 or IMAP

SMTP

DNS
Root Zone and Root Servers

- The root zone is managed by the root name servers
  - 13 root name servers worldwide
  - a. Verisign, Dulles, VA
  - c. Cogent, Herndon, VA (also Los Angeles)
  - d. U Maryland College Park, MD
  - g. US DoD Vienna, VA
  - h. ARL Aberdeen, MD
  - j. Verisign, (11 locations)
  - e. NASA Mt View, CA
  - f. Internet Software C.
    Palo Alto, CA
    (and 17 other locations)
  - b. USC-ISI Marina del Rey, CA
  - i. Autonomica, Stockholm
    (plus 3 other locations)
  - k. RIPE London
    (also Amsterdam, Frankfurt)
  - m. WIDE Tokyo

Linking the Name Servers

- Each name server knows the addresses of the root servers
- Each name server knows the addresses of its immediate children (i.e., those it delegates)

Q: how to query a hierarchy?
DNS Message Flow: Two Types of Queries

**Recursive query:**
- The contacted name server resolves the name completely

**Iterated query:**
- Contacted server replies with name of server to contact
  - “I don’t know this name, but ask this server”
Two Extreme DNS Message Flows

Issues of the two approaches?
Typical DNS Message Flow: The Hybrid Case

- Host knows only local name server
- Local name server is learned from DHCP, or configured, e.g. `/etc/resolv.conf`
- Local DNS server helps clients resolve DNS names
Typical DNS Message Flow: The Hybrid Case

- Host knows only local name server
- Local name server is learned from DHCP, or configured, e.g. /etc/resolv.conf
- Local DNS server helps clients resolve DNS names
- Benefits of local name servers
  - simplifies client
  - Caches/reuses results
Outline

- Recap
- Email security (authentication)
  - DNS
    - High-level design
    - Details
DNS Message Format?