Network Applications: Email Security, DNS

Y. Richard Yang

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

9/14/2017
Outline

- Admin and recap
- Email
  - Basic email systems design
  - Email security
- DNS
Admin

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

Hosts and routers:
- ~1 billion hosts
- 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 5-layer hourglass architecture of the Internet
Multiplexing/Demultiplexing

Diagram:
- Applications
- TCP sockets
- UDP sockets
- TCP ports: 1 2 ... 65535
- UDP ports: 1 2 ... 65535
- TCP
- UDP
- IP

Socket references
Sockets bound to ports
Formats of main protocols

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit source port number</td>
<td></td>
</tr>
<tr>
<td>16-bit destination port number</td>
<td></td>
</tr>
<tr>
<td>16-bit UDP length</td>
<td></td>
</tr>
<tr>
<td>16-bit UDP checksum</td>
<td></td>
</tr>
<tr>
<td>data (if any)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>4-bit</td>
</tr>
<tr>
<td>header length</td>
<td>4-bit</td>
</tr>
<tr>
<td>type of service (TOS)</td>
<td>8-bit</td>
</tr>
<tr>
<td>total length (in bytes)</td>
<td>16-bit</td>
</tr>
<tr>
<td>identification</td>
<td>16-bit</td>
</tr>
<tr>
<td>time to live (TTL)</td>
<td>8-bit</td>
</tr>
<tr>
<td>protocol</td>
<td>8-bit</td>
</tr>
<tr>
<td>header checksum</td>
<td>16-bit</td>
</tr>
<tr>
<td>source IP address</td>
<td>32-bit</td>
</tr>
<tr>
<td>destination IP address</td>
<td>32-bit</td>
</tr>
<tr>
<td>options (if any)</td>
<td></td>
</tr>
<tr>
<td>data</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>6</td>
</tr>
<tr>
<td>SA</td>
<td>6</td>
</tr>
<tr>
<td>Type</td>
<td>2</td>
</tr>
<tr>
<td>Data</td>
<td>46-1500</td>
</tr>
<tr>
<td>CRC</td>
<td>4</td>
</tr>
</tbody>
</table>

Ethernet frame
Minimum size = 64 bytes
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 key design features of Email

• Separate protocols for different functions
  • email access (e.g., POP3, IMAP)
  • email transport (SMTP)

• Separation of envelop and message body (end-to-end arguments)
  • envelop: simple/basic requests to implement transport control;
  • message body: fine-grain control through ASCII header and message body
    • MIME type as self-describing data type
  • Status code in response makes message easy to parse
Recap: Email Authentication Approaches

Sender Policy Frame (SPF)  DomainKeys Identified Mail (DKIM)  Authenticated Results Chain (ARC)
Sender Policy Framework (SPF RFC7208)

Key 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
  - Assumption: difficult to get private key even w/ public key distributed
DomainKeys Identified Mail (DKIM)

Is the message signed by the private key of the signing domain?
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
\[
s = h^d \mod n \quad \text{(i.e., remainder when } h^d \text{ is divided by } n)\]

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

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

The magic is a simple application of Euler's generalization of Fermat's little theorem
Key Question about DKIM?

- How does DKIM retrieve the public key of the author domain?
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?
Scalability/Robustness

- Both scalability and robustness require that multiple email servers serve the same email address.

Client needs an email server's IP address.

Mapping:

- Yale.edu
  - Mail server 130.132.50.7
  - Mail server 130.132.50.8
  - Mail server 130.132.50.9
Mapping Functions Design Alternatives

- Name (e.g., yale.edu)
  - 1 IP
    - Multiple IPs
  - Multiple IPs
    - Multiple IPs
Mapping Functions Design Alternatives

1. Mapping
   - name (e.g., yale.edu)
   - load balancer (routing)
   - switch
   - 1 IP

2. Mapping
   - name (e.g., yale.edu)
   - 1 IP
Outline

- Recap
- Email security (authentication)
  - DNS
DNS: Domain Name System

- **Function**
  - map between (domain name, service) to value, e.g.,
    - (www.cs.yale.edu, addr) -> 128.36.229.30
    - (yale.edu, email) -> chai.mail.yale.edu
    - rosehip.mail.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 of a “canonical” (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

- **Type=PTR**
  - a pointer to another name

Discussion

- Can DNS handle multiple values for the same (name, service)?
Try DNS: Examples

- `dig <name> <type>`
  - Try `yale.edu` and various types
Observations

- MX can return multiple servers
- DNS may rotate the servers in answer
- Address can also return multiple addresses
- SPY is encoded as the txt type
SPF Exercise

- telnet to netra.cs.yale.edu smtp
- cases
  - From: yry@cs.yale.edu
  - From: yry@harvard.edu
  - From: yry@utexas.edu
  - From: yry@facebook.com
- dig <domain> txt to retrieve spf

- POP retr

  http://www.zytrax.com/books/dns/ch9/spf.html
DKIM Exercise

- Send email from gmail and check message
DKIM Example

- DKIM:
  
  ```
  Msg: DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
  d=accounts.google.com; s=20161025;
  h=mime-version:date:feedback-id:message-id:subject:from:to;
  ... 
  ```

  Query:
  ```
  20161025._domainkey.accounts.google.com
  ```

  ```
  txt
  ```

- DKIM introduces a session key to allow multiple public keys
  ```
  <session>._domainkey.<domain>
  ```
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
Email Architecture + 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. ICANN Los Angeles, CA
    - i. Autonomica, Stockholm (plus 3 other locations)
    - k. RIPE London (also Amsterdam, Frankfurt)
    - m. WIDE Tokyo

See http://root-servers.org/ for more details
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 (often called resolvers)
  - simplifies client
  - caches/reuses results
Outline

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

Basic encoding decisions: UDP/TCP, how to encode domain name, how to encode answers...