Network Applications: Email Security, DNS

Y. Richard Yang

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

9/18/2018
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
Formats of main protocols

<table>
<thead>
<tr>
<th>16-bit source port number</th>
<th>16-bit destination port number</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-bit UDP length</td>
<td>16-bit UDP checksum</td>
</tr>
</tbody>
</table>

Data (if any)

<table>
<thead>
<tr>
<th>4-bit version</th>
<th>4-bit header length</th>
<th>8-bit type of service (TOS)</th>
<th>16-bit total length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16-bit identification</td>
<td>0 D F M</td>
<td>13-bit fragment offset</td>
</tr>
<tr>
<td>8-bit time to live (TTL)</td>
<td>8-bit protocol</td>
<td>16-bit header checksum</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>32-bit source IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>32-bit destination IP address</td>
</tr>
</tbody>
</table>

Options (if any)

Data

<table>
<thead>
<tr>
<th>DA</th>
<th>SA</th>
<th>Type</th>
<th>Data</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>46-1500</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 Design Features

Some key design features of Email:
- Separate protocols for different functions
  - email access (e.g., POP3, IMAP)
  - email transport (SMTP)
- A SMTP transaction consists of an envelope and a message body
  - separation of envelope and message body (end-to-end arguments)
    - envelope: 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: Evaluation of SMTP

Key questions to ask about a C-S application

- **extensible?**
  separate envelope and msg; self-describing message; ehlo negotiation

- **scalable?**
  have not seen mechanism yet

- **robust?**
  have not seen mechanism yet

- **security?**
  authentication/authorization (spoof, spam) are major issues of mail transport
Spam Trend

Source: https://www.statista.com/statistics/420400/spam-email-traffic-share-annual/
Current SMTP Authentication Approaches

Sender Policy Frame (SPF)  DomainKeys Identified Mail (DKIM)
Sender Policy Framework (SPF RFC 7208)

DomainKeys Identified Mail (DKIM)

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

- Capture and look at SFP and DKIM in email messages

See pop3-trace.txt
DomainKeys Identified Mail (DKIM; RFC 5585; RFC6376)

- A domain-level digital signature authentication framework for email, using public key crypto, typically RSA

- Basic idea of RSA type 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
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 \(n\))

   **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
Domain-based Message Authentication, Reporting, and Conformance (DMARC) [RFC7489]

Remaining issue: How to handle unauthenticated messages?

See pop3-trace.txt
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?
  - DMARC: How to find the security policy?
Scalability/Robustness

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

Client need 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

yale.edu

yale.edu

yale.edu
Mapping Functions Design Alternatives

- Single IP mapping
  - Name (e.g., yale.edu)
  - 1 IP
  - Multiple IP mapping

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

- **Name**: (e.g., yale.edu)
  - **Mapping**: 1 IP
  - **Load Balancer (Routing)**: switch
  - **Switch**: 1 IP
  - **Name**: (e.g., yale.edu)
  - **Mapping**: 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)
      - \( \rightarrow 128.36.229.30 \)
    - (yale.edu, email)
      - \( \rightarrow \text{chai.mail.yale.edu} \)
      - \( \rightarrow \text{rosehip.mail.yale.edu} \)
DNS Records

**DNS:** stores resource records (RR)

RR format: \((\text{name}, \text{type}, \text{value}, \text{ttl})\)

- **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

Try DNS: Examples

- `dig [@dnsserver] <name> <type>`
  - try yale.edu and various types
  - `dig www.yale.edu ANY`
  - `dig -x IP`  
  - try `www.yale.edu`
Observations

- A name/type can return multiple answers
- DNS may rotate the answered servers
- ...
SPF Exercise

- telnet to netra.cs.yale.edu smtp
- Some test cases
  - From: yry@yale.edu
  - From: yry@harvard.edu
- dig <domain> txt to retrieve spf
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>
  ```
Outline

- Recap
- Email security (authentication)
  - DNS
    - Interface
    - Architecture design
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

![Diagram of DNS zones and sub-domains]

called a zone
Email Architecture + DNS
The root zone is managed by the root name servers

- 13 root name servers worldwide

  a. Verisign, Dulles, VA
  b. USC-ISI Marina del Rey, CA
  c. Cogent, Herndon, VA (also Los Angeles)
  d. U Maryland College Park, MD
  e. NASA Mt View, CA
  f. Internet Software C. Palo Alto, CA (and 17 other locations)
  g. US DoD Vienna, VA
  h. ARL Aberdeen, MD
  i. Autonomica, Stockholm (plus 3 other locations)
  j. Verisign, (11 locations)
  k. RIPE London (also Amsterdam, Frankfurt)
  l. ICANN Los Angeles, CA
  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
    - Interface
    - Architecture design
    - Message design
DNS Message Format?

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

- **Capture the messages**
  - DNS server is at port 53
    - Display and clear DNS cache
      - [https://support.apple.com/en-us/HT202516](https://support.apple.com/en-us/HT202516) (e.g., MAC
        sudo killall -HUP mDNSResponder)
  - visit gmail.com
  - dig +tcp to see TCP mode

- Try to load the dns-capture file from class Schedule page, if you do not want live capture
**DNS Protocol, Messages**

**DNS protocol**: typically over UDP (can use TCP); *query* and *reply* messages, both with the same message format

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions</td>
<td>Number of answer RRs</td>
</tr>
<tr>
<td>Number of authority RRs</td>
<td>Number of additional RRs</td>
</tr>
<tr>
<td>Questions (variable number of questions)</td>
<td></td>
</tr>
<tr>
<td>Answers (variable number of resource records)</td>
<td></td>
</tr>
<tr>
<td>Authority (variable number of resource records)</td>
<td></td>
</tr>
<tr>
<td>Additional information (variable number of resource records)</td>
<td></td>
</tr>
</tbody>
</table>

URL: https://www.ietf.org/rfc/rfc1035.txt
DNS Details

- Header (Sec. 4.1.1 of https://www.ietf.org/rfc/rfc1035.txt)
- Encoding of questions (Sec. 4.1.2):
  - [Label-length label-chars]
- Encoding of answers (Sec. 4.1.3)
  - Pointer format
    (http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml)

- See example DNS packets
Name Encoding

**Queries**

```
gmail.com: type A, class IN
```

<table>
<thead>
<tr>
<th>Name: gmail.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Name Length: 9]</td>
</tr>
<tr>
<td>[Label Count: 2]</td>
</tr>
<tr>
<td>Type: A (Host Address) (1)</td>
</tr>
<tr>
<td>Class: IN (0x0001)</td>
</tr>
</tbody>
</table>

```
0000 00 21 d7 75 74 00 6c 40 08 98 57 82 08 00 45 00
0010 00 37 16 b7 00 00 40 11 2e c6 ac 1b 05 91 82 84
0020 01 09 81 9b 00 35 00 23 93 65 63 32 01 00 00 01
0030 00 00 00 00 00 00 05 67 6d 61 69 6c 03 63 6f 6d
0040 00 00 01 00 01
```

```
.!.ut.1@ ..W...E.
.7...@. ........
......5.# .ec2....
........gmail.com
```
Many features: typically over **UDP** (can use TCP); query and reply messages with the **same message format**; **length/content encoding of names**; **simple compression**; additional info as **server push**

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions</td>
<td>Number of answer RRs</td>
</tr>
<tr>
<td>Number of authority RRs</td>
<td>Number of additional RRs</td>
</tr>
</tbody>
</table>

- **Identification**
- **Flags**
- **Number of questions**
- **Number of answer RRs**
- **Number of authority RRs**
- **Number of additional RRs**

Discussion: What DNS did Right