CS615 - Aspects of System Administration

HTTPS, TLS, SMTP

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/615/
Team Missions

Red team:
https://is.gd/pbcgc5
https://is.gd/mJoJEV

Black team:
https://is.gd/xCRWDn
https://is.gd/xa2LSp

Blue team:
https://is.gd/onqXl6

Green team:
https://is.gd/7jG0n3
https://is.gd/pzrga0
https://is.gd/o4Gcqm
HTTP

http://ec2-54-82-75-174.compute-1.amazonaws.com/
HTTP

```bash
$ sudo tcpdump -w post.pcap port 80 2>/dev/null &
$ fg
^C
$ sudo chmod a+r post.pcap
```

Now use `tcpdump(1)` to extract the plain text data you sent to the web server from your `pcap` file.
HTTP

14:14:35.348492 IP 172.16.1.20.52941 > 54.160.173.145.80: Flags [P.], seq 1:668,
0x0000: 4500 02cf 0000 4000 4006 a6d3 ac10 0114 E.....@.@... ....
0x0010: 36a0 ad91 cecd 0050 6d61 ffbe ab1f 5284 6......Pma....R.
0x0020: 8018 080a 8d1c 0000 0101 080a 53ec 8097 ..........S...
0x0030: 0000 0001 5040f 5354 2e63 6769 2048 54502f 1..POST/cgi-bin.
0x0040: 312e 310d 0a48 6f743a 2065 6332 2d35 1.1..Host: ec2-5
0x0050: 342d 3160 52476a 6e6f7465 6e742d 4c69657364 4-160-amazona...
0x0060: 6d707465 2d312e 616d617a6f6e732e 636f6d 0d0a 557067726164 1.http-1.amazonaws.com..Upgrade-
0x0070: 696e63757265 733a 2031 0d0a 444e543a 2031 0d0a 436f64696e 6Insecure-Request:
0x0080: 673a 20677a6970 2c 6465666c617465 0a41 636570742d 6gzip,.defl
0x0090: 4c616e675c 61757365726e616d65 6a5f706173776f7264 6-Lang&j_password=
0x00a0: 265f656e7449645f70726f63656564 3d346365636f6f6f 6=&ventId_proceed=

HTTPS, TLS, SMTP

March 23, 2020
HTTPS

```bash
$ </dev/null openssl s_client -connect ec2-54-82-75-174.compute-1.amazonaws.com:443 |
  openssl x509 -text -noout | more

$ sudo tcpdump -w post.pcap port 443 2>/dev/null &
$ fg
^C
$ sudo chmod a+r post.pcap
```

HTTPS, TLS, SMTP  March 23, 2020
HTTPS

HTTPS stands for...

HTTP over SSL.
HTTPS

HTTPS stands for...

HTTP over SSL.

HTTP over TLS.
HTTPS

HTTPS stands for...

HTTP over SSL.

HTTP over TLS.

Secure HTTP.
HTTPS

HTTPS stands for...

HTTP over SSL.

HTTP over TLS.

Secure HTTP.

HTTP Secure.
HTTPS

HTTPS stands for...

HTTP over SSL.

HTTP over TLS.

Secure HTTP.

HTTP Secure.

But it uses TLS. And used to use SSL. Although hopefully not any more. Although probably still.

SSL is dead. Don’t use it. Seriously, don’t.

We should really only call it TLS. HTTP.
TLS

Transport Layer Security

- set of cryptographic protocols
- operates on layer 6 of OSI stack (Presentation Layer) (or 5? 4? 7? none? all?)
- independent of HTTP
- TLS 1.2 (RFC5246) standardized in 2008
- TLS 1.3 (RFC8446) standardized in 2018

Two distinct security mechanisms:

1. encryption of data in transit
2. authentication of parties
TLS

Protocol:
- Client Hello, present list of supported cipher suites
- Server Hello, chosen cipher suite
- Server Certificate
- (Server Key Exchange Message), (Client Certificate Request), (Client Certificate)
- Client Key Exchange Message
- (Certificate Verify)
- (Client Change Cipher Spec), (Server Change Cipher Spec)

See also: https://tls.ulfheim.net/
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0000000000</td>
<td>168.64.7.99</td>
<td>155.246.88.84</td>
<td>TCP</td>
<td>78</td>
<td>Seq=3580837243 Win=27630 Len=0 MSS=1460 WS=0</td>
</tr>
<tr>
<td>2</td>
<td>000000359</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TCP</td>
<td>74</td>
<td>Seq=3580837243 Ack=2770027821 Win=14490</td>
</tr>
<tr>
<td>3</td>
<td>0000006801</td>
<td>168.64.7.99</td>
<td>155.246.88.84</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3580837244 Ack=2770027822 Win=33576</td>
</tr>
<tr>
<td>4</td>
<td>0000010820</td>
<td>168.64.7.99</td>
<td>155.246.88.84</td>
<td>TLSv1.2</td>
<td>583</td>
<td>803 Client Hello</td>
</tr>
<tr>
<td>5</td>
<td>0000013809</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3580837241 Ack=2770027822 Win=15552</td>
</tr>
<tr>
<td>6</td>
<td>0000021535</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TLSv1.2</td>
<td>15</td>
<td>Server Hello</td>
</tr>
<tr>
<td>7</td>
<td>0000022247</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TCP</td>
<td>15</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>8</td>
<td>0000022667</td>
<td>168.64.7.99</td>
<td>155.246.88.84</td>
<td>TCP</td>
<td>66</td>
<td>Seq=3580837243 Ack=2770030719 Win=32128</td>
</tr>
<tr>
<td>9</td>
<td>0000022655</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TLSv1.2</td>
<td>13</td>
<td>Certificate</td>
</tr>
<tr>
<td>10</td>
<td>0000028284</td>
<td>168.64.7.99</td>
<td>155.246.88.84</td>
<td>TLSv1.2</td>
<td>192</td>
<td>192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake</td>
</tr>
<tr>
<td>11</td>
<td>0000033175</td>
<td>155.246.88.84</td>
<td>168.64.7.99</td>
<td>TLSv1.2</td>
<td>117</td>
<td>Change Cipher Spec, Encrypted Handshake Message</td>
</tr>
</tbody>
</table>

**Handshake Protocol:** Client Hello
**Handshake Type:** Client Hello (1)
**Length:** 512

**Random**
GMT Unix Time: Nov 26, 2000 15:19:43.000000000 EST
Random Bytes: 79db8a5297a8394794c8a5e6e9b83c4321edd5f1ebd7...
Session ID Length: 0
Cipher Suites Length: 120
**Cipher Suites (00 suites)**
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc03b)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc03e)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256 (0xc028)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc02f)

<table>
<thead>
<tr>
<th>Time</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>0x0f</td>
<td>0x00</td>
</tr>
<tr>
<td>00001</td>
<td>0x0f</td>
<td>0x01</td>
</tr>
<tr>
<td>00002</td>
<td>0x0f</td>
<td>0x02</td>
</tr>
<tr>
<td>00003</td>
<td>0x0f</td>
<td>0x03</td>
</tr>
<tr>
<td>00004</td>
<td>0x0f</td>
<td>0x04</td>
</tr>
<tr>
<td>00005</td>
<td>0x0f</td>
<td>0x05</td>
</tr>
<tr>
<td>00006</td>
<td>0x0f</td>
<td>0x06</td>
</tr>
<tr>
<td>00007</td>
<td>0x0f</td>
<td>0x07</td>
</tr>
<tr>
<td>00008</td>
<td>0x0f</td>
<td>0x08</td>
</tr>
<tr>
<td>00009</td>
<td>0x0f</td>
<td>0x09</td>
</tr>
<tr>
<td>00010</td>
<td>0x0f</td>
<td>0x0a</td>
</tr>
<tr>
<td>00011</td>
<td>0x0f</td>
<td>0x0b</td>
</tr>
<tr>
<td>00012</td>
<td>0x0f</td>
<td>0x0c</td>
</tr>
<tr>
<td>00013</td>
<td>0x0f</td>
<td>0x0d</td>
</tr>
<tr>
<td>00014</td>
<td>0x0f</td>
<td>0x0e</td>
</tr>
<tr>
<td>00015</td>
<td>0x0f</td>
<td>0x0f</td>
</tr>
<tr>
<td>00016</td>
<td>0x0f</td>
<td>0x10</td>
</tr>
<tr>
<td>00017</td>
<td>0x0f</td>
<td>0x11</td>
</tr>
<tr>
<td>00018</td>
<td>0x0f</td>
<td>0x12</td>
</tr>
<tr>
<td>00019</td>
<td>0x0f</td>
<td>0x13</td>
</tr>
<tr>
<td>00020</td>
<td>0x0f</td>
<td>0x14</td>
</tr>
<tr>
<td>00021</td>
<td>0x0f</td>
<td>0x15</td>
</tr>
<tr>
<td>00022</td>
<td>0x0f</td>
<td>0x16</td>
</tr>
</tbody>
</table>

**Packet Size:** 21
**Display Size:** 21
**Marked:** 0
**Load Time:** 0.00s
**Profile:** Default
$ openssl s_client -connect www.stevens.edu:443

[...] New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Early data was not sent

---

GET / HTTP/1.0

Post-Handshake New Session Ticket arrived:

SSL-Session:
  Protocol : TLSv1.3
  Cipher : TLS_AES_256_GCM_SHA384
  Session-ID: 11A6C0CF6C661080EED2E0A82356F164FFFFB798DF00758E6ABDE35375871480
  Session-ID-ctx:
  Resumption PSK: 48CBBD750915769BB0C86C89DA7E9C0DE0E88311504F847FEFD4CC50E360B538A...
$ openssl s_client -tls1_2 -connect www.stevens.edu:443

[...]

New, TLSv1.2, Cipher is ECDHE-RSA-CHACHA20-POLY1305
Server public key is 2048 bit
Secure Renegotiation IS supported
SSL-Session:
  Protocol : TLSv1.2
  Cipher : ECDHE-RSA-CHACHA20-POLY1305
  Session-ID: 5AEA1C7F5402937F72688473F585FAE0B51FCBE75CB0B214EBAE7C9EAF55BDFF
  Session-ID-ctx: Master-Key: BAE87DF4DFD95DF4539B67178248A13535FE847C8297B36C14E45F573DB020517DB2A
  PSK identity: None
  PSK identity hint: None
  SRP username: None
  TLS session ticket lifetime hint: 64800 (seconds)
  TLS session ticket:
$ openssl s_client -connect www.stevens.edu:443 | \ 
    openssl x509 -text -noout

[...]
Serial Number:
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = US, ST = MI, L = Ann Arbor, O = Internet2, OU = InCommon, 
       CN = InCommon RSA Server CA
Validity
  Not Before: Apr 22 00:00:00 2019 GMT
  Not After : Apr 21 23:59:59 2021 GMT
Subject: C = US, postalCode = 07030, ST = NJ, 
       L = Hoboken, street = Castle Point on Hudson, 
       O = Stevens Institute of Technology, OU = IT, 
       CN = stevens.edu

[...]
X509v3 Subject Alternative Name:
  DNS:stevens.edu, DNS:*:stevens-tech.edu, DNS:*:stevens.edu
TLS Authentication

Use of X.509:

- public key certificates
- certificate revocation lists (CRLs) / Online Certificate Status Protocol (OCSP)
- certificate path validation under a Public Key Infrastructure (PKI)
- certificate chains depend on trust anchors
TLS

1. User / Company generates a *Certificate Signing Request* (CSR), containing:
   - identifying information (distinguished name etc.)
   - signature of data by private key
   - chosen public key
TLS

1. User / Company generates a *Certificate Signing Request* (CSR)

2. CSR submitted to Certificate Authority (CA)
TLS

1. User / Company generates a *Certificate Signing Request* (CSR)

2. CSR submitted to Certificate Authority (CA)

3. CA verifies information
TLS

1. User / Company generates a *Certificate Signing Request* (CSR)

2. CSR submitted to Certificate Authority (CA)

3. CA verifies information

4. CA returns certificate signed with its private key
TLS

1. User / Company generates a *Certificate Signing Request* (CSR)

2. CSR submitted to Certificate Authority (CA)

3. CA verifies information

4. CA returns certificate signed with its private key

5. clients can verify signatures against trusted *root CAs*
TLS

User generates key pair, KU_{pub}, KU_{priv}.

User generates CSR, containing CN=www.example.com, KU_{pub}, signed with KU_{priv}.

User submits CSR to CA.

CA verifies User / ownership of www.example.com.

CA issues certificate, containing CN=www.example.com, KU_{pub}, signed with KU_{priv}.

User installs certificate, KU_{priv}, intermediate on server(s).

Root signed with KRoot_{priv} (i.e. self-signed).

Intermediate signed with KRoot_{priv}.

CA signs cert with KInt_{priv}.

Client requires CA's root cert to be installed and trusted.

Client connects to www.example.com.

Server presents certificate(s).

Client verifies signature on certificate was made with KInt_{priv}, signature on intermediate was made with KRoot_{priv}.

This diagram is in the public domain. Originally made by @jschauma.
### TLS Pitfalls

**Keychain Access**

Click to unlock the System Roots keychain.

<table>
<thead>
<tr>
<th>Keychain Access</th>
<th>Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keychains</td>
<td></td>
</tr>
<tr>
<td>login</td>
<td></td>
</tr>
<tr>
<td>Local Items</td>
<td></td>
</tr>
<tr>
<td>System</td>
<td></td>
</tr>
<tr>
<td>System Roots</td>
<td></td>
</tr>
</tbody>
</table>

- **TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı**
  - Root certificate authority
  - Expires: Friday, December 22, 2017 at 1:37:19 PM Eastern Standard Time
  - This certificate is valid

<table>
<thead>
<tr>
<th>Name</th>
<th>Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thawte Server CA</td>
<td>certificate</td>
</tr>
<tr>
<td>Thawte Timestamping CA</td>
<td>certificate</td>
</tr>
<tr>
<td>TRUST2408 DOES Primary CA</td>
<td>certificate</td>
</tr>
<tr>
<td>Trusted Certificate Services</td>
<td>certificate</td>
</tr>
<tr>
<td>Trustio FPS Root CA</td>
<td>certificate</td>
</tr>
<tr>
<td>TÜBİTAK UMEKAE Kök Sertifika Hizmet Sağlayıcısı - Sürüm 3</td>
<td>certificate</td>
</tr>
<tr>
<td>TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı</td>
<td>certificate</td>
</tr>
<tr>
<td>TWCA Global Root CA</td>
<td>certificate</td>
</tr>
<tr>
<td>TWCA Root Certification Authority</td>
<td>certificate</td>
</tr>
<tr>
<td>UCA Global Root</td>
<td>certificate</td>
</tr>
<tr>
<td>UCA Root</td>
<td>certificate</td>
</tr>
<tr>
<td>UTN - DATAcorp SGC</td>
<td>certificate</td>
</tr>
<tr>
<td>UTN-USERFirst-Client Authentication and Email</td>
<td>certificate</td>
</tr>
<tr>
<td>UTN-USERFirst-Hardware</td>
<td>certificate</td>
</tr>
<tr>
<td>UTN-USERFirst-Network Applications</td>
<td>certificate</td>
</tr>
<tr>
<td>UTN-USERFirst-Object</td>
<td>certificate</td>
</tr>
<tr>
<td>VeriSign Class 1 Public Primary Certification Authority - G3</td>
<td>certificate</td>
</tr>
<tr>
<td>VeriSign Class 2 Public Primary Certification Authority - G3</td>
<td>certificate</td>
</tr>
<tr>
<td>VeriSign Class 3 Public Primary Certification Authority - G3</td>
<td>certificate</td>
</tr>
</tbody>
</table>

195 root CAs on this laptop...
TLS Pitfalls

Just because a site has a valid certificate does not mean it’s a trustworthy site.

https://ec2-54-160-173-145.compute-1.amazonaws.com/

https://www.netmeister.org/tumblr/

https://www.netmeister.org/owa/auth/logon.aspx
TLS Pitfalls

Lack of universal HTTPS exposes users to significant risks; many sites don’t understand the importance of authentication and encryption for non-sensitive content.

https://is.gd/ghiOhU

Middle boxes, often advertized as a security mechanism, are actively harmful to users and prohibit secure protocol development.

In order to serve content, you need to have the private key → privkey available at perimeter and exposed, high-risk systems.

Rotation/renewal of keys requires routine processes, which may further expose the private key.

Control of a CA or a CA’s key grants you near universal powers.
TLS Pitfalls

Complex protocols, buggy implementations, intentional weaknesses and backwards compatibility are just the high level points.

- SSLv2 obsoleted in 1996; 2016: DROWN attack
- SSLv3 obsoleted in 1999; 2014: POODLE attack
- BEAST, CRIME, BREACH, HEARTBLEED, GotoFail...
- obsolete and broken algorithms widely used (RC4, MD5, SHA1, ...)

HTTPS, TLS, SMTP

March 23, 2020
TLS

Additional related topics:
- HSTS and TLS stripping attacks
- HPKP and Trust On First Use (TOFU)
- Certificate Transparency
- Content Security Policy (CSP)
- “Secure” cookies vs. HttpOnly cookies
- attacks on domain name registrars

Security is difficult. More on that in a future lecture.
Hooray!

5 Minute Break
Email... still popular

Bad news, everybody: Slack has not yet replaced email.
Email... still popular

Good news, everybody: Slack has not yet replaced email. (And it’s not going to.)

- 4.6 billion - number of email accounts.
- 269 billion - Average number of email messages per day. That’s 3.1 million emails per second.
- 121 - Average number of emails an office worker receives.
- 42 - Percentage of Americans that check their email in the bathroom.
- 18 - Percentage of Americans that check their email while driving.
- >70 - Percentage of emails that are Spam.
- 99.95 - Percentage of SysAdmins, SREs, and DevOps who rely on email for monitoring.
The Mail System

Divided into:

- *Mail User Agent* or MUA, such as `mutt(1)`, *Mail.app*, *Outlook*, a browser (ugh) ... 
- *Mail Transfer Agent* or MTA, such as `postfix`, `sendmail`, `qmail`, ...
- *Mail Delivery Agent* or MDA, such as `procmail`
- *Access Agent* providing access via *POP*, *IMAP* etc.

In addition, many MUAs nowadays interpret HTML:

- browser now the most common MUA
- facilitates phishing (via link obscuring, logos etc.)
- facilitates tracking (via beacons, cookies)
Sending...

# tcpdump -i xennet0 -w /tmp/t.out port not 22 2>/dev/null &
# mail -s "CS615 - SMTP Exercise" jschauma@netmeister.org -f jschauma@stevens.edu
Hello,

SMTP is so simple!

-Jan
.
EOT

# fg
tcpdump -i xennet0 -w /tmp/t.out port not 22 2>/dev/null
^C
Sending...

# tail -6 /var/log/maillog
Mar 25 14:19:59 ip-10-168-152-198 postfix/pickup[5939]: A76DB2FFC2: uid=0 from=<jschauma@stevens.edu>
Mar 25 14:19:59 ip-10-168-152-198 postfix/cleanup[5564]: A76DB2FFC2: message-id=<20190325141959.A76DB2FFC2@ip-10-168-152-198.ec2.internal>
Mar 25 14:19:59 ip-10-168-152-198 postfix/qmgr[1846]: A76DB2FFC2: from=<jschauma@stevens.edu>, size=386, nrcpt=1 (queue active)
Mar 25 14:20:00 ip-10-168-152-198 postfix/smtp[7163]: A76DB2FFC2: to=<jschauma@netmeister.org>, relay=panix.netmeister.org[166.84.7.99]:25, delay=0.48, delays=0.03/0.01/0.29/0.15, dsn=2.0.0, status=sent (250 2.0.0 Ok: queued as 2223965341)
Mar 25 14:20:00 ip-10-168-152-198 postfix/qmgr[1846]: A76DB2FFC2: removed
# tcpdump -n -t -t smtp-client.pcap port 53
IP 172.16.0.23.53 > 10.168.152.198.63685: 1736 1/0/0 MX panix.netmeister.org. 50 (54)
IP 10.168.152.198.63684 > 172.16.0.23.53: 64083+ A? panix.netmeister.org. (38)
IP 172.16.0.23.53 > 10.168.152.198.63684: 64083 1/0/0 A 166.84.7.99 (54)
IP 10.168.152.198.63683 > 172.16.0.23.53: 16542+ AAAA? panix.netmeister.org. (38)

$ host -t mx netmeister.org
netmeister.org mail is handled by 50 panix.netmeister.org.
$ host panix.netmeister.org
panix.netmeister.org has address 166.84.7.99
$
$ tcpdump -n -t -r smtp-client.pcap 'tcp[tcpflags] & tcp-push != 0 and port 25'

IP 166.84.7.99.25 > 10.168.152.198.65528: Flags [P.], seq 1:41, ack 1
    SMTP: 220 panix.netmeister.org ESMTP Postfix

IP 10.168.152.198.65528 > 166.84.7.99.25: Flags [P.], seq 1:38, ack 41
    SMTP: EHLO ip-10-168-152-198.ec2.internal

IP 166.84.7.99.25 > 10.168.152.198.65528: Flags [P.], seq 41:174, ack 38
    SMTP: 250-panix.netmeister.org

IP 10.168.152.198.65528 > 166.84.7.99.25: Flags [P.], seq 38:159, ack 174
    SMTP: MAIL FROM:<jschauma@stevens.edu> SIZE=386

IP 166.84.7.99.25 > 10.168.152.198.65528: Flags [P.], seq 174:239, ack 159
    SMTP: 250 2.1.0 Ok

IP 10.168.152.198.65528 > 166.84.7.99.25: Flags [P.], seq 159:554, ack 239
    SMTP: Received: by ip-10-168-152-198.ec2.internal (Postfix, from userid 0)

IP 166.84.7.99.25 > 10.168.152.198.65528: Flags [P.], seq 239:290, ack 554
    SMTP: 250 2.0.0 Ok: queued as 2223965341
SMTP Codes

SMTP codes consist of three digits in five classes:

- **1xx** – Mail server has accepted the command, but does not yet take any action. A confirmation message is required.
- **2xx** – Mail server has completed the task successfully without errors.
- **3xx** – Mail server has understood the request, but requires further information to complete it.
- **4xx** – Mail server has encountered a temporary failure. If the command is repeated without any change, it might be completed. Try again, it may help!
- **5xx** – Mail server has encountered a fatal error. Your request can’t be processed.
Sending...

```
$ telnet panix.netmeister.org 25
Trying 2001:470:30:84:e276:63ff:fe72:3900...
telnet: connect to address 2001:470:30:84:e276:63ff:fe72:3900: No route to host
Trying 166.84.7.99...
Connected to panix.netmeister.org.
Escape character is '^[']'.
220 panix.netmeister.org ESMTP Postfix
EHLO ip-10-168-152-198.ec2.internal
250-panix.netmeister.org
[...]
MAIL FROM: <jschauma@stevens.edu>
250 2.1.0 Sender OK
RCPT TO: <jschauma@netmeister.org>
250 2.1.5 Recipient OK
```
Sending...

**DATA**
354 Start mail input; end with <CRLF>.<CRLF>
To: jschauma@netmeister.org
Subject: CS615 - SMTP Exercise
Mon, 25 Mar 2019 14:19:59 +0000 (UTC)
From: Charlie Root <jschauma@stevens.edu>

Hello,

SMTP is so simple!

–Jan
.
250 2.0.0 Ok: queued as 522DF65341
Sending...
Receiving...

$ tcpdump -n -t -r smtp-server.pcap 'tcp[tcpflags] & tcp-push != 0 and port 25'
IP 166.84.7.99.25 > 54.160.173.145.65528: Flags [P.], seq 641894792:641894832, ack 3475020053
  SMTP: 220 panix.netmeister.org ESMTP Postfix
IP 54.160.173.145.65528 > 166.84.7.99.25: Flags [P.], seq 1:38, ack 40
  SMTP: EHLO ip-10-168-152-198.ec2.internal
IP 166.84.7.99.25 > 54.160.173.145.65528: Flags [P.], seq 40:173, ack 38
  SMTP: 250-panix.netmeister.org
IP 54.160.173.145.65528 > 166.84.7.99.25: Flags [P.], seq 38:159, ack 173
  SMTP: MAIL FROM:<jschauma@stevens.edu> SIZE=386
IP 166.84.7.99.25 > 54.160.173.145.65528: Flags [P.], seq 173:238, ack 159
  SMTP: 250 2.1.0 Ok
IP 54.160.173.145.65528 > 166.84.7.99.25: Flags [P.], seq 159:554, ack 238
  SMTP: Received: by ip-10-168-152-198.ec2.internal (Postfix, from userid 0)
IP 166.84.7.99.25 > 54.160.173.145.65528: Flags [P.], seq 238:289, ack 554
  SMTP: 250 2.0.0 Ok: queued as 2223965341
Receiving

```
$ sudo grep 2223965341 /var/log/maillog
<mail.info>Mar 25 10:20:01 panix postfix/smtpd[5089]: 2223965341:
<mail.info>Mar 25 10:20:01 panix postfix/cleanup[10085]: 2223965341:
    message-id=<20190325141959.A76DB2FFC2@ip-10-168-152-198.ec2.internal>
<mail.info>Mar 25 10:20:01 panix postfix/qmgr[1932]: 2223965341:
    from=<jschauma@stevens.edu>, size=627, nrcpt=1 (queue active)
<mail.info>Mar 25 10:20:21 panix postfix/pipe[10375]: 2223965341:
    to=<jschauma@netmeister.org>, relay=spamassassin, delay=20, delays=0.15/0/0/20,
    dsn=2.0.0, status=sent (delivered via spamassassin service)
<mail.info>Mar 25 10:20:21 panix postfix/qmgr[1932]: 2223965341: removed
```
Receiving

<table>
<thead>
<tr>
<th>No.</th>
<th>Source Port</th>
<th>Destination Port</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>34.229.283.237</td>
<td>166.84.7.99</td>
<td>TCP</td>
<td>66</td>
<td>87</td>
</tr>
<tr>
<td>4</td>
<td>166.84.7.99</td>
<td>34.229.283.237</td>
<td>DNS</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>166.84.7.99</td>
<td>166.84.7.99</td>
<td>DNS</td>
<td>364</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>166.84.7.99</td>
<td>166.84.7.99</td>
<td>DNS</td>
<td>180</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>166.84.7.99</td>
<td>166.84.7.99</td>
<td>DNS</td>
<td>544</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>34.229.283.237</td>
<td>166.84.7.99</td>
<td>SMTP</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>166.84.7.99</td>
<td>34.229.283.237</td>
<td>SMTP</td>
<td>199</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>34.229.283.237</td>
<td>166.84.7.99</td>
<td>SMTP</td>
<td>187</td>
<td>1</td>
</tr>
</tbody>
</table>

Frame 11: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits) on interface 0
- Internet Protocol Version 4, Src: 34.229.283.237, Dst: 166.84.7.99

**Simple Mail Transfer Protocol**
- **Command Line:** MAIL FROM:<jschauma@stevens.edu> SIZE=384
- **Command:** MAIL
- **Request parameter:** FROM:<jschauma@stevens.edu> SIZE=384
- **Command Line:** RCPT TO:<jschauma@netmeister.org> SIZE=384
- **Command:** RCPT
- **Request parameter:** TO:<jschauma@netmeister.org> SIZE=384
- **Command Line:** DATA
- **Command:** DATA

---

HTTPS, TLS, SMTP

March 23, 2020
Receiving...

Date: Mon, 25 Mar 2019 14:19:59 +0000 (UTC)
From: Charlie Root <jschauma@stevens.edu>
To: jschauma@netmeister.org
Subject: CS615 - SMTP Exercise

Hello,

SMTP is so simple!

-Jan
STARTSSL

EHLO ec2-54-160-173-145.compute-1.amazonaws.com
250-panix.netmeister.org
250-PIPELINING
250-SIZE 10240000
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
STARTTLS
220 2.0.0 Ready to start TLS
now what?
Connection closed by foreign host.
STARTSSL

$ openssl s_client -starttls smtp -crlf -connect panix.netmeister.org:25
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 4096 bit
SSL-Session:
    Protocol : TLSv1.2
    Cipher : ECDHE-RSA-AES256-GCM-SHA384
[...]
helo ec2-54-160-173-145.compute-1.amazonaws.com
[...]

HTTPS, TLS, SMTP
STARTTLS

HTTPS, TLS, SMTP

March 23, 2020
STARTTLS is Opportunistic Encryption

- MitM can strip STARTTLS
- Should failure to verify certificate lead to mail to being delivered?
- DNS-Based Authentication of Named Entities (DANE) (RFC7672)
- SMTP MTA Strict Transport Security (MTA-STS) (RFC8461)

```
$ host -t txt _mta-sts.yahoo.com
_mta-sts.yahoo.com descriptive text "v=STSv1; id=20161109010200Z;"
$ curl https://mta-sts.yahoo.com/.well-known/mta-sts.txt
version: STSv1
mode: testing
mx: *.am0.yahoodns.net
mx: *.mail.gm0.yahoodns.net
mx: *.mail.am0.yahoodns.net
max_age: 86400
```
Hello,

SMTP is so simple!

-Jan
Anatomy of an email message

An email consists of:

- mandatory headers (such as "From ", "Delivered-To: ", ...)
- optional headers (such as "From: ", "To: ", "Subject: ", ...)
- the body of the message
  - content independent of SMTP
  - Multipurpose Internet Mail Extensions (MIME) enables non-ascii, multipart, encodings, ...
Receiving...

From jschauma@stevens.edu  Mon Mar 25 10:20:21 2019
Return-Path: <jschauma@stevens.edu>
X-Original-To: jschauma@netmeister.org
Delivered-To: jschauma@netmeister.org
Received: by panix.netmeister.org (Postfix, from userid 1004)
    id 0E9C0654CE; Mon, 25 Mar 2019 10:20:21 -0400 (EDT)
X-Spam-Checker-Version: SpamAssassin 3.4.2 (2018-09-13) on panix.netmeister.org
X-Spam-Level:
X-Spam-Status: No, score=0.5 required=5.0 tests=BAYES_05,RDNS_DYNAMIC
    autolearn=no autolearn_force=no version=3.4.2
Received: from ip-10-168-152-198.ec2.internal (ec2-54-160-173-145.compute-1.amazonaws.co
    by panix.netmeister.org (Postfix) with ESMTP id 2223965341
    for <jschauma@netmeister.org>; Mon, 25 Mar 2019 10:20:01 -0400 (EDT)
Received: by ip-10-168-152-198.ec2.internal (Postfix, from userid 0)
    id A76DB2FFC2; Mon, 25 Mar 2019 14:19:59 +0000 (UTC)
To: jschauma@netmeister.org
Subject: CS615 - SMTP Exercise
Message-Id: <20190325141959.A76DB2FFC2@ip-10-168-152-198.ec2.internal>
Date: Mon, 25 Mar 2019 14:19:59 +0000 (UTC)
From: jschauma@stevens.edu (Charlie Root)
Status: RO
Content-Length: 33
Lines: 5
Authenticity and SPAM

https://www.youtube.com/watch?v=_bW4vEo1F4E
Relaying mail

$ telnet stevens-edu.mail.protection.outlook.com 25
Trying 104.47.36.36...
Connected to stevens-edu.mail.protection.outlook.com.
Escape character is '^[']'.
220 SN1NAM02FT055.mail.protection.outlook.com Microsoft ESMTP MAIL Service
ready at Mon, 23 Mar 2020 16:06:29 +0000
EHLO localhost
250-SN1NAM02FT055.mail.protection.outlook.com Hello [54.82.75.174]
MAIL FROM: <leaks@whitehouse.gov>
250 2.1.0 Sender OK
RCPT TO: <fauci@nih.gov>
550 5.7.64 TenantAttribution; Relay Access Denied
[SN1NAM02FT055.eop-nam02.prod.protection.outlook.com]
quit
221 2.0.0 Service closing transmission channel
Connection closed by foreign host.
Authenticity and SPAM

220 panix.netmeister.org ESMTP Postfix
EHLO ec2-54-160-173-145.compute-1.amazonaws.com
250 panix.netmeister.org
MAIL FROM: <barack@obama.org>
250 2.1.0 Ok
RCPT TO: <jschauma@netmeister.org>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: "Barack Obama" <barack@obama.org>
To: "Jan Schaumann" <jschauma@stevens.edu>
Subject: Friday

Yo,

Party at my house.
BYOB.

-B

250 2.0.0 Ok: queued as A1D5D65341
Authenticity

Date: Mon, 25 Mar 2019 13:09:06 -0400 (EDT)
From: Barack Obama <barack@obama.org>
To: Jan Schaumann <jschauma@stevens.edu>
Subject: Friday

Yo,

Party at my house.
BYOB.

-B
$ tail -f /var/log/maillog

<mail.info>Mar 25 13:08:31 panix postfix/smtpd[15759]:
    connect from ec2-54-160-173-145.compute-1.amazonaws.com[54.160.173.145]
<mail.info>Mar 25 13:08:38 panix postfix/smtpd[15759]: A1D5D65341:
<mail.info>Mar 25 13:08:46 panix postfix/cleanup[15274]: A1D5D65341:
    message-id=<>  
<mail.info>Mar 25 13:08:46 panix postfix/qmgr[1932]: A1D5D65341:
    from=<barack@obama.org>, size=396, nrcpt=1 (queue active)
<mail.info>Mar 25 13:08:46 panix spamd[18739]: spamd:
    clean message (4.8/5.0) for spamd:1004 in 0.2 seconds, 383 bytes.
<mail.info>Mar 25 13:08:46 panix spamd[18739]: spamd:
    result: . 4 - BAYES_40,HELO_DYNAMIC_IPADDR,MISSING_DATE,MISSING_MID,RDNS_DYNAMIC
    scantime=0.2,size=383,user=spamd,uid=1004,required_score=5.0,
    rhost=::1,raddr=::1,rport=59084,mid=(unknown),bayes=0.258339,autolearn=no
    autolearn_force=no
<mail.info>Mar 25 13:08:48 panix postfix/smtpd[15759]:
    disconnect from ec2-54-160-173-145.compute-1.amazonaws.com[54.160.173.145]
<mail.info>Mar 25 13:09:06 panix postfix/qmgr[1932]: A1D5D65341: removed
**Authenticity and SPAM**

```
$ tcpdump -n -t -r smtp-spam-server.pcap port 53
IP 166.84.7.99.60228 > 166.84.67.2.53: 10483+ PTR? 145.173.160.54.in-addr.arpa. (45)
IP 166.84.67.2.53 > 166.84.7.99.60228: 10483 1/5/6 PTR ec2-54-160-173-145.compute-1.amazonaws.com.
IP 166.84.7.99.60227 > 166.84.67.2.53: 8466+ A? ec2-54-160-173-145.compute-1.amazonaws.com.
IP 166.84.67.2.53 > 166.84.7.99.60227: 8466 1/5/5 A 54.160.173.145 (502)
IP 166.84.7.99.60226 > 166.84.67.2.53: 23794+ MX? obama.org. (27)
IP 166.84.67.2.53 > 166.84.7.99.60226: 23794 5/2/12 MX aspmx.l.google.com. 1,
    MX aspmx3.googlemail.com. 10, MX aspmx2.googlemail.com. 10,
    MX alt2.aspmx.l.google.com. 5, MX alt1.aspmx.l.google.com. 5 (501)
IP 166.84.7.99.60225 > 166.84.67.2.53: 22084+ A? ec2-54-160-173-145.compute-1.amazonaws.com.
IP 166.84.67.2.53 > 166.84.7.99.60225: 22084 1/5/5 A 54.160.173.145 (502)
IP 166.84.7.99.60224 > 166.84.67.2.53: 13128+ A? 145.173.160.54.sbl.spamhaus.org. (49)
IP 166.84.67.2.53 > 166.84.7.99.60224: 13128 NXDomain 0/1/0 (113)
IP 166.84.7.99.56261 > 166.84.67.2.53: 40648+ [1au] A? 145.173.160.54.bl.mailspike.net.
IP 166.84.7.99.56261 > 166.84.67.2.53: 15871+ [1au] A? 145.173.160.54.dnsbl.sorbs.net.
IP 166.84.7.99.56261 > 166.84.67.2.53: 62257+ [1au] TXT? 145.173.160.54.sa-accredit.bondedsenders.com.
IP 166.84.7.99.56261 > 166.84.67.2.53: 59439+ [1au] A? 145.173.160.54.iadb.isipp.com.
IP 166.84.67.2.53 > 166.84.7.99.56261: 15871 NXDomain 0/1/1 (115)
IP 166.84.7.99.56261 > 166.84.67.2.53: 21500+ [1au] A? 145.173.160.54.bl.score.senderscore.com.
IP 166.84.7.99.56261 > 166.84.67.2.53: 4312+ [1au] A? 145.173.160.54.zen.spamhaus.org.
IP 166.84.7.99.56261 > 166.84.67.2.53: 59439 NXDomain 0/1/1 (105)
IP 166.84.67.2.53 > 166.84.7.99.56261: 21500 NXDomain 0/1/1 (130)
```

HTTPS, TLS, SMTP

March 23, 2020
IP 166.84.7.99.56261 > 166.84.67.2.53: 33325+ [1au] A? 145.173.160.54.list.dnswl.org. 
IP 166.84.7.99.56261 > 166.84.67.2.53: 60189+ [1au] TXT? 145.173.160.54.bl.spamcop.net. 
IP 166.84.67.2.53 > 166.84.7.99.56261: 33325 NXDomain 0/1/1 (106) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 63286 NXDomain 0/1/1 (109) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 4312 NXDomain 0/1/1 (124) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 62257 NXDomain 0/0/1 (66) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 33947 NXDomain 0/0/1 (71) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 60189 NXDomain 0/1/1 (111) 
IP 166.84.7.99.56261 > 166.84.67.2.53: 8981+ [1au] TXT? _adsp._domainkey.obama.org. (55) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 8981 0/1/1 (117) 
IP 166.84.7.99.56261 > 166.84.67.2.53: 19917+ [1au] MX? obama.org. (38) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 19917 5/2/14 MX alt2.aspmx.l.google.com. 5, MX 
IP 166.84.67.2.53 > 166.84.7.99.56261: 35638 0/1/1 (139) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 40648 NXDomain 0/0/1 (60) 
IP 166.84.67.2.53 > 166.84.7.99.56261: 6046 NXDomain 0/0/1 (60)
Authenticity and SPAM

HTTPS, TLS, SMTP

March 23, 2020
**Sender Policy Framework**

**SPF (RFC7208)** can help detect email spoofing by identifying the list of allowed sending MXs by way of specifically formatted **TXT** records.

```
$ host -t txt obama.org | grep spf
obama.org descriptive text "v=spf1 include:_spf.salesforce.com include:_spf.google.com
include:bounce.bluestatedigital.com include:sendgrid.net ~all"

$ host -t txt yahoo.com | grep spf
yahoo.com descriptive text "v=spf1 redirect=_spf.mail.yahoo.com"

$ host -t txt _spf.mail.yahoo.com | grep spf
_spf.mail.yahoo.com descriptive text "v=spf1 ptr:yahoo.com ptr:yahoo.net ?all"

$ host -t txt netmeister.org | grep spf
netmeister.org descriptive text "v=spf1 a mx ~all"
```

HTTPS, TLS, SMTP

March 23, 2020
Sender Policy Framework

Softfail:

$ host -t txt obama.org | grep spf
obama.org descriptive text "v=spf1 include:_spf.salesforce.com include:_spf.google.com include:bounce.bluestatedigital.com include:sendgrid.net ~all"

Authentication-Results: spf=softfail (sender IP is 54.160.173.145)
  smtp.mailfrom=obama.org; stevens.edu; dkim=none (message not signed)
  header.d=none;stevens.edu; dmarc=fail action=oreject
  header.from=obama.org;compauth=fail reason=000
Received-SPF: SoftFail (protection.outlook.com: domain of transitioning
  obama.org discourages use of 54.160.173.145 as permitted sender)
Sender Policy Framework

Hardfail:

$ host -t txt stevens.edu | grep spf
stevens.edu descriptive text "v=spf1 ip4:155.246.0.0/16 include:_netblocks.google.com include:_netblocks2.google.com include:spf.protection.outlook.com include:_spf.acquia.com ip4:52.35.7.203 ip4:74.208.4.192/26 " " ip4:66.132.220.97 ip4:198.187.196.100 ip4:66.132.220.95 -all"

Authentication-Results: spf=fail (sender IP is 54.160.173.145)
smtp.mailfrom=stevens.edu; stevens.edu; dkim=none (message not signed)
header.d=none;stevens.edu; dmarc=none action=none
header.from=stevens.edu;compauth=fail reason=601

Received-SPF: Fail (protection.outlook.com: domain of stevens.edu does not designate 54.160.173.145 as permitted sender)
receiver=protection.outlook.com; client-ip=54.160.173.145;
helo=ip-10-168-152-198.ec2.internal;
DomainKeys Identified Mail aka **DKIM**

*DKIM* can help detect email spoofing by providing a digital signature across parts of the message.

- developed by Yahoo with help from Cisco, PGP, and Sendmail
- RFC4871, published in 2007, updated via RFC6376
- DKIM-Signature headers
- more DNS TXT records (<s>._domainkey.<d>) – we really rely on and trust DNS quite a bit, don’t we?
DKIM Example

DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;
   d=stevens0.onmicrosoft.com; s=selector1-stevens-edu;
   h=From:Date:Subject:Message-ID:Content-Type:
   MIME-Version:X-MS-Exchange-SenderADCheck;
   bh=JACUpIBf890+LLb3naV0x1KcKzH82I+/G5T/iFkJ2A=;
   b=Qa4evi5FIY6z+5i8B70m0wxLIFwh5cVPRLFxhoorepLJ1q5/LfKdouIam6+MXhXj1u1EDmG
   jzeVDXu45xjrgkqctUrjE/Ykz5/6mEGLeVb8s4t56FNGKPKiz3UCZ4+ojqHt8tMw0pn8o675Kwa68...

$ host -t txt selector1-stevens-edu._domainkey.stevens0.onmicrosoft.com
selector1-stevens-edu._domainkey.stevens0.onmicrosoft.com
descriptive text "v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUA4GNADCBiQKkgQCl/
JSw4q2rARSbh/vPn1mOmpitEG2PsUz59tT0jt5R4QAsvKyaJAmtnBQXtxZiVakZDTeIKY9gpZ4
lvL0o7FSNeUsxZHkQZolKm+f6q6Zipdag9zIS+R0a9DC2AmIqXX6g14TkIx0prJgAv1D57nCGyX8L
io4pVfFLK61CYTwIDAQAB; n=1024,1452130342,1"
Domain-based Message Authentication, Reporting and Conformance

**DMARC** provides a policy of which validation mechanisms should be employed for a given domain.

- RFC7489
- uses SPF and DKIM
- more DNS TXT records (_dmarc.<domain>)
- extends across From and From: alignment
- provides report mechanism

```
$ dig +short txt _dmarc.yahoo.com
"v=DMARC1; p=reject; pct=100; rua=mailto:dmarc_y_rua@yahoo.com;"
```
DMARC in action

$ telnet gmail-smtp-in.l.google.com 25
Trying 172.217.197.27...
Connected to gmail-smtp-in.l.google.com.
Escape character is '^]'.
220 mx.google.com ESMTP q16si1000312qtb.313 - gsmtp
EHLO ec2-54-160-173-145.compute-1.amazonaws.com
250 mx.google.com at your service
MAIL FROM: <jschauma@yahoo.com>
250 2.1.0 OK q16si1000312qtb.313 - gsmtp
RCPT TO: <jschauma@gmail.com>
250 2.1.5 OK q16si1000312qtb.313 - gsmtp
DATA
354 Go ahead q16si1000312qtb.313 - gsmtp
Subject: DMARC fail
From: jschauma@yahoo.com

This should fail.

550-5.7.1 Unauthenticated email from yahoo.com is not accepted due to domain’s
550-5.7.1 DMARC policy. Please contact the administrator of yahoo.com domain if
550-5.7.1 this was a legitimate mail. Please visit 550-5.7.1
https://support.google.com/mail/answer/2451690 to learn about the
550 5.7.1 DMARC initiative. q16si1000312qtb.313 - gsmtp
SMTP is a Simple Mail Transfer Protocol.

- TCP port 25
- DNS MX records
- mail may be relayed or processed by many servers in transit
- transport is in clear text
- STARTTLS may provide (opportunistic) transport encryption
- SPAM controls may include DNS lookups, bayesian scoring, ...
- authenticity not guaranteed, although DMARC, DKIM, SPF can help
Service Considerations

- outsourcing versus in-house
- privacy considerations
- spam protections
- phishing protections
- mail delivery cannons for notifications vs. spam lists
- high volume traffic demands fine-tuned systems
- high volume traffic implications on logging

See also:

- https://is.gd/JQp1zM
- https://is.gd/cXyrwX
- https://is.gd/o6Y5f8
Reading

SMTP

- DKIM: https://is.gd/VnC09f,
Reading

HTTPS / TLS:

- RFC5246 (TLS 1.2) and RFC6176 (prohibiting SSL)
- RFC8446 (TLS 1.3)
- [https://bugzilla.mozilla.org/show_bug.cgi?id=647959](https://bugzilla.mozilla.org/show_bug.cgi?id=647959)
- [https://cabforum.org](https://cabforum.org)
- [https://tls.ulfheim.net/](https://tls.ulfheim.net/)
- [https://tls13.ulfheim.net/](https://tls13.ulfheim.net/)