Communication Protocols and Internet Architectures

Harvard University

Lecture #10

Instructor: Len Evenchik
cs40@evenchik.com or evenchik@fas.harvard.edu

ALIGHLSOD1701

Lecture Agenda

- Course Logistics
- Q&A and Topics from Last Week
- Application Layer Protocols
- Email Protocols (SMTP) and Architecture
- Network and System Security (part 1)
- Cryptography
- Hashing
- Authentication
- One Minute Wrap-Up

Course Logistics

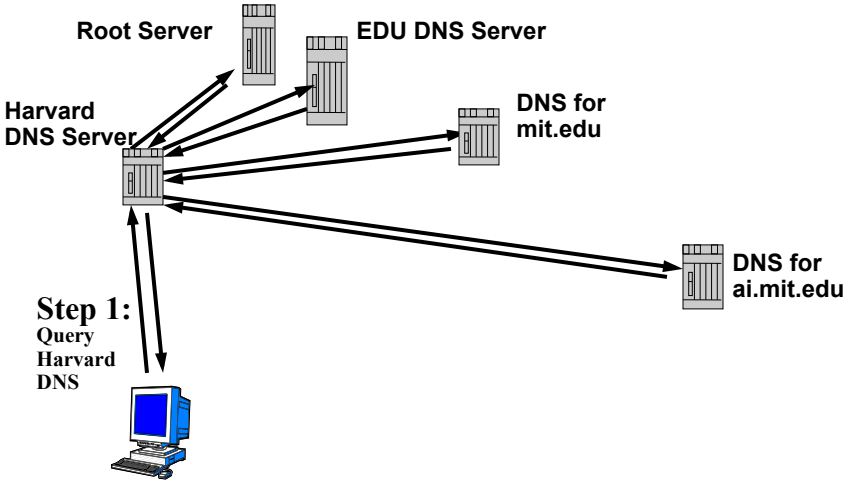
Course Logistics

- Midterm – The exam is being graded and will be returned via the course website.
- Upcoming Guest Lectures
- Homework #4 has been posted.
- Always check the weekly course information sheet for any updated schedule information for section meetings.
- **Please submit a one minute wrap-up each week. Thank You!**

Q&A
Topics from Last Week

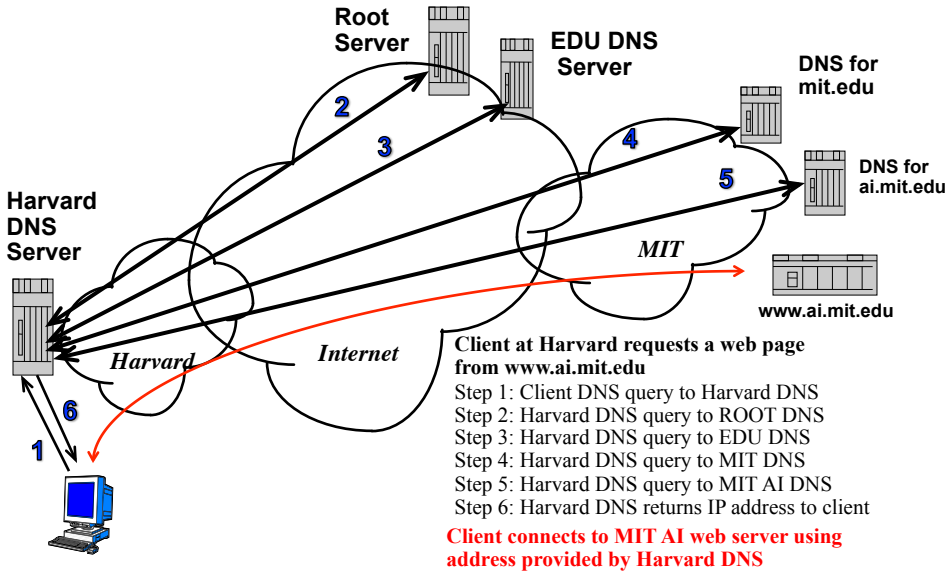
DNS Address Resolution

DNS Address Resolution



Machine at Harvard with a web request for www.ai.mit.edu

DNS Address Resolution



Client at Harvard requests a web page from www.ai.mit.edu

Step 1: Client DNS query to Harvard DNS
Step 2: Harvard DNS query to ROOT DNS
Step 3: Harvard DNS query to EDU DNS
Step 4: Harvard DNS query to MIT DNS
Step 5: Harvard DNS query to MIT AI DNS
Step 6: Harvard DNS returns IP address to client

Client connects to MIT AI web server using address provided by Harvard DNS

dig +norec www.csail.mit.edu +trace
Part 1 of 2

```
Cs40mac:$ dig +norec www.csail.mit.edu +trace | more

; <<>> DiG 9.8.3-P1 <<>> +norec www.csail.mit.edu +trace
;; global options: +cmd
.                497572  IN      NS      a.root-servers.net.
.                497572  IN      NS      b.root-servers.net.
.                497572  skipped some lines
.                497572  IN      NS      d.root-servers.net.
;; Received 508 bytes from 75.75.75.75#53(75.75.75.75) in 41 ms

edu.             172800  IN      NS      a.edu-servers.net.
edu.             172800  skipped some lines
edu.             172800  IN      NS      l.edu-servers.net.
;; Received 270 bytes from 192.203.230.10#53 in 40 ms

mit.edu.         172800  IN      NS      usw2.akam.net.
mit.edu.         172800  skipped some lines
mit.edu.         172800  IN      NS      use5.akam.net.
;; Received 414 bytes from 192.5.6.30#53in 15 ms
```


dig +norec www.csail.mit.edu +trace
Part 1 of 2

SEE previous page for initial steps

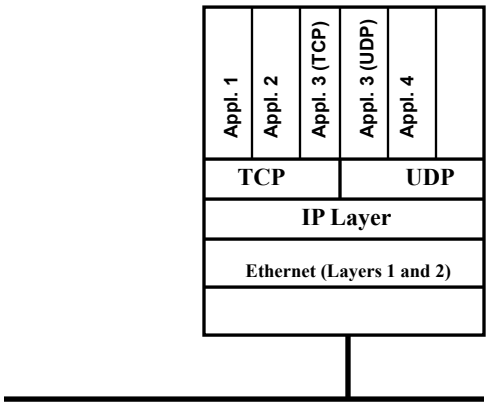
```
cs40mac: dig +norec www.csail.mit.edu +trace | more

csail.mit.edu.      1800    IN      NS      auth-ns3.csail.mit.edu.
csail.mit.edu.      1800    skipped some lines
csail.mit.edu.      1800    IN      NS      auth-ns0.csail.mit.edu.
;; Received 191 bytes from 95.100.175.64#53(95.100.175.64) in 87 ms

www.csail.mit.edu.  1800    IN      A       128.30.2.155
;; Received 51 bytes from 18.24.0.120#53(18.24.0.120) in 10 ms
```

dig +norec www.oxford.edu +trace

```
cs40ac$ dig +norec www.oxford.edu +trace

; <<>> DiG 9.6-ESV-R4-P3 <<>> +norec www.oxford.edu
.                       252830 IN       NS      a.root-servers.net.
.                       252830 IN    skipped some lines
.                       252830 IN       NS      l.root-servers.net.
;; Received 228 bytes from 140.247.233.163#53 in 18 ms

edu.                    172800 IN       NS      a.edu-servers.net.
edu.                    172800 IN    skipped some lines
edu.                    172800 IN       NS      l.edu-servers.net.
;; Received 267 bytes from 128.63.2.53#53(h.root-servers.net)

oxford.edu.             172800 IN       NS      dns0.ox.ac.uk.
oxford.edu.             172800 IN       NS      dns2.ox.ac.uk.
;; Received 78 bytes from 192.5.6.30#53(a.edu-servers.net) in 39 ms

www.oxford.edu.              3600    IN      A       163.1.0.90
oxford.edu.             86400   IN      NS      dns0.ox.ac.uk.
;; Received 126 bytes from 163.1.2.190#53(dns2.ox.ac.uk) in 88 ms
```

Application Layer Protocols

Application Layer Software Schematic



Email Protocols

SMTP Electronic Mail (1)

- Email and its derivative applications drove the growth of the original ARPAnet and the Internet, and most corporate networks.
- Mail systems provide for the delayed delivery of messages and mail forwarding. Mail is not real time.
- There is a difference between the format of the email message and the protocol that is used to deliver the message.
- Mail is comprised of three parts: the envelope, the headers and the body. **The headers and the body together make up the email message.** All three originally used simple ASCII characters.

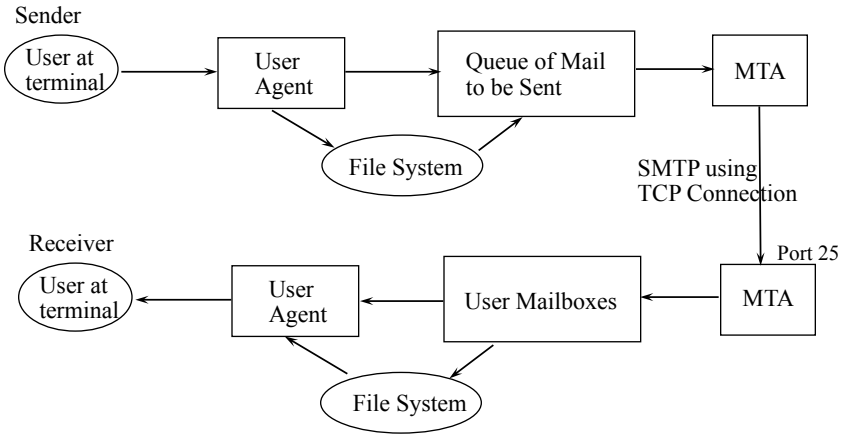
SMTP Electronic Mail (2)

- RFC 5321 (October 2008) describes the Simple Mail Transfer Protocol. This obsoletes RFC 2821 which updated the original RFC 821.
- SMTP uses a TCP connection for email transport.
- RFC 5322 describes the format of mail messages. This obsoletes RFC 2822 which updated the original RFC 822
- SMTP mail servers are found via MX records in DNS.

SMTP Electronic Mail (3)

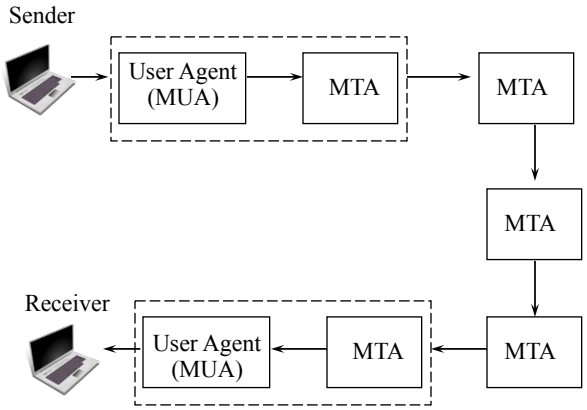
- SMTP is a very simple protocol.
- In the beginning, email was (not surprisingly) text based but MIME extended the functionality to images, audio, video, etc., etc. However, many of the details of current email systems can be better understood if you remember the text based nature of the original protocol.
- In the beginning, email was not typically encrypted (except for military applications.) A lot of work is being done today on secure email but we will not have time to discuss it.
- We will discuss the basic SMTP protocol. Extended SMTP (ESMTP) is now commonly used and it offers more flexibility and additional functionality

Mail System Architecture from the 1980s



Source and date: unknown, but circa 1980s

Simplified Mail System Architecture

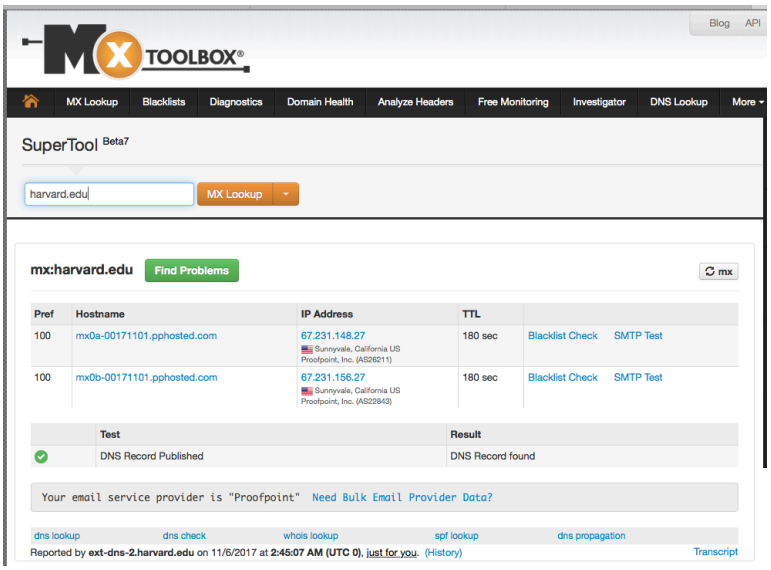


Finding a Network Resource such as a Mail Server or VoIP Server via DNS

(This is different than finding an IP address for a name.)

- "A" records provide a mapping between names and addresses. This is what you would expect the DNS to handle. IPv6 uses AAAA.
- But how do you find a resource such a mail server for an organization when you don't know the specific name of the server?
- For example, email to webmaster@harvard.edu must be delivered to the mail server for Harvard, even though you do not know the name (or IP address) of the specific mail server that handles incoming mail.
- **The answer, as previously discussed, is the MX record.**

MX Lookup at https://mxtoolbox.com
(Of course this can also be done via DIG)



Simplified SMTP Procedure

>>> HELO Alpha.EDU
250 Beta.COM Hello Alpha.EDU, pleased to meet you
>>> MAIL FROM:<Smith@Alpha.EDU>
250 OK
>>> RCPT TO:<Jones@Beta.COM>
250 OK
>>> RCPT TO:<Green@Beta.COM>
550 No such user here
>>> DATA
354 Start mail input; end with <CRLF>.<CRLF>
>>> headers go here
>>>
>>> blah, blah, message body goes here
>>> blah, blah, more message
>>> <CRLF>.<CRLF>
250 OK
>>> QUIT
221 Beta.COM delivering mail for you

Example: Comer Textbook

Reply Code Meanings

Code	Description
1yz	Positive preliminary reply, another reply to be sent
2yz	Positive completion reply, a new command can be sent
3yz	Positive intermediate reply, the command has been accepted but another command must be sent
4yz	Transient negative completion reply
5yz	Permanent negative completion reply
x0z	Syntax error
x1z	Information
x2z	Replies referring to the control or data connections
x3z	Authentication and accounting
x4z	Unspecified
x5z	Filesystem status

Typical Reply Codes with Possible Message String

125 - Data connection already open, transfer starting

250 – OK

331 - Username OK, password required

452 - Error writing file

500 - Syntax error, unrecognized command

501 - Syntax error, invalid arguments

Watching the Exchange of SMTP Messages

- One MTA connects to another MTA to deliver an email message by opening a TCP connection to port 25 on the remote MTA. This would be an unencrypted connection, and today, an encrypted connection could use TLS and other mechanisms.
- In other words, opening a TCP connection to port 25 allows you to send an email message, given that the message is properly formatted.
- TELNET is a long standing approach to doing this for mail system debugging. See:
  - http://ubuntuwiki.net/index.php/SMTP,_testing_via_Telnet
  - https://technet.microsoft.com/en-us/library/bb123686(v=exchg.160).aspx
  - https://www.cisco.com/c/en/us/support/docs/security/email-security-appliance/118234-technote-esa-00.html

Sending Email (a)

(Simple example using Telnet connection)

```
ls03:~ % telnet mail.dce.harvard.edu 25
Trying 140.247.197.xxx...

Connected to mail.dce.harvard.edu (140.247.197.xxx).
Escape character is '^]'.
220 mail.dce.harvard.edu ESMTP Exim Mon,
        24 Oct 2017 18:25:54 -0500
```


Sending Email (b)

(Simple example using Telnet connection)

```
ls03:~ %
ls03:~ % telnet mail.dce.harvard.edu 25
Trying 140.247.197.235...
Connected to mail.dce.harvard.edu (140.247.197.235).
Escape character is '^]'.
220 mail.dce.harvard.edu ESMTP
        Exim Mon, 24 Oct 2017 18:25:54 -0500

HELO somemachine.edu
250
MAIL FROM:<le@harvard.edu>
250 <le@harvard.edu> is syntactically correct

RCPT TO:<cscie40@mail.dce.harvard.edu>
250 <cscie40@mail.dce.harvard.edu> verified

DATA
```

Sending Email (c)

```
220 mail.dce.harvard.edu ESMTP Exim Mon, 24 Oct 2016 18:25:54 -0500
MAIL FROM:<le@harvard.edu>
250 <le@harvard.edu> is syntactically correct
RCPT TO:<csci-40@mail.dce.harvard.edu>
250 <csci-40@mail.dce.harvard.edu> verified
DATA
354 Enter message, ending with "." on a line by itself
From: Len at Lectern
To: The TAs in the course
Date: Wed, Dec 1, 1901
Re: Planning for the midterm

Dear TAs,

Should we include anything on the exam on this
new thing called a telephone?

... Len
.
250 OK id=1AOQ7C-0000CR-00
```


Sending Email (d)
Mail as Delivered (headers off)

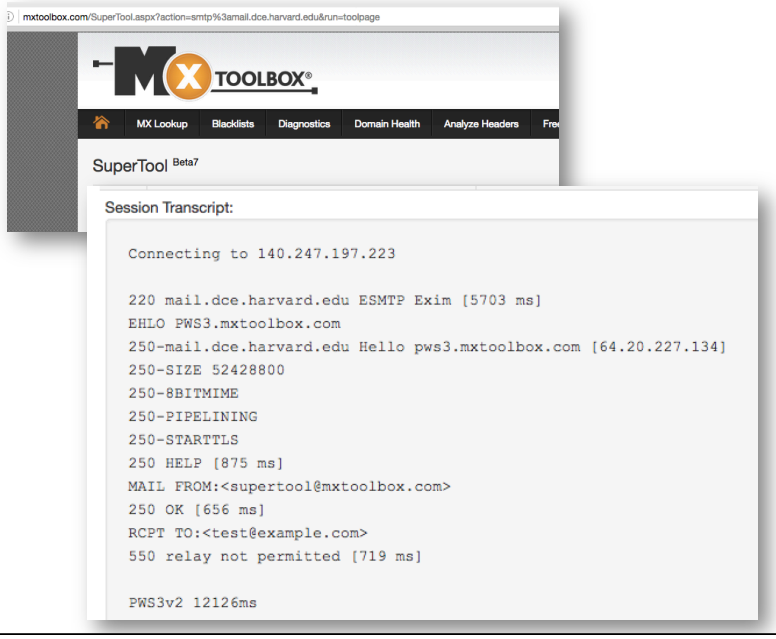
```
Date: Wed, Dec 1, 1901 18:29:29 -0500
From: Len at Lectern
To: The TAs in the course

Dear TAs,

Should we include anything on the midterm on this
new thing called a telephone?

.. Len
```

Sending Email (e)
Mail as Delivered (headers on)

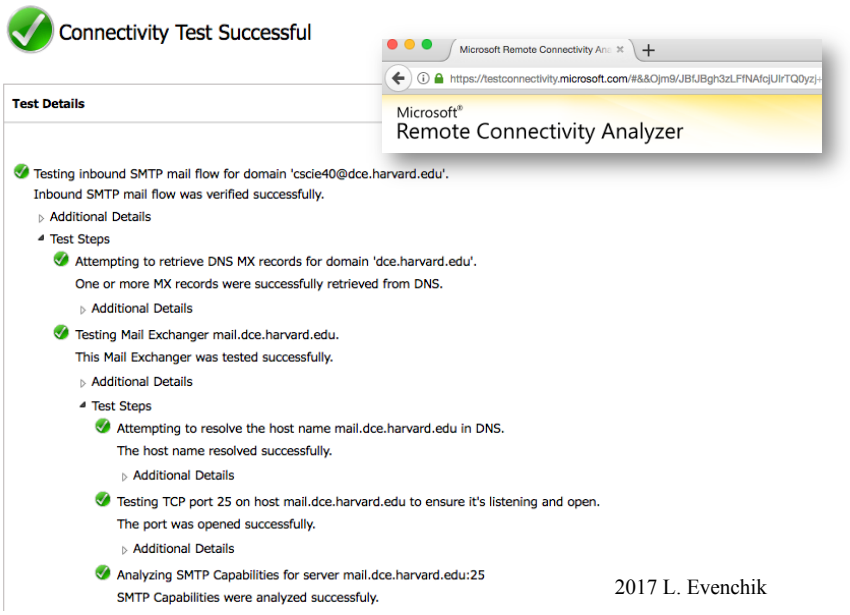
Return-path: <le@harvard.edu>
Envelope-to: csci-40@mail.dce.harvard.edu
Delivery-date: Mon, 24 Oct 2017 18:31:09 -0500
Received: from ls03.fas.harvard.edu [140.247.34.xxx] (evenchik)
by mail.dce.harvard.edu with smtp (Exim)
for csci-40@mail.dce.harvard.edu
id 1AOQ7C-0000CR-00; Mon, 24 Oct 2017 18:29:29 -0500
From: Len at Lectern
To: The TAs in the course
Date: Wed, Dec 1, 1901
Re: Planning for the midterm
Message-Id: <E1AOQ7C-0000CR-00@barkley.dce.harvard.edu>
Date: Mon, 24 Oct 2016 18:29:29 -0500

Dear TAs,
Should we include anything on the midterm on this
new thing called a telephone?
.. Len

Email Test Tool at http://mxtoolbox.com/



Email Test Tool at https://testconnectivity.microsoft.com//



2017 L. Evenchik

Email Delivery Problems

What can happen when the destination mail system is not available?

----- The following addresses had transient non-fatal errors -----
<websupt@lab.dce.harvard.edu>

----- Transcript of session follows -----
451 4.4.1 <websupt@lab.dce.harvard.edu>... Deferred: Connection reset
Warning: message still undelivered after 4 hours
Will keep trying until message is 5 days old

Reporting-MTA: dns; smtp3.fas.harvard.edu
Arrival-Date: Thu, 25 Oct 2012 15:35:05 -0400 (EDT)
Action: delayed
Status: 4.4.2
Last-Attempt-Date: Thu, 25 Oct 2012 19:54:26 -0400 (EDT)
Will-Retry-Until: Tue, 30 Oct 2012 15:35:05 -0400 (EDT)
….. a copy of the original message followed….

Email Delivery Problems (Part 1a)

What can happen when the destination mail system is not available?

Return-Path: <MAILER-DAEMON@fas.harvard.edu>
Received: from localhost by smtp3.fas.harvard.edu
Date: Thu, 25 Oct 2012 19:54:27 -0400 (EDT)
From: Mail Delivery Subsystem <MAILER-DAEMON@fas.harvard.edu>
To: <evenchk@fas.harvard.edu>
MIME-Version: 1.0
Content-Type: multipart/report; report-type=delivery-status;

Subject: Warning: could not send message for past 4 hours
Auto-Submitted: auto-generated (warning-timeout)

Email MTA Forwarding

Date: Sat, 1 Dec 2012 17:10:26 -0500 (EST)
From: csci-40@mail.dce.harvard.edu
To: len@alum.mit.edu
Subject: Message to test MTA forwarding

This is a test of forwarding by MTAs.

--

Email MTA Forwarding
(With header option turned on.)

Date: Sat, 1 Dec 2012 17:10:26 -0500 (EST)
From: csci-40@mail.dce.harvard.edu
To: len@alum.mit.edu
Subject: Message to test MTA forwarding
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII

This is a test of forwarding by MTAs.

--

Email MTA Forwarding

Forward 4

FORWARD 4
Received: from ALUM.MIT.EDU [18.7.21.81]
by smtp3.fas.harvard.edu with ESMTP id... 1 Dec 2012 17:10:29
Return-Path: <csci-40@mail.dce.harvard.edu>

Forward 3

FORWARD 3
Received: from smtp2.fas.harvard.edu [140.247.34.52])
by alum.mit.edu with ESMTP for <len@alum.mit.edu>;
1 Dec 2012 17:10:28 -0500 (EST)
From: csci-40@mail.dce.harvard.edu

Forward 2

FORWARD 2
Received: from mail.dce.harvard.edu {140.247.197.235]) by
smtp2.fas.harvard.edu with ESMTP 1 Dec 2012 17:10:28 -0500 (EST)

Forward 1

FORWARD 1
Received: from csci-40 by mail.dce.harvard.edu with local-esmtp for
len@alum.mit.edu
id 16AIL4-0000PB-00; Sat, 01 Dec 2012 17:10:26 -0500

Email message

Date: Sat, 1 Dec 2012 17:10:26 -0500 (EST)
To: len@alum.mit.edu
Subject: Message to test MTA forwarding
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII
This is a test of forwarding by MTAs.

Email MTA Forwarding

FORWARD 4 - not shown (see next page)
FORWARD 3 - not shown (see next page)

FORWARD 2
Received: from mailaa.dce.harvard.edu {140.247.197.235]) by
smtp2.fas.harvard.edu with ESMTP 1 Dec 2012 17:10:28 -0500 (EST)

FORWARD 1
Received: from csci-40 by mailaa.dce.harvard.edu with local-esmtp for
len@alum.mit.edu
id 16AIL4-0000PB-00; Sat, 01 Dec 2012 17:10:26 -0500

actual email message….

Email MTA Forwarding

FORWARD 4
Received: from ALUM.MIT.EDU [18.7.21.81]
by smtp3.fas.harvard.edu with ESMTP id... 1 Dec 2001 17:10:29
Return-Path: <csci-40@mail.dce.harvard.edu>

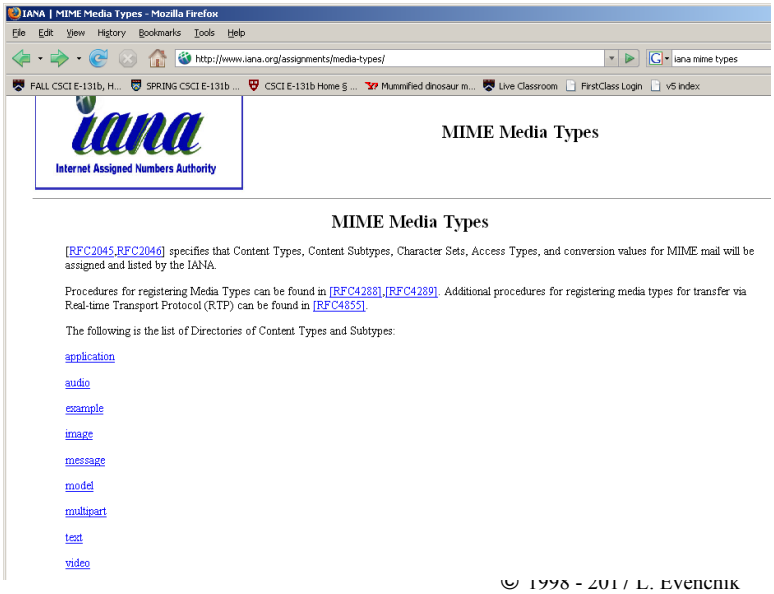
FORWARD 3
Received: from smtp2.fas.harvard.edu [140.247.34.52])
by alum.mit.edu with ESMTP for <len@alum.mit.edu>;
1 Dec 2012 17:10:28 -0500 (EST)
From: csci-40@mail.dce.harvard.edu

FORWARD 2 - not shown
FORWARD 1 - not shown

MIME
(Multipurpose Internet Mail Extensions)

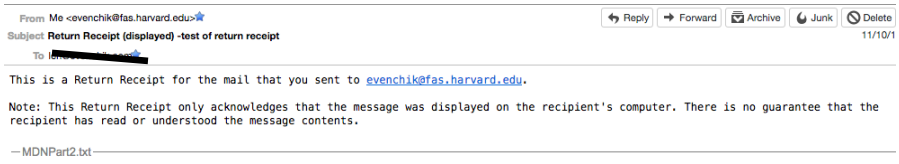
- Original email RFCs talked about ASCII messages.
- MIME defines encoding rules to allow for non-ASCII messages. Multiple RFCs (2045 - 2049, and more)
- Defines additional message headers within email message.
- Content-Tansfer-Encoding defines how the body is wrapped for transmission. Schemes include: 7-bit ASCII, 8-bit characters, base64 encoding, quoted-printable, binary
- Content-Type describes the nature of the message. Types include: text, image, audio, video application, multipart
- Sub-types are present for each Content-Type
- Defined first for email, has been applied to HTTP, RTP and SIP. MIME listing available at IANA

WWW.IANA.ORG



Is Email Reliable?

- Email uses TCP, but what does that really mean for whether or not an email message has been delivered to the intended recipient?
- What does an email delivery notification mean, and how is it done?
- See RFC 8098, Feb 2017, for information on Message Disposition Notification



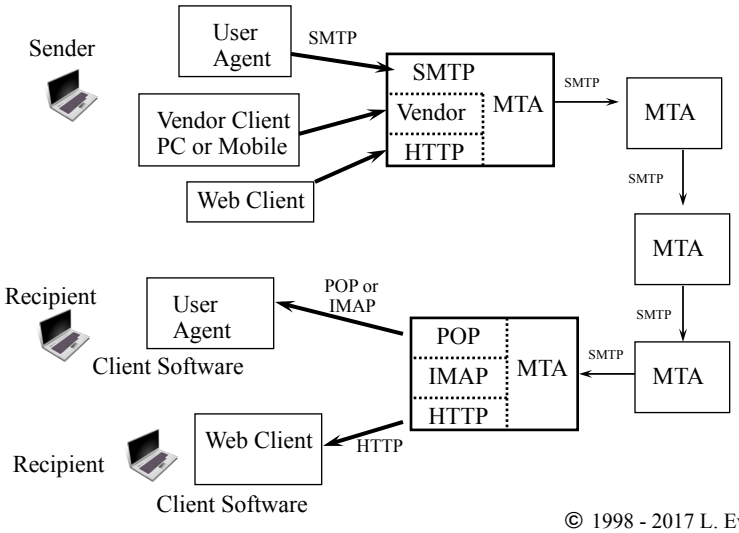
Headers Fields Used for Message Disposition

- In non-proprietary email systems, SMTP header fields are used to track email delivery.

```
Return-Path: <evenchik@fas.harvard.edu>
Received: MULTIPLE SYSTEMS LISTED AS THE EMAIL
                MOVED TO DESTINATION
To: "cs40@evenchik.com" <cs40@evenchik.com>
From: Len Evenchik <evenchik@fas.harvard.edu>
Subject: Test of Return Receipt feature in SMTP
Message-ID: <jdfjoioiejk@fas.harvard.edu>
Disposition-Notification-To: Len Evenchik <evenchik@fas.harvard.edu>
Date: Fri, 10 Nov 2017 20:45:49 -0500
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
MIME-Version: 1.0
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 7bit
Content-Language: en-US

This is my simple test message to see how delivery notification
is done.
```

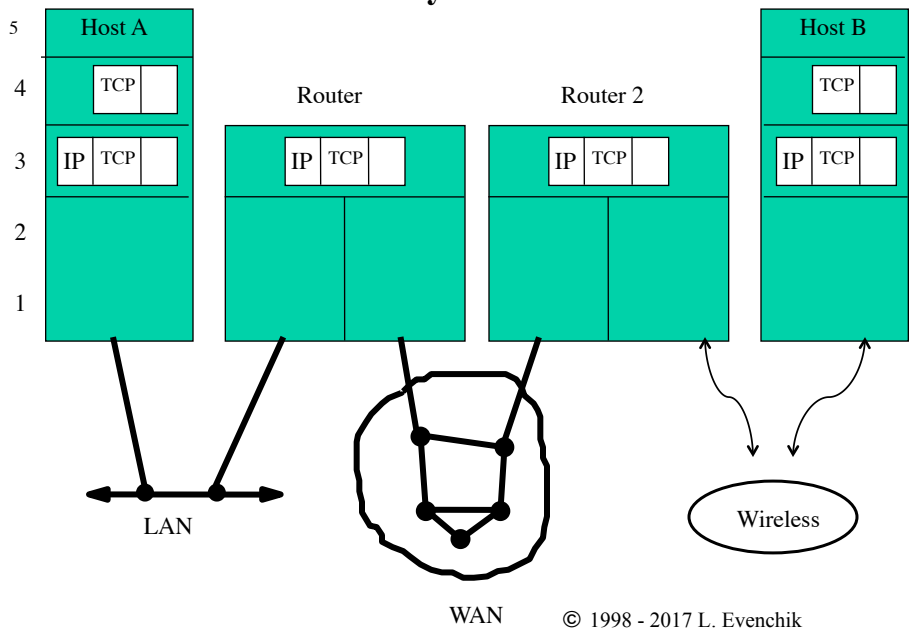
Header field used for delivery notification. Must be handled by client

Mail System Architecture
Protocols for Client to MTA



Network and System Security

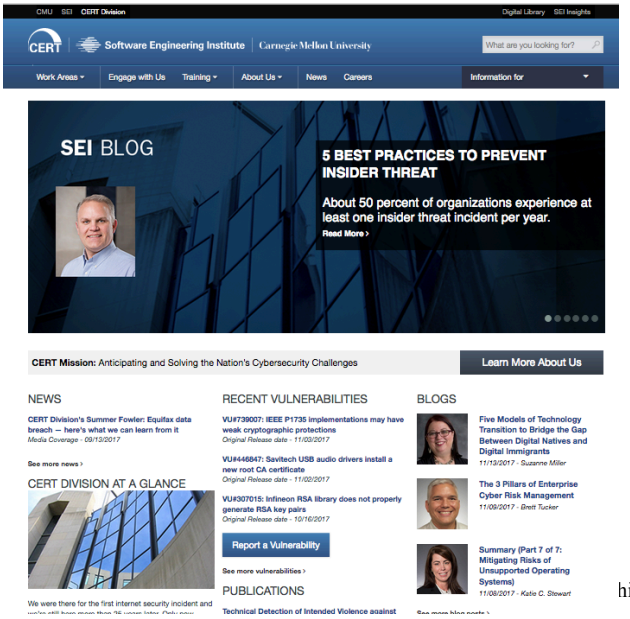
Where Should Security be Used in this Network?



Security Resources
No Single Resource is Enough

- CERT Coordination Center
  - 412-268-7090 (always have the current tel #)
  - www.cert.org
  - cert@cert.org
- US-CERT Coordination Center
  - 1-888-282-0870 (always have the current tel #)
  - http://www.us-cert.gov/
  - soc@us-cert.gov
- Your corporate IT group and legal department.
- IETF working groups, other well known security organizations
- Your ISP
- Your firewall, router and other equipment vendors

WWW.CERT.ORG



hik

The Basics

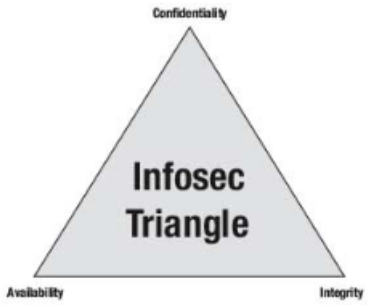
Network and System Security - Some of the Obvious Threats and Problems

- Breaking into computers, networks, etc.
- Eavesdropping, monitoring of networks (the older term for this in POTS was Wiretapping.)
- Stealing money, your identity and data
- Stealing your password
- Denial of service attacks
- Replaying prior conversations as an original
- Masquerading as someone else (Identity theft)
- Inserting a Trojan horse, worm or virus on your PC, phone, car, thermostat (IoT)
- Changing the message that was sent
- Etc., etc., etc.

Infosec Triangle or CIA Triad

This is a common business oriented approach to understanding security; we will complement this with a more technical framework.

- Confidentiality
- Availability
- Integrity



Security – the Most Basic Building Blocks

- Physical security for systems and networks
- Password and 2-factor security for systems and networks
- Shared secret encryption system
- Public key encryption system
- Hashes and Digital signatures
- Firewalls
- VPN (Virtual Private Networks)
- Encryption and authentication of web pages (TLS)
- Proper procedures and training!

Security is a system issue which requires hardware, software, procedures, and people who understand and care about it.

But First, Everyone Needs to Understand that Security is NOT the same as a Password

- Classic and most common security is based solely on password protection, but most passwords are too easy to guess. Users use easy passwords so that they can remember them. Multi-factor security is MUCH better, but this approach still has problems.
- Consumer and some business PCs and other equipment are still shipped with "friendly" password settings and provide no security protection. Users do not change them!
- IoT devices can have default passwords that cannot be changed.
- The use of "security questions" to allow a user to reset their password has significant security risks.
- It is very important to provide physical security to prevent common methods of attack. Stealing a USB stick is easy.

The Most Common Passwords

Most Common & Worst Passwords of 2014

Rank	Password	Change from 2013
1	123456	Unchanged
2	password	Unchanged
3	12345	Up 17
4	12345678	Down 1
5	qwerty	Down 1
6	123456789	Unchanged
7	1234	Up 9
8	baseball	New
9	dragon	New
10	football	New
11	1234567	Down 4
12	monkey	Up 5
13	letmein	Up 1
14	abc123	Down 9
15	111111	Down 8
16	mustang	New
17	access	New
18	shadow	Unchanged

These examples are a few years old but things have not improved much.
Source of tables – trade press

COMMON PASSWORDS

The Worst Passwords of 2012, including their current ranking and any changes from the 2011 list:

1. password (Unchanged)
2. 123456 (Unchanged)
3. 12345678 (Unchanged)
4. abc123 (Up 1)
5. qwerty (Down 1)
6. monkey (Unchanged)
7. letmein (Up 1)

1. **123456**
   I can't be bothered to take even the most basic step to protect my personal information. Seriously, just go ahead and take it.
2. **password**
   I failed to understand the question.
3. **12345678**
   I tried "123456," but the computer said I had to use at least eight characters.
4. **qwerty**
   Aren't I clever? My password is written right there on the keyboard.
5. **abc123**
   I'm a fan of the Jackson Five.

In Summary, Security Requires:

- Hardware
- Software
- Written Procedures and Processes
- People educated on what security means and how to properly do it.

Security is a system issue which requires all of the above, but without a doubt, people who understand and care about the issues are the most important element.

A More Technical Approach to Security

Structured way to Think about Security: Five Important Elements

- Privacy and confidentiality
- Authentication
- Authorization
- Integrity
- Nonrepudiation

Eavesdropping and Wiretapping

- Eavesdropping: it is easy to physically connect to a wired network, and very easy of course to listen to a wireless network, and it can even be done on a fiber based network. Monitoring of network traffic can also be done remotely.
- It is very difficult to prevent this, or detect that it is going on.
- This eavesdropping is a straightforward way to capture passwords and other sensitive information. Hence the need for encryption.
- Encrypting a single link (wired or wireless) prevents the compromise of information from wiretapping and snooping on that specific link, but not other links. Hence this is just the beginning of a solution. End to end encryption is needed.
- One way you hear about to reduce the risk in a network is to segment the traffic using ethernet switches. Why does this help, but what is the weakness? What about routers?

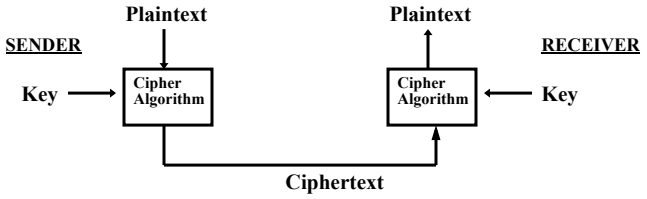
Cryptography

Approaches to Cryptography

- Symmetric cryptography
  - Shared secret key system
  - Same key used to encrypt and decrypt messages
  - Key length determines the "strength" of encryption
  - Key management is difficult
  - Examples are 3DES, IDEA, RC4 and AES
- Asymmetric cryptography (called Public Key)
  - Key pair - one public, one private
  - Data encrypted by one key must be decrypted by the other key
  - Examples are Diffie-Hellman (1976) and RSA (1978)

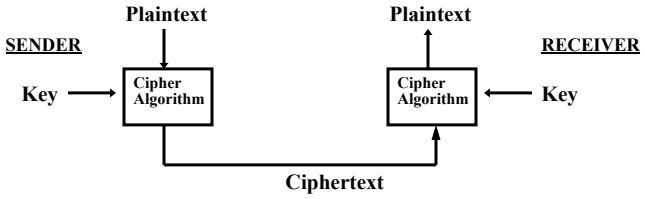
Data Encryption Standard (DES) Historical Reference ONLY

- DES was one of the original shared secret key encryption schemes (FIPS standard), now about 40 years old. **No longer secure, historical reference only.**
- Encryption in 64 bit blocks with chaining or feedback
- Key length determines the "strength" of encryption.
- DES used a 56 bit key which is **now too weak. We will see multiple examples of this trend: time and research make secure systems insecure.**



Advanced Encryption Standard (AES)

- Advanced Encryption Standard (AES) is a shared secret key encryption scheme.
- Provides Confidentiality for your data.
- Developed by NIST using a public evaluation process (15 candidates.) AES published in Nov. 2002.
- AES is a symmetric block cipher encryption algorithm
- AES supports key lengths of 128, 192 and 256 bits. Why different sizes?



Public Key Encryption

- Key distribution and management has always been the weak link in shared key systems
- Public key systems solve this problem by having two keys, a "private key" and a "public key"
- Users publish their "public key" and other people can send them encrypted messages by using this specific "public key"
- The "magic" that makes this possible is the use of complex algorithms that make it "very hard" to guess a private key even if you know the public key and the underlying algorithm.
- This approach is used extensively today for web traffic, email, other applications.

Public Key Algorithms

- Key distribution and management has always been the weak link in shared key systems
- Public key systems solve this problem by being able to publish a "public key"
- Algorithm must provide the following functionality:
  - D(E(P)) = P
  - It is very difficult to deduce D from E
  - E cannot be broken by a chosen plaintext attack
- Appropriate algorithms are based on hard problems such as taking the log of a number or factoring large numbers.

ssh-keygen

```
cscie40@courses (~): ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the
                key (/home/web/c/s/cscie40/.ssh/id_rsa): test4
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test4.
Your public key has been saved in test4.pub.
The key fingerprint is:
25:dc:21:44:0b:bd:25:76:67:83:1e:86:0c:fa:ce:20 cscie131b@barkley
cscie40@dcepea (~):
```

Private Key

cmd **(~): cat test4**
-----BEGIN RSA PRIVATE KEY-----
```
MIICXQIBAAKBgQC/aSKmm6VdcqL6IQzK81998Ac8Coes/V2l4KGZItcSYboSE1e7
s3RVssdY9Xqol1cVEXhhQ/SnzcQhKti4CrC6dxyOwpVDdDSo7ZW8LWRg2Gw1jFoU
KDUeLsEqbzmEBdteuvixbUITaMGqjtKnjdFo8fi3Y7MW5sS2ZvdpweSkWwIBIwKB
gQC58RpYtHTBLYhgsmQy3cp6VuJ01wd02N6hI1snC+beqBPXC3nMR+1akGnhY353
43kqaL4VV/T6x+MY58s2cMjt9/5Llecfpt/uW53U56TZwniPgAQEiQe5nPowdzNZ
vs0QoVOpHUWPyvjTPAZVcEm68BaYI6FlESYdpOvwWkDlawJBAOujhwyWnX3u47po
VGJnfHNhVsO8wzsA4U26heTVE8nxgRwI0vsn4REJRNDByXqJ6pyLJHEVq6Y9lCnJ
QnwJYmkCQQDP80FugasGuZsgKQygS6ngndKXqD0y95RlFCda+t8krVuPdBnE6S1h
Iappr/6YKMVtVlCv0aFUhYtAlaDKlLAjAkEA13DwgIm0kGVifotF1k/8wUMNf8KG
nFiTekQipVUZaC1C182Nsm2akzusoZs73b/scd5NNDER9xO6qdyUjql+iwJBAIin
Kv95yCj9oHQ4O39HqiXkDgvjle5K7Hzv/JrfX2/fopF4LjDxAJBJUrp698MTemxr
6+FAnTeK9RvQCpPq2iUCQQDWkdkUUcmTuF15zKZ/M+5mUwfdpvErt7FICBeQl4X8
iak2TSIoZ1uBUq4YUdf38oCJX+QJVESdi8PovTVdUii3
```
-----END RSA PRIVATE KEY-----
cmd **(~):**

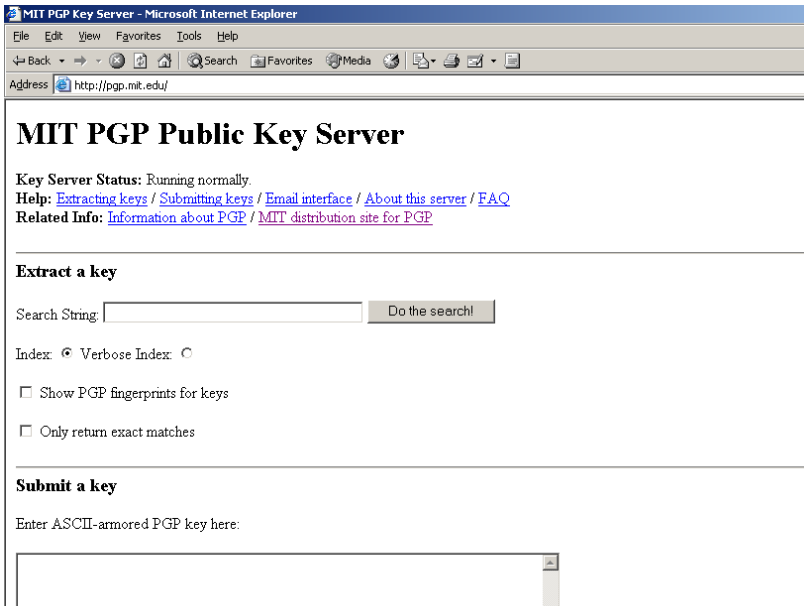
Public Key

cmd **(~): cat test4.pub**

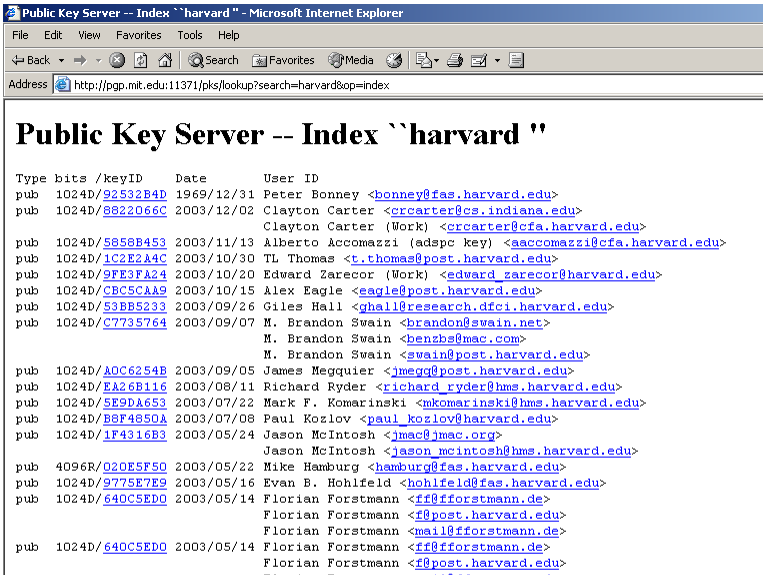
```
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIE
Av2kippulXXKi+iEMyvNfffAHPAqHrP1dpeCh
mSLXEmG6EhNXu7N0VbLHWPV6qJdXFRF4
YUP0p83EISrYuAqwunccjsKVQ3Q0qO2VvC1
kYNhsNYxaFCg1Hi7BKm85hAXbXrr4sW1CE2
jBqo7Sp43RaPH4t2OzFubEtmb3acHkpFs=
```

cmd **(~):**

http://pgp.mit.edu/



Search for pgp keys for "Harvard"



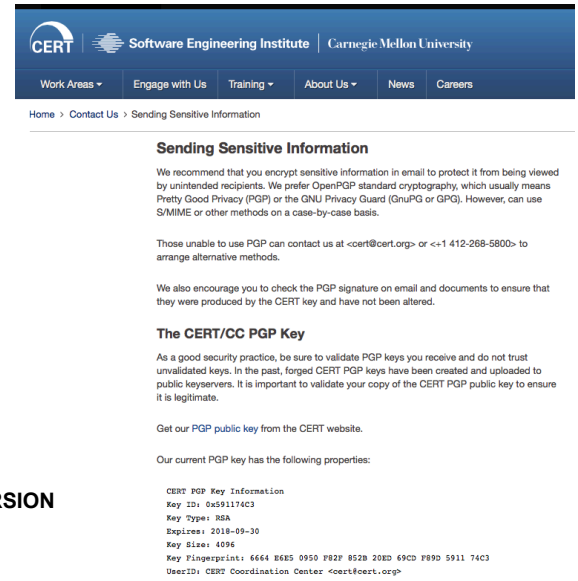
Portion of a Public key

Public Key Server -- Get "0x924a01145f19

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: SKS 1.1.6
Comment: Hostname: pgp.mit.edu

mQGiBEsNlOsRBAD1JnUpIxnbhrHXBmA0pBW01pZWqPyerHmmYpZp4O8tWwm/1HPWvEN09hdX
Hz78yAYvErlTaWqBa/pMI+bjHCCkUTDeQ1iQodng4kkjeOmfwW/w2agM0fIcG7eozyTxJ8qQ
PoNaE1ZYa35qyCnD3sxDWHVNheViyw2Hglo+NeykzwCg4FszvPCcAU9GGKyDG6rR4xwg/6sE
ANmLDcu8khid09ZULhTasNP3UaBI1pPf0IrOlgu4ER1t2u4eS3eFyszWM2WeZpGZrwmH26Y5
ljql1+UXb41rw1uDajEctGCQPTYPWmUC4G4JkkkIBF6gzPgLfUjTYqUUeI5+elvtL3LP+jdP
n2yQGDyKVtd0k3mAp+Rx7T1v9m2WBACiTZbuN3KzEYqjBg+n6583a6Xb9Pt7eOkjCHgqHEkg
Gx3gMcDnguv89BI69t/jdmVpuM24VJ/bTCyXBz7AitDzEvNqAFK2iViUkJJ9afGxajm74SUH
SrlsZYqOKMfMhvhQOc6UbqVJuqHM7gksPM0xe+IrTKSpMZ0/yKal3Xjb27QxTGVuIEV2ZW5j
aGlrIChIYXJ2YXJkKSA8ZXZlbmNoaWtAZmFzLmhhcnZhcmQuZWR1PohgBBMRAgAgBQJLDZTr
AhsDBgsJCAcDAgQVAggDBBYCAwECHgECF4AACgkQkkoBFF8Z80jgcACgtn8hztLjB5Xn1WMJ
+wueaA3wbbQAoNgRQv0SP4ngNKe0R0yW3gALSsDGuQINBEsNlPkQCACSjhUOTEIQJvobRl/c
xYACClvtsvvolvP67Z/exKXhpPNhN6bbQlEFWOOpcN4vfFX1IYW58xs9ppUsYbRptd5QsF9r
4Qh7wC8MhZ/PM/bV1e/P3N5zSGzElWwAPOubMOE18PdpApuU88Pw/IKGrDJx6mD0EFy8BZzq
oxomKeA+Q22T9Ey8H83nD7yn1uHKyl0vYGyshplXosexiU62N8U5P1ECAACiFwnYsUUPSlpW
```


http://pgp.mit.edu/ - submit a key

https://www.cert.org/contact/sensitive-information.cfm



CHECK FOR CURRENT VERSION

What Do You Want to Accomplish?

- How can you use Public/Private Key encryption to send a secure message?
- How can you use Public/Private Key encryption to authenticate one of the people in the conversation?

Public Key Encryption (1)

Assume that you want to send a private message.

But note that public key encryption algorithms are much slower than symmetric key algorithms.

Therefore a combination of the two cryptographic approaches are used by most systems:

- Sender does….
- Recipient does….

Let's fill in the details…..

Public Key Encryption (2)

Assume that you want to send a private message.

But note that public key encryption algorithms are much slower than symmetric key algorithms.

Therefore a combination of the two cryptographic approaches are used by most systems:

- Sender creates a session key (secret AES key)
- Sender encrypts message with that session key
- Sender then encrypts session key with recipient's public key
- Sender sends encrypted key and encrypted message to recipient.
- Recipient decrypts session key with private key
- Recipient then decrypts message using session key

Public Key Encryption

What functionality is NOT provided by the procedures we just described (on the previous slide)?

© 1998 - 2017 L. Evenchik

Public Key Encryption (3)

Assume now that you want to send a private message **and** authenticate the sender's identity.

Add a few steps to the procedure:

- Sender creates a session key
- Sender encrypts message with session key
- Sender encrypts session key with recipient's public key
- Sender encrypts the key again with the sender's private key. (This means the key is encrypted twice.)
- Sender sends encrypted key and encrypted message

- *What does the Recipient do?*

Hashing and Message Digests

Hashing Functions and Message Digests

- Hash functions take an arbitrarily long piece of plaintext and compute from it a fixed length string.
- Hash functions are based on the fact that there are mathematical transformations that are easy to do but very, very hard to undo.
  - In mathematical terms y=f(x)
  - Given f and x, it is very easy to compute y
  - Given f and y, it is very hard to compute x
- Common message digests are 128 bits or longer
- **Hash functions can show that a message has not changed, but they do not provide confidentiality.**

Hashing Functions and Message Digests (2)

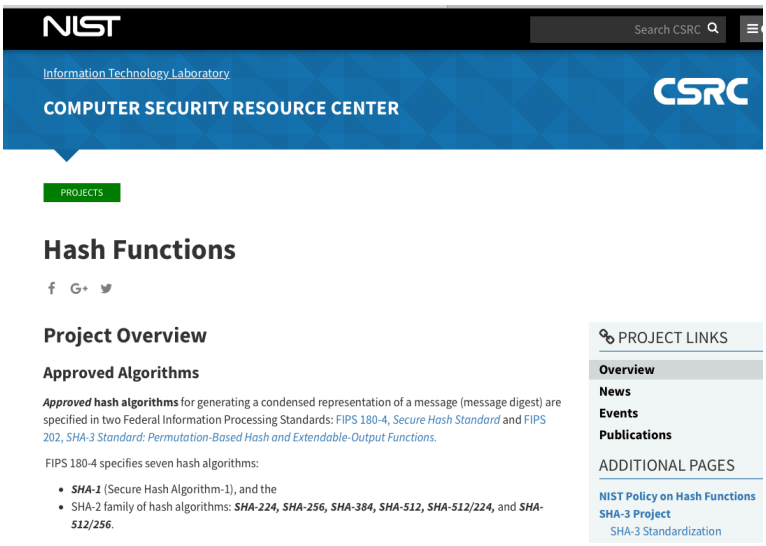
- MD5 was the 5th hash function designed by Ron Rivest (1992). Security issues are well known with MD5 and it is no longer considered secure. However it is still used in some systems. See the CERT notes about this.
- Although it is still widely used, SHA-1 has been deprecated by NIST as of Jan. 2014. See RFC 6194 and the following:

  http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

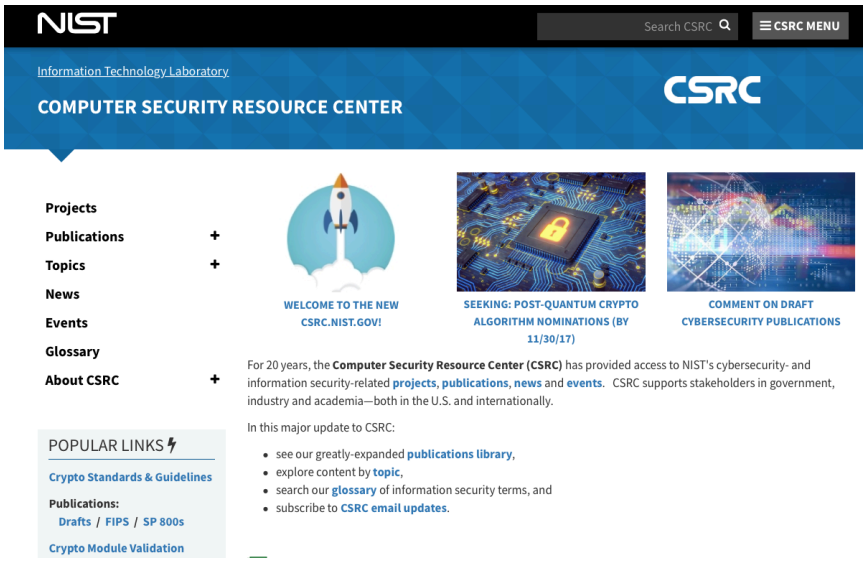
  http://googleonlinesecurity.blogspot.com/2014/09/gradually-sunsetting-sha-1.html
- SHA-2 (SHA-256) and SHA-3 are the current hash functions that have been standardized by NIST. See the NIST website:

  https://csrc.nist.gov/projects/hash-functions
- Remember, Hash functions do not provide confidentiality.

https://csrc.nist.gov/projects/hash-functions



https://csrc.nist.gov



Print of "testfile1"

cmd (~): **cat testfile1**
this is a test file to be used in the networks and
protocols class...
abcdefghijklmnopqrstuvwxyz1234567890
Hello World
This is line five (5) of this file.
cmd (~):

SHA-1 of a file called "testfile1"

cmd (~): **cat testfile1**
this is a test file to be used in the networks and
protocols class...
abcdefghijklmnopqrstuvwxyz1234567890
Hello World
This is line five (5) of this file.
cmd (~):

cmd (~): **sha1 testfile1**
sha1 (testfile1) = 88a5b867c3d110207786e66523cd1e4a484da697
cmd (~):

NOTE THAT WE ARE USING SHA-1 ONLY AS AN SIMPLE EXAMPLE. IT IS NO LONGER SECURE!

Comparison of "testfile1" and "testfile2"
Note the small difference on the line with "Hello World"

cmd (~): **cat testfile1**
this is a test file to be used in the networks and protocols class...
abcdefghijklmnopqrstuvwxyz1234567890
Hello World
This is line five (5) of this file.
cmd (~):

cmd (~): **cat testfile2**
this is a test file to be used in the networks and protocols class...
abcdefghijklmnopqrstuvwxyz1234567890
Hello World !
This is line five (5) of this file.
cmd (~):

SHA-1 Comparison for files "testfile1" and "testfile2"

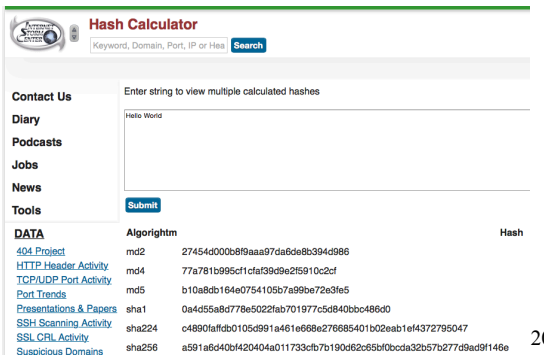
cmd (~): **SHA1 testfile1**
SHA-1 testfile1) = 88a5b867c3d110207786e66523cd1e4a484da697
cmd (~):

cmd (~): **SHA-1 testfile2**
SHA-1 (testfile2) = 874945e767b56391e8234780ce1d5150c11d9060
cmd (~):

NOTE THAT WE ARE USING SHA-1 ONLY AS AN SIMPLE EXAMPLE. IT IS NO LONGER SECURE!

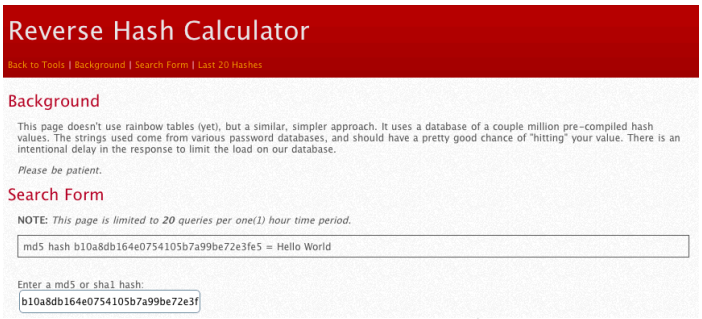
Online Hashing Calculator

- There are many online hash calculators that demonstrate how hash functions work.
- We will take a look at https://isc.sans.edu/tools/md5.html but please note that we cannot vouch for the cryptographic correctness of this implementation
- There are also reverse hash calculators on the net.



Online Reverse Hash Calculator

- A reverse hash calculator does just what is sounds like it does.
- We'll take a look at https://isc.sans.edu/tools/reversehash.html
- It is critically important that you understand that every security tool and system has limitations and you need to understand the details in order to use them properly.

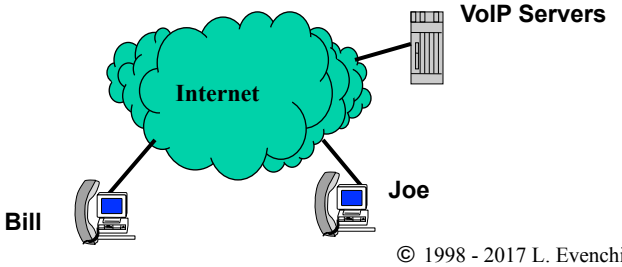


User Authentication in a VoIP System via Hash Functions

© 1998 - 2017 L. Evenchik

VoIP/SIP User Authentication

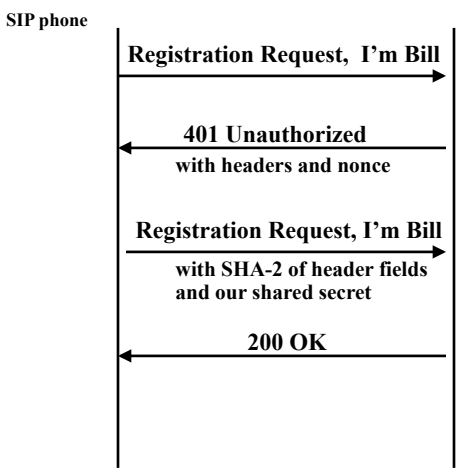
- The question is: How does a SIP VoIP server or system know that you are who you say you are? If a user is not authenticated, it would be easy for anyone to say that they are bill@harvard.edu and get that user's telephone calls.
- To answer this, first consider how this is done for your home telephone service (POTS), your cell phone, and the process of logging into your company mail server.



User Authentication

- The question is: How does a remote server know that you are who you say you are? If a user is not authenticated, it would be easy for anyone to say that they are bill@harvard.edu and get that user's telephone calls, or access to another type of service.
- The name for this is user authentication and and it requires that the VoIP phone and the VoIP server know a shared secret but the secret should never be sent as clear text over the net. The technique is called HTTP Digest Authentication.
- The SHA-2 (or other hash) of the combination of the user name, shared secret, realm, and nonce (plus some other fields) is computed, sent, and then compared to the expected value to authenticate the user. The nonce provides protection against later replay.
- Let's study at an example using a VoIP SIP phone.

SIP VoIP User Registration



This is an abridged trace of the packet flows

Registration to VoIP Proxy Server (Step 1)

```
Session Initiation Protocol
        Request-Line: REGISTER sip:siplearn.com:5060 SIP/2.0
        Method: REGISTER
  Message Header
        Via: SIP/2.0/UDP
140.247.250.181;branch=z9hG4bKf7f8d7477263E836
        Transport: UDP
        Sent-by Address: 140.247.250.181
        Branch: z9hG4bKf7f8d7477263E836
    From: "Bill at ext 6003" <sip:bill@siplearn.com>;tag=8F21…
    To: <sip:bill@siplearn.com>
    CSeq: 1 REGISTER
        Call-ID: bc8e2e39-68f1d8c0-b947fe7b@140.242.250.181
        Contact: <sip:bill@140.247.250.181>……
        Contact Binding: <sip:bill@140.247.250.181>;
        methods="INVITE…..
        etc…
        etc…
```

I'M BILL

abridged trace

401 Unauthorized (Step 2)

Session Initiation Protocol
Status-Line: SIP/2.0 401 Unauthorized
Message Header
Via: SIP/2.0/UDP 140.247.250.181; branch=xxx, received=140.247.250.181
Transport: UDP
Sent-by Address: 140.247.250.181
From: "Bill 6003" <sip:bill@siplearn.com>;tag=8F215C5A-D94BE88D
To: <sip:bill@siplearn.com>;tag=as47f93dba
Call-ID: bc8e2e39-68f1d8c0-b947fe7b@140.247.250.181
CSeq: 1 REGISTER
User-Agent: Asterisk PBX
Allow: register…
WWW-Authenticate:
Authentication Scheme: Digest
Algorithm: SHA-2
Realm: "siplearn.com"
Nonce Value: "0810d7034435aed35c"
Content-Length: 0

NONCE

abridged trace

Register with Response to Challenge (Step 3)

Session Initiation Protocol
Request-Line: REGISTER sip:siplearn.com:5060 SIP/2.0
Method: REGISTER
Message Header
Via: SIP/2.0/UDP 140.247.250.181;branch=z9hG4bK5149a9846EA080EF
From: "Bill 6003" <sip:bill@siplearn.com>;tag=8F215C5A-D94BE88D
To: <sip:bill@siplearn.com>
CSeq: 2 REGISTER
Sequence Number: 2
Call-ID, Contact, etc, etc, etc
Authorization:
Authentication Scheme: Digest
Username: "bill"
Realm: "siplearn.com"
Nonce Value: " 0810d7034435aed35c "
Authentication URI: "sip:siplearn.com:5060"
Digest Authentication Response:
"9d68372b3929befa2a2eeaa0dcbf03df"
Algorithm: SHA-2

I'M BILL

NONCE

SHA-2 OF ID, NONCE AND SECRET

abridged trace

One Minute Wrap-Up

- Please do this Wrap-Up at the end of each lecture.
- Please fill out the form on the website.
- The form is anonymous (but you can include your name if you want.)
- Please answer three questions:
  - What is your grand "Aha" for today's class?
  - What concept did you find most confusing in today's class?
  - What questions should I address next time
- **Thank you!**