Week 9
SSH

26 November 2018

CS 35L Lab 4

Jeremy Rotman

Announcements

- Assignment #8 is due Saturday by 11:55pm
- Assignment #10 Presentations
  - **Email me to tell me what story you are choosing**
  - Here is the link to see what stories people have signed up for already
- For assignments 8 and 10
  - There will be separate submission links for each lab
  - Make sure you submit yours to the correct submission link
    - We are Lab 4
- Quick reminder for the upcoming assignments (9 and 10)
  - No late submissions

Outline

- SSH
- Assignment 8

Questions?

Communicating Over the Internet

What guarantees do we want?

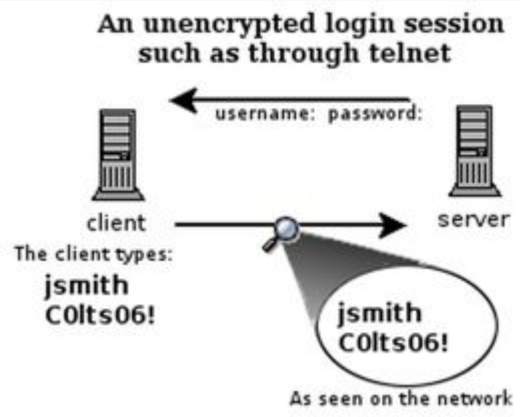
Communicating Over the Internet

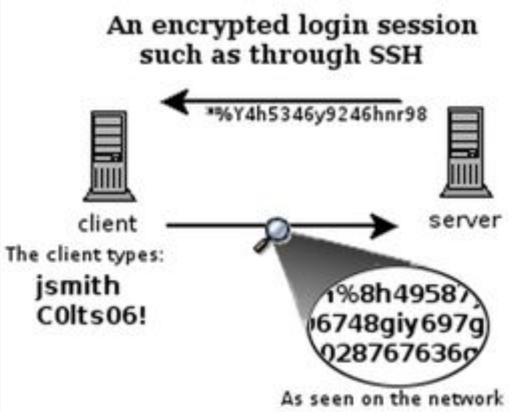
What guarantees do we want?

- Confidentiality
  - Message secrecy
- Data integrity
  - Message consistency
- Authentication
  - Identity confirmation
- Authorization
  - Resource access rights specification

SSH

- Secure Shell
- Used to remotely access shell
- Successor of telnet
- Encrypted and better authenticated session

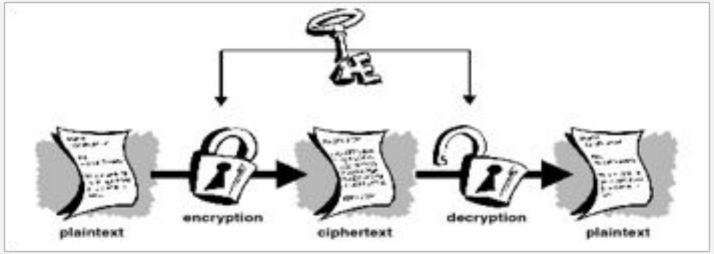


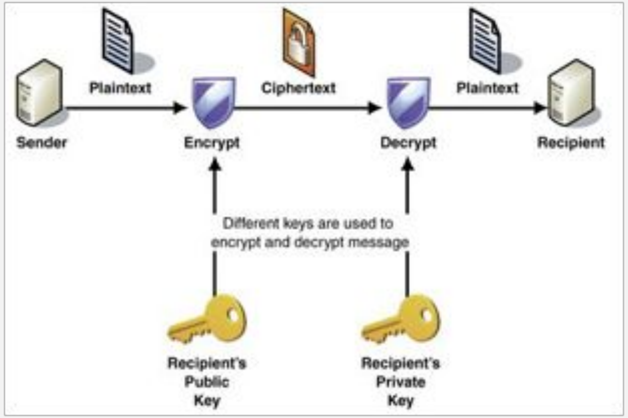


Encryption Types

- Symmetric Key Encryption
  - shared/secret key
  - Key used to encrypt is the same as key used to decrypt
- Asymmetric Key Encryption
  - public/private
  - 2 different, but related keys, public and private key
    - Private key cannot be derived from the public key
  - Data encrypted with public key can only be decrypted with private key
  - Public key can be seen by anyone
  - **Never** publish private key

Symmetric vs. Asymmetric





Symmetric

Asymmetric

SSH Protocol

Client SSH's to remote server:

- If this is the first time:
  - ssh does not know the host
  - Shows hostname, IP address, and fingerprint of the server's public key so you can confirm the correct host
  - If the client accepts, connection, it saves the public key to its known hosts
    - ~/.ssh/known_hosts

SSH Protocol

- The next time, if the host's public key does not match what was already saved
  - Client encrypts a message with the public key
  - The host decrypts this with its private key to prove that it is the real host
- Once the host is verified
  - Host and client agree on a symmetric encryption key
    - For the session
  - All messages between the host and client are encrypted and decrypted with the session key

SSH Protocol

- User Authentication
  - Password authentication
    - User is prompted for password
  - Key-based authentication
    - Generate key-pair
    - Copy the public key to the server
    - Server authenticates client if it can demonstrate it has the private key
    - Private key can have a passphrase attached
      - But this forces you to enter passphrase every time the private key is used

ssh-agent

- ➔ Program that works with OpenSSH that provides a secure way of storing private key
- ➔ `ssh-add` prompts user for passphrase once and adds to the list maintained by `ssh-agent`
- ➔ Once passphrase is added to `ssh-agent`, user will not be prompted for it again when using SSH
- ➔ OpenSSH will talk to ssh-agent and retrieve private key automatically

X Window System

- Windowing system for GUIs on UNIX
- X is network based
  - A program can run on one computer but display on another
  - X session forwarding

Lab 8

- You will have to team up with one other person
  - May need one team of 3 since we have an odd number of people
- You will need to set up your BeagleBone as described on Piazza
- Use OpenSSH to enable secure login to each other's hosts
- Generate key-pairs
- Use ssh-agent to make your logins convenient
  - Passphrase entered only once
- Use port forwarding to run a command from a remote host on your host

Lab Environment Setup

- If you’re on Ubuntu
  - Make sure you have openssh-server and openssh-client installed
    - `dpkg --get-selections | grep openssh`
    - Should give you
      - openssh-client   install
      - openssh-server install
    - If not, install them
      - sudo apt-get install openssh-client
      - sudo apt-get install openssh-server

Server Steps

- Generate public and private keys
  - `ssh-keygen` (by default saved to ~/.ssh/is_rsa and id_rsa.pub)
- Create an account for the client on the server
  - `sudo useradd -d /home/<homedir_name> -m <username>`
  - `sudo passwd <username>`
- Create .ssh directory for new user
  - `cd /home/<homedir_name>`
  - `sudo mkdir .ssh`

Server Steps

- Change ownership and permission on .ssh directory
  - `sudo chown -R username .ssh`
  - `sudo chmod 700 .ssh`
- Optionally, disable password-based authentication
  - `emacs /etc/ssh/sshd_config`
  - Change the PasswordAuthentication option to no

Client Steps

- Generate public and private keys
  - `ssh-keygen`
- Copy your public key to the server for key-based authentication
  - `ssh-copy-id -i <username>@<server_ip_addr>`
- Add private key to authentication agent
  - `ssh_add`
- SSH to server
  - `ssh -X <username>@<server_ip_addr>`
- Run command on remote host
  - E.g. `firefox`

Checking IP address

- ifconfig
  - Configure or display the current network interface configuration information
- hostname -I
  - Give IP address of your machine directly
- ping <ip_addr> (packet internet groper)
  - Test reachability of a host on IP network
  - Measure round-trip time for messages sent from source to a destination

Questions?