Systems Integration

Week 8 – Remote Access

Eoin Rogers (eoin.rogers@tudublin.ie)

Two technologies this week

- FTP – File transfer protocol
- SSH – Secure shell

File Transfer Protocol

- TCP-based
- Not at all secure!
  - Supports authentication, but no encryption (!)
  - But, still useful for downloading!
- Client-server model
- Text-based (partially!), in many ways quite similar to HTTP

Types of FTP transfer

- Anonymous – access without user accounts is possible
- Authenticated – users have to log in
  - Arguably, this is a really bad idea. Any ideas why?

FTP ports

- Uses TCP
- Uses at least two TCP connections:
  - One for control
  - One for data
- HTTP-like, text-based commands are sent over the control channel
- Data are transferred over the data channel

FTP ports and modes

- FTP uses port 21 for the command channel
- The port number used for the data channel is variable, and depends on the mode that FTP operates in:
  - Active mode – Port 20 is used for data transfer
  - Passive mode – The server tells the client which port to use for the data via commands sent over the data channel
- Passive mode exists to work around firewalls!

Common FTP commands

- CWD – Change working directory
- LIST – List files
- RETR – Retrieve (download)
- PWD – Present/print working directory
- CDUP – Change to parent directory
- USER/PASS – Enter username/password
- PASV – Enter passive mode

What does FTP share?

- It can share the whole filesystem if needed
  - Obviously, this is very insecure!
  - Generally, we limit users to only see their home directory (chroot_local_user config parameter)
- Anonymous transfers are treated as transfers from a user called ftp. Thus, ftp's home directory is what the server shares when an anonymous transfer takes place

Security extensions for FTP

- FTPS – FTP with SSL/TLS
- SFTP – FTP with SSH
- These are different protocols! :-)
- SFTP is more secure, but requires users to have a full user account
  - Also, often requires new server/client software

Let’s move onto SSH!

SSH

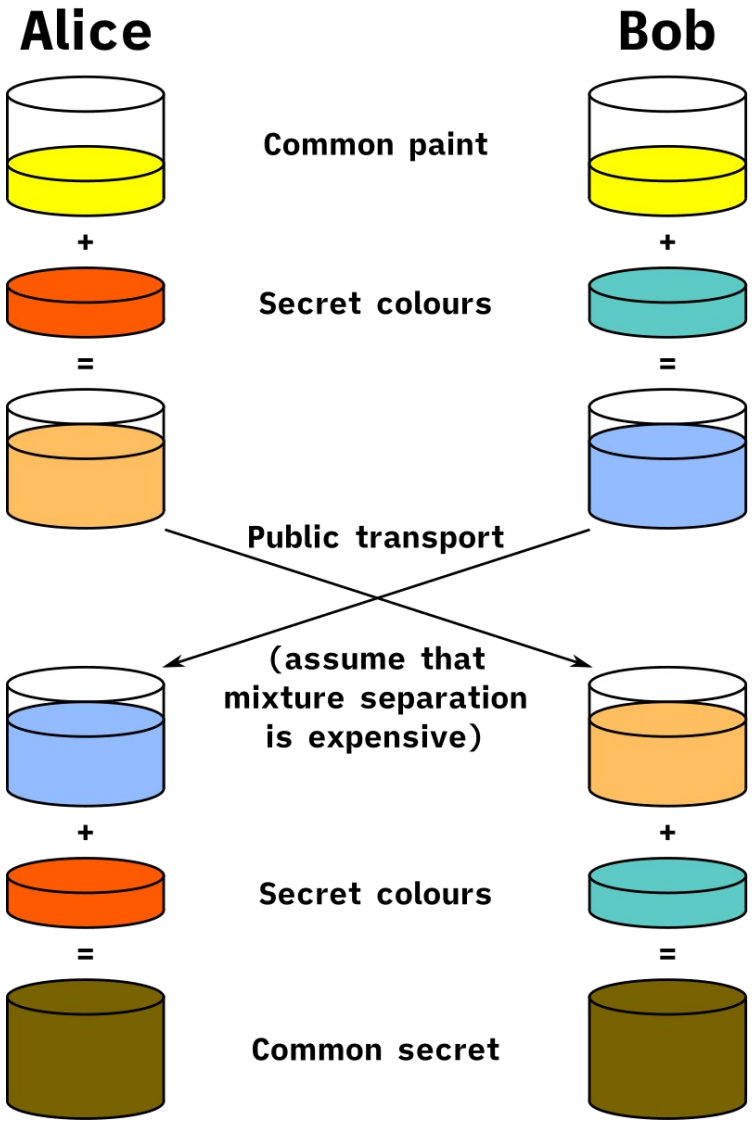
- Secure Shell
  - Originally used to allow shell sessions to take place over the network
- Has been extended to do other things (for example, you'll probably have used scp in previous labs, which uses SSH internally)
- OpenSSH is the dominant modern implementation

Cryptography terminology

- Do people know what these mean?
  - Symmetric encryption
  - Asymmetric encryption
  - Hashing

Initiating an SSH session

- The client connects to the server via port 22
- The communication actually takes place using symmetric encryption
  - Diffie-Hellman key exchange
- Optionally, an asymmetric key pair can be used instead of a password for authentication



Diffie-Hellman key exchange

Asymmetric authentication

- Create a new keypair
  - ssh-keygen -t *algorithm*
- Now ~/.ssh will contain the pair:
  - id_dsa – Private key
  - id_dsa.pub – Public key
- Contents of id_dsa.pub should be moved to ~/.ssh/authorized_keys on the server

That’s all for this week!

Thanks for your attention!