WEB SYSTEMS

Intro to Operating Systems (OS)

- Software that sits between all programs and the computer’s software
- Provides the interface between user and hardware
- Manages the computer and provides services to programs and users
- Protects users and programs from each other

Common Operating Systems (OS)

Large systems

- Pioneered in the 60s - z/OS by IBM
- Supercomputers now run Linux
- Minicomputers use openVMS, IBM OS/400 or UNIX

Personal Computers

- Linux, Microsoft Windows,Mac OX/S (Unix)

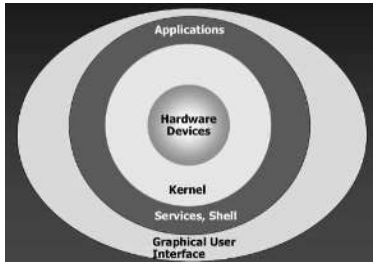
Configuration

- Configuration files normally stored for a Unix system is /etc
    - Files for a particular user on a Unix system are stored in another area - Usually in files and directories in the users ~home directory
- Parent of the linuxgym-data directory -/usr/local
- Exact path to linuxgym data directory - /course/linuxgym

Embedded Systems

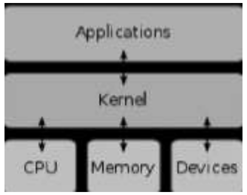
- Military, IoT, Telecomms

ONION MODEL



Hardware devices

- Central processing unit (CPU) e.g. Intel Core 2 or 7
- Memory
- Input/Output devices e.g. mouse, keyboard, monitor, printer
- Storage e.g. flash, hard drive



Kernel

- Controls the hardware directly: device drivers, firmware, etc
- Provides resources and services to applications
    - E.g. CPU, memory, storage, video, mouse, keyboard, memory, etc
- Managers access to privileged resources

Applications - programs to do something for the user

Services - programs that run behind the scenes/usually provides system support e.g. security, networking

Shell/User Interface (UI)

- Command Line Interface (CLI), cmd prompt
- Graphical user Interface (GUI)
    - User friendly interface on top of the operating system
    - Runs the shell commands transparently - WEB interface
- POSIX - Portable Operating System Interface

Interfaces

- Designing a user interface
    - Pick intended audience, Good workflow, Polished look, Consistency, Psychology/Human computer interaction

Psychology of user interfaces

- Cognitive scientists analyse how people think - to predict how people will react to certain stimulus

GUI vs CLI - neither is better - each of them has an appropriate and important role in computing

Graphical user Interface (GUI) - Interact via windows, icons, menu, pointer device (WIMP interface)

- 1983: Apple - Lisa -> Mac OS
- 1984: Unix - gnome, KDE
- 1985: Microsoft Windows 1.0
- 2001: Apple - Mac OS X & Microsoft Windows XP
- 2006: Microsoft Vista Aero
- 2010: Microsoft Metro

Strengths	Weaknesses
- Little/no experience required - Good for graphics e.g. artwork, desktop, publishing - User friendly, intuitive - Hides complexity from users	- Cant do everything - keyboard can be faster - Can crash the system and slows computer down - User is unsure of what the OS is really doing - Needs better hardware - more memory, processor, etc - Hides complexity from users

Command Line Interface (CLI) - interact through the keyboard and a monitor which only prints texts

- Sh (shell) 1969: predecessor of bash, cch
- CPM 1973: predecessor of MS-DOS
- Cmd.exe - windows shell - replaced by powershell

Strengths	Weaknesses
- Greater flexibility - combine commands - Fine tuning > parameters - Essential for system administration - Faster, less overhead - Runs on simple hardware - Can run remotely - Robust - difficult to crash	- Hard to learn - Cryptic commands & parameters - More than 1 way to do things - many options - Output can be often cryptic or non-existent - Inconsistent commands - Different versions of Unix? DOS? - No graphics - No safety net/expert mode

CLI Scripting

Batch files and Scripting languages

- You can automate CLI’s via a Batch file
    - Putting sequence of commands into an executable file
        - CLI treats the file as a command
- Most CLI’s include programming features - logic, calculations, variables, user input
- Some GUI’s have batch facilities = scripting language
    - E.g. sh (shell), bash, k shell, windows , python, applescript

Bash syntax improvements

- Integer mathematics
- Backslash escapes
- I/O redirection
- In process regular expressions

Characteristics

- Variables are usually untyped - loosely bound
  - The same variable can be used as as number or a strong
- Language syntax is often inconsistent
- Often designed and created by one person to get a particular job done
- Usually run through an interpreter, not a compiler

Evolution of scripting languages

- Usually gain extra features as they evolve
- Perl - started as scripting language > generic programming language
- Windows Shell > replaced by powershell
- Bash (Linus default CLI) > includes arrays, data types, etc

UNIX

- Used since 1969 - esp those with internet
  - Web servers, domain name servers, email servers and web hosting
- Many versions of Unix - o.g. AT&T
- Stayed alive because no one owns it and any group is free to implement and remix
  - Based on simple stuff
  - Written in the programming language, conceptually the same, usually free, very efficient, stable and relatively secure, very simple commands

FIle Systems

- Function of an operating system - manages the hardware and the software and the stuff in between
- Manages data storage and access
- Classified into:
  - ***Logical File System***
    - User view of a file system
      - Files
      - directories/subdirectories - C:\home
        - For organisational views
      - Partitions - hard drives, USB, etc, drives C;\
        - Physical divisions in the file systems
  - ***Physical File System***

Grep - 'grep' was derived from the search function of the ed unix command 'grep' is used to search/find for strings in files.

Theory of Trees

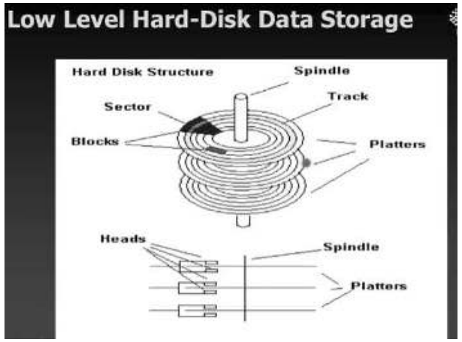
- Tree - collection of nodes - parenthood - nodes don't have parents
- Edge - “branch” of the tree - a > b, = a is parent of b
- Leaf has no children - usually a file
- Siblings usually have the same parent

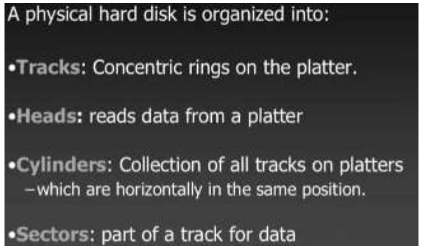
File Storage

File systems and file manipulation

How hard disks and SSD is managed and organised by an operating system:

- **Disk physical structure**
  - Tracks, heads, cylinders, sectors
  - Disk formatting - creates the **physical** disk structure/marking the surface of a disk into tracks, sectors, and cylinders
  - Platter, sector, track, cylinder


Low Level Hard-Disk Data Storage


A physical hard disk is organized into:

- **Tracks:** Concentric rings on the platter.
- **Heads:** reads data from a platter
- **Cylinders:** Collection of all tracks on platters
  - –which are horizontally in the same position.
- **Sectors:** part of a track for data

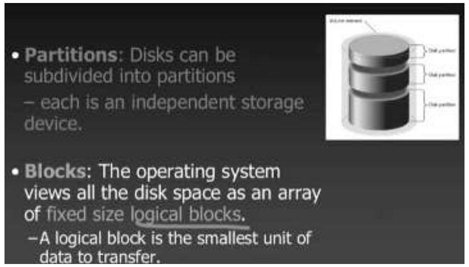
Not blocks

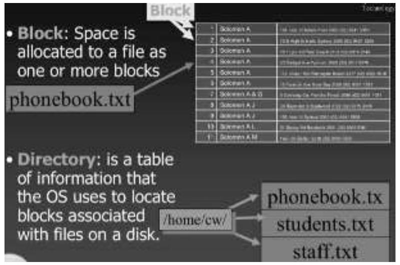
A file of 9 blocks resides on a filesystem using chained allocation. Each block contains a sequence of records. On average, how many blocks will it need to read to find a particular record. - 5 or 4.5

A file of 9 blocks resides on a filesystem using chained allocation. If the program knows that the record resides on the 7th block, how many blocks does it need to read to access the record and change it? - 7

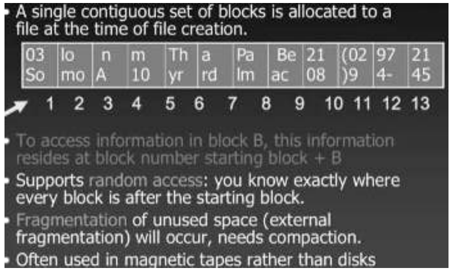
A contiguous file system has a file made up of 100 blocks, numbered from 1 to 100. How many time does the file system have to be accessed to find the 50th block? - 1

- **Disk logical structures**


- Partitions: Disks can be subdivided into partitions
  - – each is an independent storage device.
- Blocks: The operating system views all the disk space as an array of fixed size logical blocks.
  - –A logical block is the smallest unit of data to transfer.


- Block: Space is allocated to a file as one or more blocks
- Directory: is a table of information that the OS uses to locate blocks associated with files on a disk.

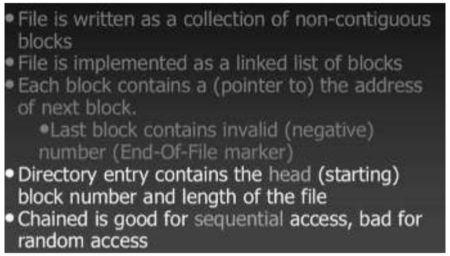
- **File allocation methods**
  - **Contiguous allocation**


- A single contiguous set of blocks is allocated to a file at the time of file creation.

03 So	lo mo	n A	m 10	Th yr	a rd	Pa lm	Be ac	21 08	(02)9	97 4-	21 45
1	2	3	4	5	6	7	8	9	10	11	12	13

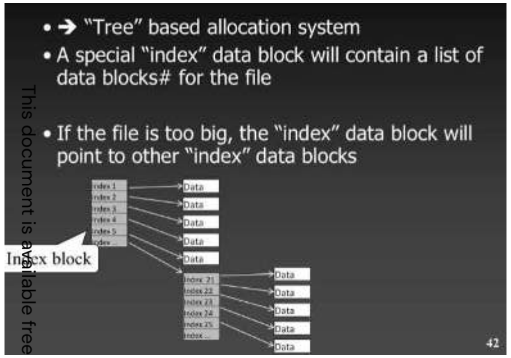
- To access information in block B, this information resides at block number starting block + B
- Supports random access: you know exactly where every block is after the starting block.
- Fragmentation of unused space (external fragmentation) will occur, needs compaction.
- Often used in magnetic tapes rather than disks

- **Chained or Linked allocation**



- File is written as a collection of non-contiguous blocks
- File is implemented as a linked list of blocks
- Each block contains a (pointer to) the address of next block.
  - Last block contains invalid (negative) number (End-Of-File marker)
- Directory entry contains the head (starting) block number and length of the file
- Chained is good for sequential access, bad for random access

- **Indexed allocation**



- → "Tree" based allocation system
- A special "index" data block will contain a list of data blocks# for the file
- If the file is too big, the "index" data block will point to other "index" data blocks

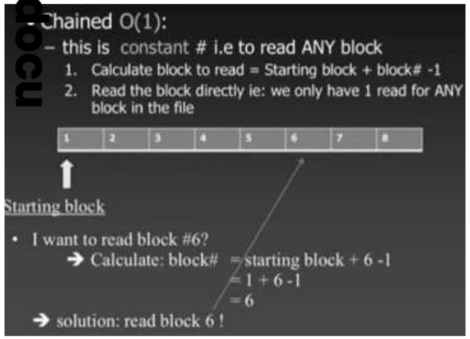
iNodes and Directories in Unix

- System in Unix
- Numbered and stores file metadata

Complexity Theory

- How much time/reads are required to find a particular block of the file?
  - Most systems run on Indexed allocation file systems

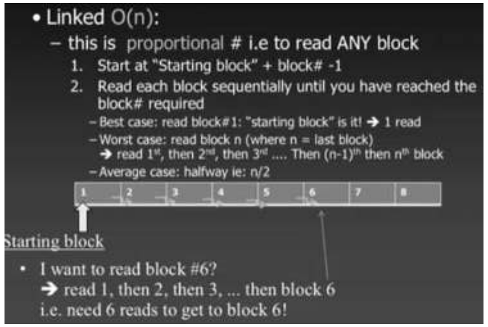
Most efficient disk allocation algorithm - contiguous



Chained O(1):
- this is constant # i.e to read ANY block
  1. Calculate block to read = Starting block + block# -1
  2. Read the block directly ie: we only have 1 read for ANY block in the file

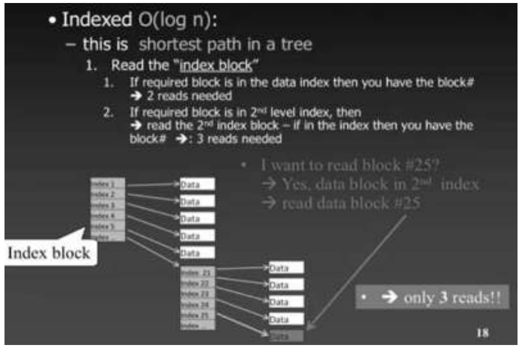
- I want to read block #6?
  → Calculate: block# = starting block + 6 -1
  = 1 + 6 -1
  = 6

→ solution: read block 6 !

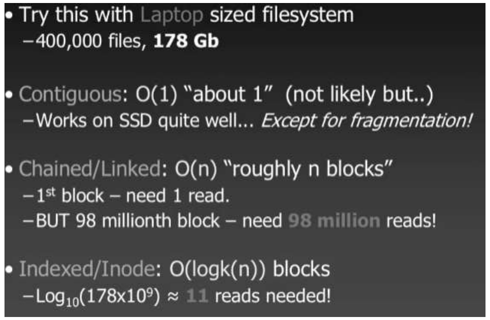


- Linked O(n):
  - this is proportional # i.e to read ANY block
    1. Start at "Starting block" + block# -1
    2. Read each block sequentially until you have reached the block# required
       - Best case: read block#1: "starting block" is it! → 1 read
       - Worst case: read block n (where n = last block)
         → read 1st, then 2nd, then 3rd .... Then (n-1)th then nth block
       - Average case: halfway ie: n/2

- I want to read block #6?
  → read 1, then 2, then 3, ... then block 6
  i.e. need 6 reads to get to block 6!



- Indexed O(log n):
  - this is shortest path in a tree
    1. Read the "index block"
       1. If required block is in the data index then you have the block#
          → 2 reads needed
       2. If required block is in 2nd level index, then
          → read the 2nd index block – if in the index then you have the block# → : 3 reads needed
  - I want to read block #25?
    → Yes, data block in 2nd index
    → read data block #25
  - → only 3 reads!!

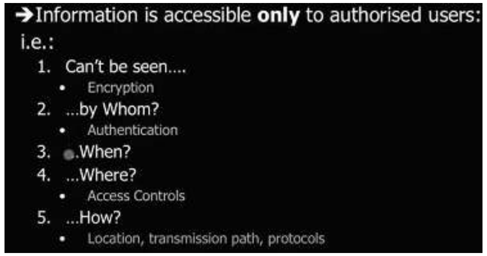


- Try this with Laptop sized filesystem
  - 400,000 files, **178 Gb**
- Contiguous: O(1) "about 1" (not likely but..)
  - Works on SSD quite well... *Except for fragmentation!*
- Chained/Linked: O(n) "roughly n blocks"
  - 1st block – need 1 read.
  - BUT 98 millionth block – need 98 million reads!
- Indexed/Inode: O(logk(n)) blocks
  - $\text{Log}_{10}(178\text{x}10^9) \approx 11$ reads needed!

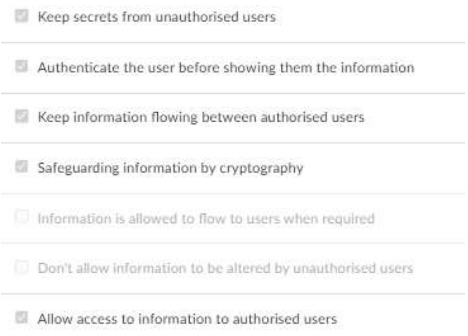
WEEK 4 More OS and Intro to the Web

Security & Encryption: Principles (CIA)

Confidentiality



→ Information is accessible **only** to authorised users: i.e.:
1. Can't be seen....
   - Encryption
2. ...by Whom?
   - Authentication
3. ...When?
4. ...Where?
   - Access Controls
5. ...How?
   - Location, transmission path, protocols



☑ Keep secrets from unauthorised users

☑ Authenticate the user before showing them the information

☑ Keep information flowing between authorised users

☑ Safeguarding information by cryptography

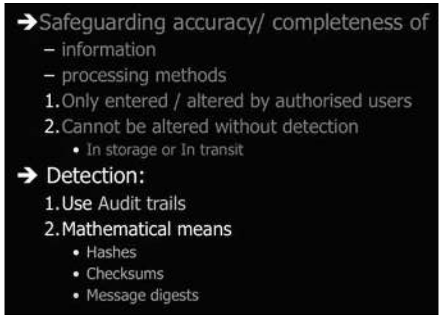
□ Information is allowed to flow to users when required

□ Don't allow information to be altered by unauthorised users

☑ Allow access to information to authorised users

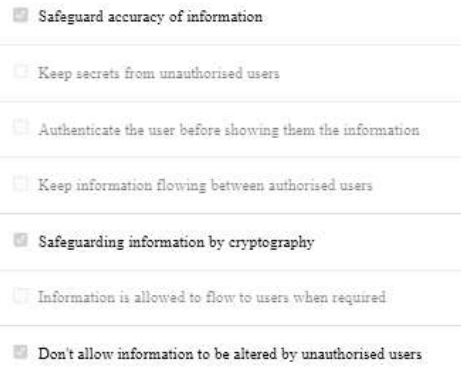
Integrity

Data integrity - check that the data has not been changed in transmission



→ Safeguarding accuracy/ completeness of
- information
- processing methods
1. Only entered / altered by authorised users
2. Cannot be altered without detection
   - In storage or In transit

→ Detection:
1. Use Audit trails
2. Mathematical means
   - Hashes
   - Checksums
   - Message digests



☑ Safeguard accuracy of information

□ Keep secrets from unauthorised users

□ Authenticate the user before showing them the information

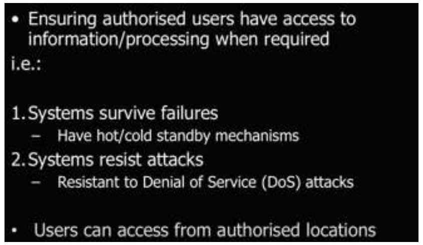
□ Keep information flowing between authorised users

☑ Safeguarding information by cryptography

□ Information is allowed to flow to users when required

☑ Don't allow information to be altered by unauthorised users

Availability

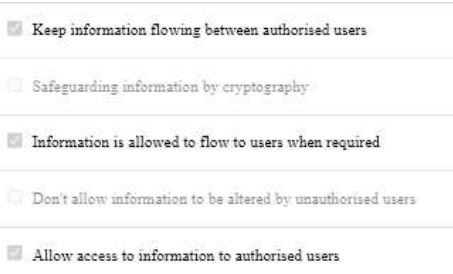


- Ensuring authorised users have access to information/processing when required

i.e.:

1. Systems survive failures
   - Have hot/cold standby mechanisms
2. Systems resist attacks
   - Resistant to Denial of Service (DoS) attacks

- Users can access from authorised locations



☑ Keep information flowing between authorised users

□ Safeguarding information by cryptography

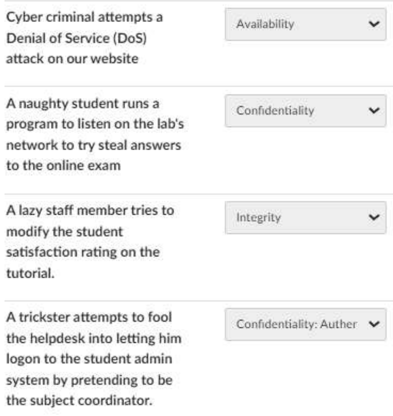
☑ Information is allowed to flow to users when required

□ Don't allow information to be altered by unauthorised users

☑ Allow access to information to authorised users

Security Service - makes use of one or more security mechanisms

Security mechanisms - designed to detect, prevent, or recover from a security attack



A cyber criminal attempts a Denial of Service (DoS) attack on our website — Availability

A naughty student runs a program to listen on the lab's network to try steal answers to the online exam — Confidentiality

A lazy staff member tries to modify the student satisfaction rating on the tutorial. — Integrity

A trickster attempts to fool the helpdesk into letting him logon to the student admin system by pretending to be the subject coordinator. — Confidentiality; Auther

Security attack - any action that compromises the security of information

- Types:

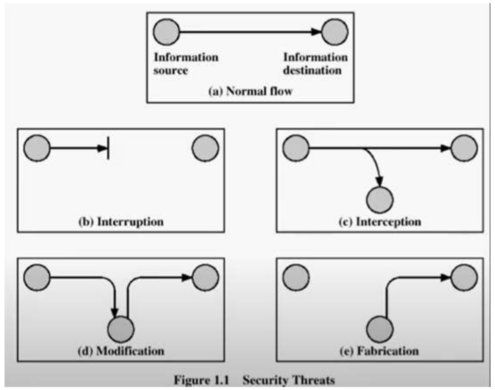


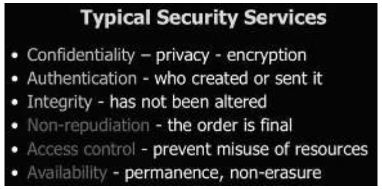
Figure 1.1 Security Threats

risk assessment would you consider if there is a change in grades - An Internal integrity threat

b) Availability attack c) Confidentiality attack d) Integrity attack e) Authenticity attack

Best tag for pre-formatted text - pre
New-line break - br
External style sheet - <link rel="stylesheet" type="text/css"href="style.css">
Change background colour for h1 - h1 { background-color:black }

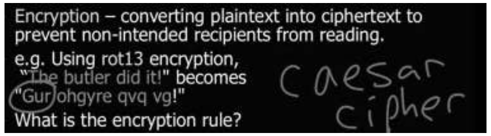
Works with the security services



Typical Security Services

- Confidentiality – privacy - encryption
- Authentication - who created or sent it
- Integrity - has not been altered
- Non-repudiation - the order is final
- Access control - prevent misuse of resources
- Availability - permanence, non-erasure

Confidentiality - Encryption

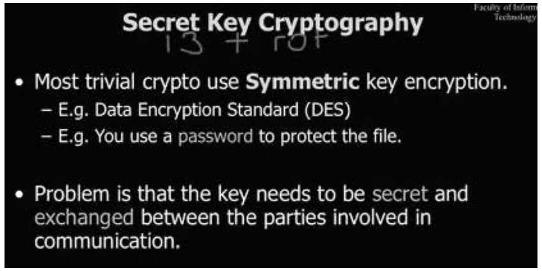


Encryption – converting plaintext into ciphertext to prevent non-intended recipients from reading.
e.g. Using rot13 encryption,
"The butler did it!" becomes
"Gur ohgyre qvq vg!"
What is the encryption rule?

Caesar cipher

- Shifts the characters by 13 - A = 1 N = 13 - so A becomes N

Secret Key Cyrptography

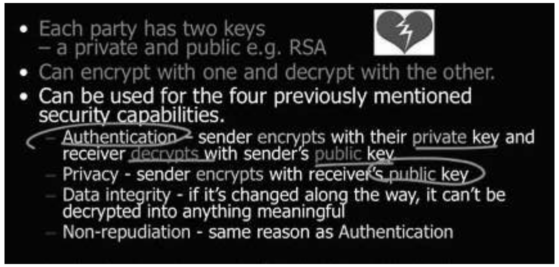


Secret Key Cryptography

13 + rot

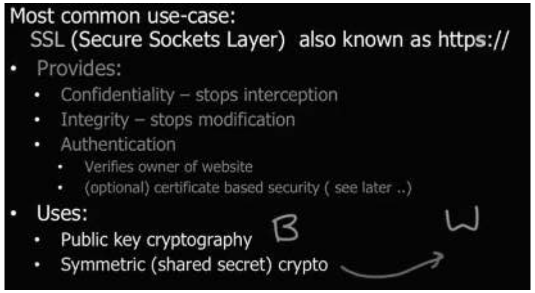
- Most trivial crypto use **Symmetric** key encryption.
  – E.g. Data Encryption Standard (DES)
  – E.g. You use a password to protect the file.
- Problem is that the key needs to be secret and exchanged between the parties involved in communication.

Public Key Cryptography



- Each party has two keys
  – a private and public e.g. RSA
- Can encrypt with one and decrypt with the other.
- Can be used for the four previously mentioned security capabilities.
  – Authentication - sender encrypts with their private key and receiver decrypts with sender's public key
  – Privacy - sender encrypts with receiver's public key
  – Data integrity - if it's changed along the way, it can't be decrypted into anything meaningful
  – Non-repudiation - same reason as Authentication

Web Security: Encryption



Most common use-case:
SSL (Secure Sockets Layer) also known as https://

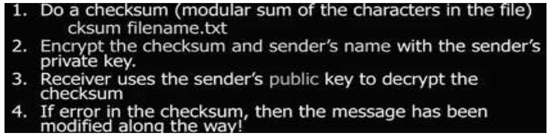
- Provides:
  - Confidentiality – stops interception
  - Integrity – stops modification
  - Authentication
    - Verifies owner of website
    - (optional) certificate based security (see later ..)
- Uses:
  - Public key cryptography
  - Symmetric (shared secret) crypto

B W

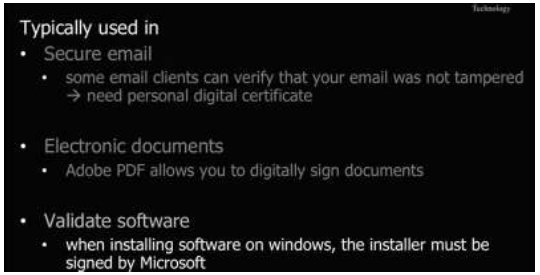
B = browser - W = web server

Web security Integrity - Hashing

- Putting code on data



1. Do a checksum (modular sum of the characters in the file)
   cksum filename.txt
2. Encrypt the checksum and sender's name with the sender's private key.
3. Receiver uses the sender's public key to decrypt the checksum
4. If error in the checksum, then the message has been modified along the way!


Typically used in
- Secure email
  - some email clients can verify that your email was not tampered → need personal digital certificate
- Electronic documents
  - Adobe PDF allows you to digitally sign documents
- Validate software
  - when installing software on windows, the installer must be signed by Microsoft

Non-repudiation - private key - paypal to prove who you are

Security Service: Authentication

How to work out who?

- Something you know
  - Pin, passwords
- Something you have
  - Keys, token, RSA tag, pin card, physical items, your phone sms (MFA)
- Something you are
  - Fingerprint, face or voice recognition, retinal scan etc

3 types of security

1. Basic authentication - pop ups of the website
   - Easy to break and hack - roth 64 (scrambled but not encrypted)

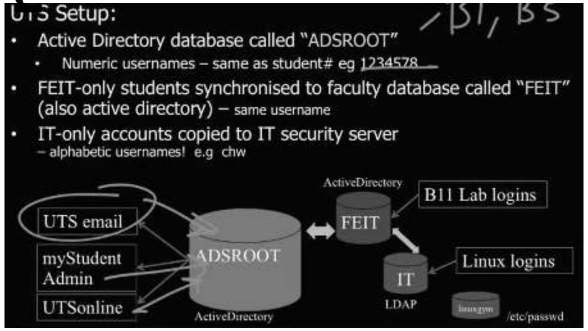
2. Web security forms - https - encrypted
3. Client side certificate - 3rd party has validated who you are and have given you an electronic keycard (file with encrypted password) that gives you access
   - E.g. AUSkey - with java

UNIX security - userid & pw

Saved in password file /etc/passwd

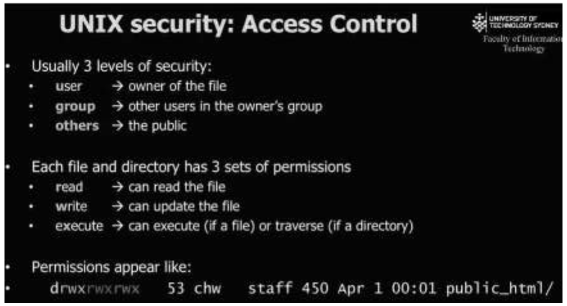
Larger scale - stored in a central directory service
- E.g. Active Directory (microsoft)
- E.g. LDAP (everyone else)

UTS Setup


UTS Setup: /B1, B5
- Active Directory database called "ADSROOT"
  - Numeric usernames – same as student# eg 1234578
- FEIT-only students synchronised to faculty database called "FEIT" (also active directory) – same username
- IT-only accounts copied to IT security server – alphabetic usernames! e.g chw

Access control

- Physical - key card, pin
- Logical - firewall, an application - needs configuration


UNIX security: Access Control

University of Technology Sydney
Faculty of Information Technology

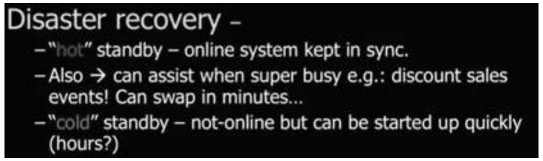
- Usually 3 levels of security:
  - user → owner of the file
  - group → other users in the owner's group
  - others → the public
- Each file and directory has 3 sets of permissions
  - read → can read the file
  - write → can update the file
  - execute → can execute (if a file) or traverse (if a directory)
- Permissions appear like:

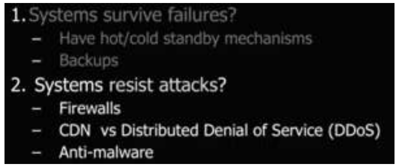
```
drwxrwxrwx    53 chw    staff 450 Apr 1 00:01 public_html/
```

Security mechanisms: audit logs

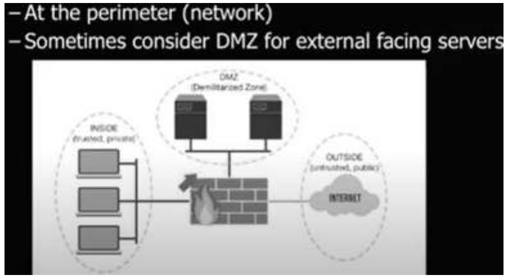
- Audit trails/logs are essential → Needed to measure effectiveness, Do forensics, Create alerts

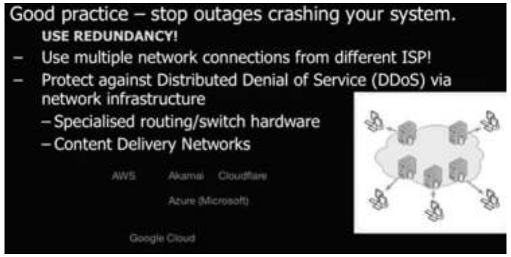
C-I-Availability - Ensuring authorised users have access to info when required


Disaster recovery –
- "hot" standby – online system kept in sync.
- Also → can assist when super busy e.g.: discount sales events! Can swap in minutes...
- "cold" standby – not-online but can be started up quickly (hours?)


1. Systems survive failures?
   - Have hot/cold standby mechanisms
   - Backups
2. Systems resist attacks?
   - Firewalls
   - CDN vs Distributed Denial of Service (DDoS)
   - Anti-malware

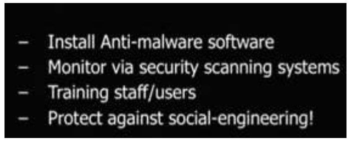
Firewalls - stops threats from crashing your system


- At the perimeter (network)
- Sometimes consider DMZ for external facing servers


Good practice – stop outages crashing your system.

USE REDUNDANCY!
- Use multiple network connections from different ISP!
- Protect against Distributed Denial of Service (DDoS) via network infrastructure
  - Specialised routing/switch hardware
  - Content Delivery Networks

AWS
Akamai
Cloudflare
Azure (Microsoft)
Google Cloud


- Install Anti-malware software
- Monitor via security scanning systems
- Training staff/users
- Protect against social-engineering!

Virus, trojans and worms are malware