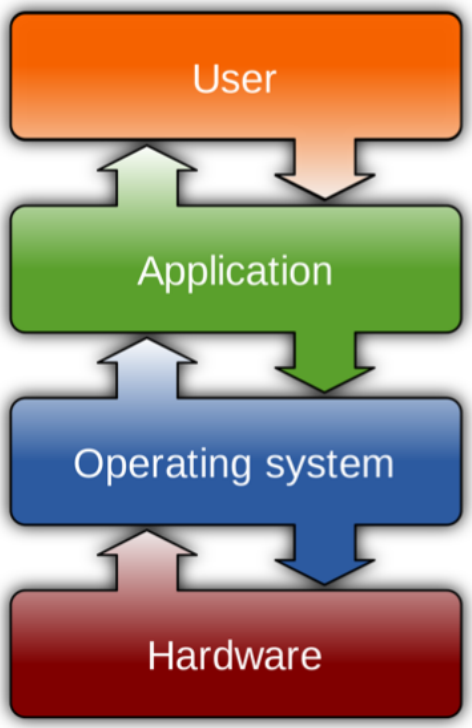
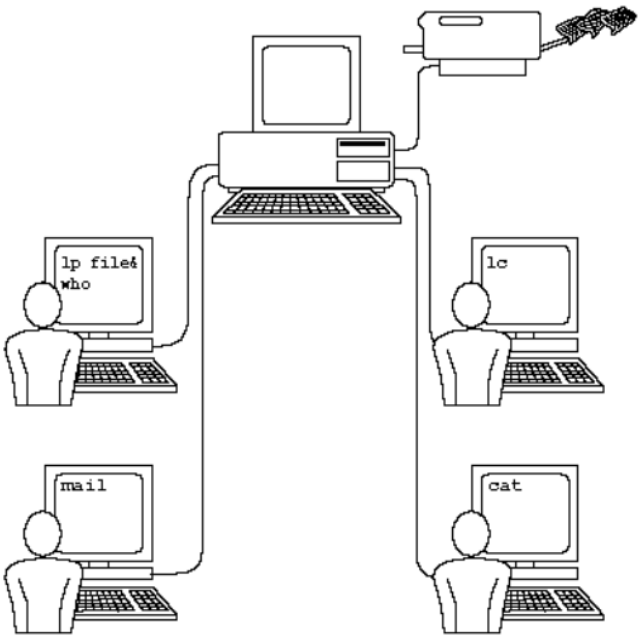
CS Revision Lecture 17, 18

- ## Lecture 17 - Operating Systems : Key concepts & security principles
  - ### Operating Systems
    - An OS provides the interface between the users of a computer and that computer's hardware
    - The OS handles the management of low-level hardware resources:

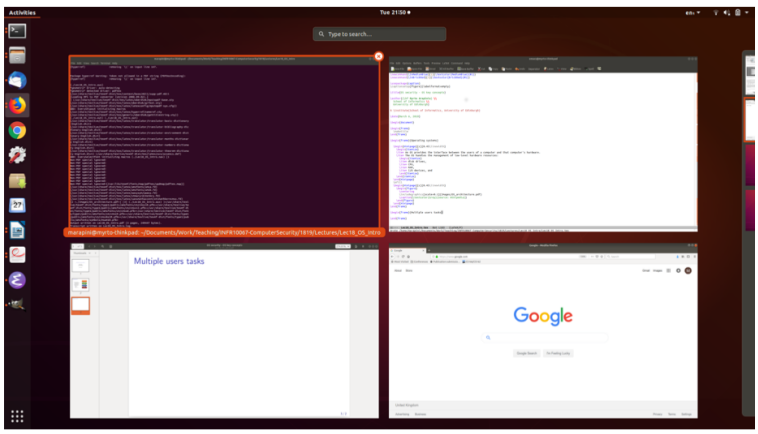


      - disk drives
      - CPU
      - RAM
      - I/O devices
      - Network interfaces
  - ### Multi-users
    - OS must allow for multiple users with potentially different levels of access to the same computer

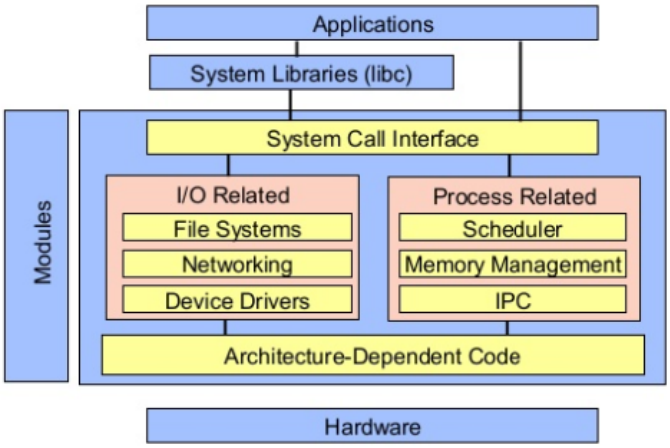


- The OS needs to have in place mechanisms to isolate different users

- Multi-tasking
  - OS must allow multiple application programs to run at the same time

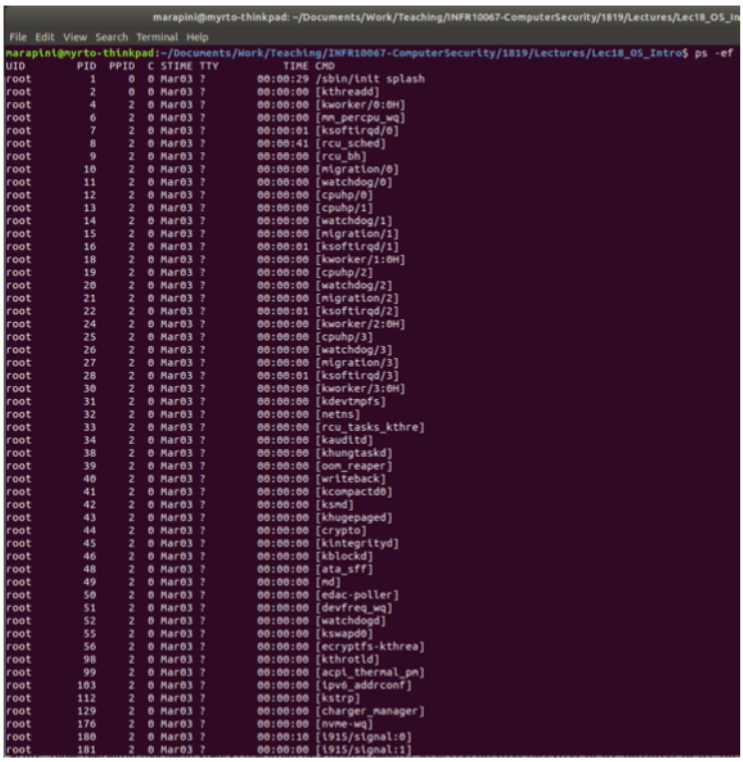


  - The OS needs to have in place mechanisms to **isolate different applications running**
- Essential Unix architecture



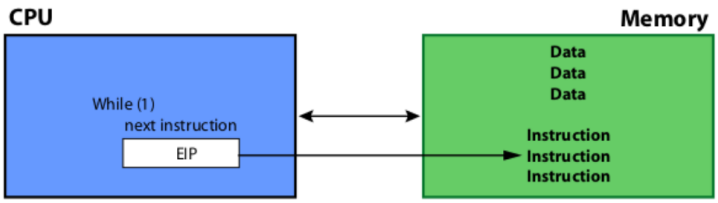
- Kernel
  - key component of the OS
  - supports secure sharing of low-level resources between users/applications
  - kernel limits how applications access computer resources
- Execution modes
  - User mode -access to resources through syscall to kernel
  - Kernel mode - direct access to resources
- **System calls** are usually contained in a collection of programs, e.g. a library such as the C library libc:
  - open() close() read() write()
  - wait() fork() exec() exit()

- Processes and process management



- A process is an instance of a program that is currently executing
- To actually be executed the program must be loaded intoRAM and uniquely identified
- Each process running is identified by a **unique process ID (pid)**
- To a pid, we can associate its CPU time, memory usage, userID (uid), program name, etc
- A process might control other processes (fork)
- Child process inherits context from parent process

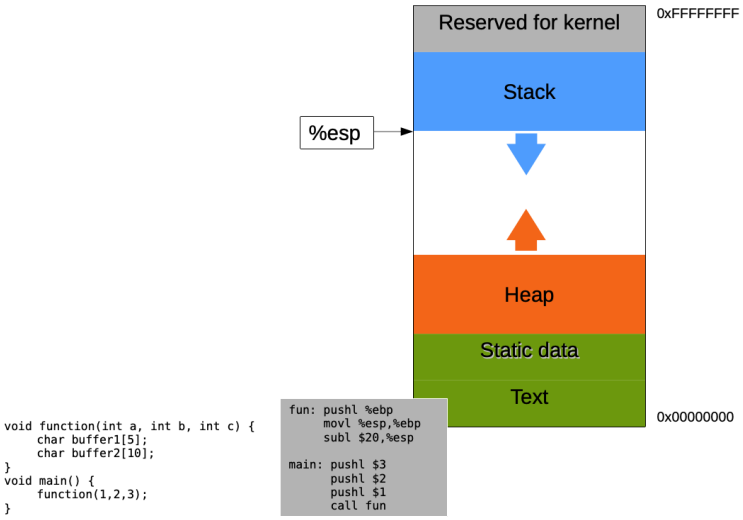
- x86 CPU/Memory



- To actually be executed the program must be loaded into RAM and uniquely identified

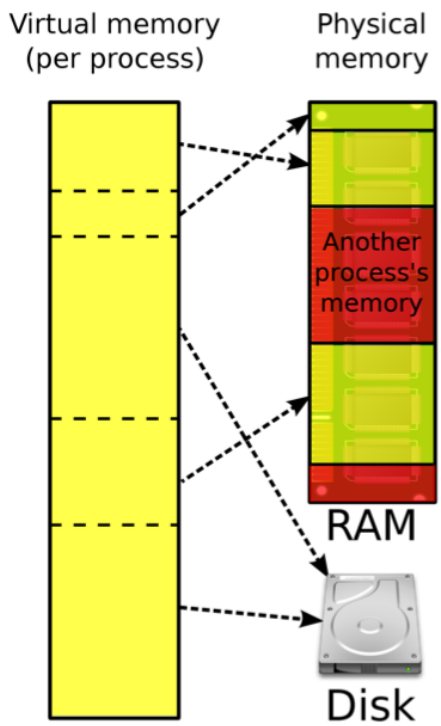
- The RAM memory allocated to a process is its address space
- It contains both the code for the running program, its input data and its working memory
- CPU interprets instructions - **%eip points to next instruction**

- x86 process memory layout(simplified)



- Stack: from top to bottom
- Heap: from bottom to top

- Virtual memory



- Common technique used in a computer's OS
- Sometimes **available RAM is not enough to run several programs at one time**. This is where virtual memory comes in
- A system using virtual memory uses a section of the hard drive to emulate RAM - secondary memory treated as though it were main meory
- A **memory management unit (MMU)** maps a logical address space to a corresponding physical address

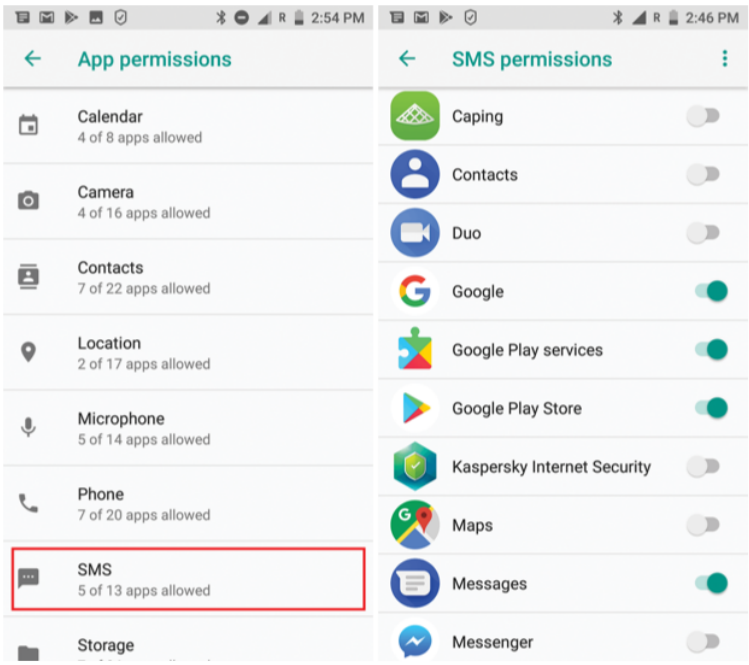
- Live CD attacks on memory
  - The attack:
    - Attacker with physical access to computer powers off the computer (without properly shutting down)
    - Attacker boots to different OS via external media
    - Attacker retrieves the *Pagefile.sys, Swapfile.sys, Hiberfile.sys* files
    - Attacker gains access to passwords and sensitive information that were stored in memory
  - Mitigation:
    - Hard disk encryption must be used!
- Security principles
  - Defence-in-depth

- Security protections built in multiple layers of the system: if one mechanism fails, another steps up immediately behind to thwart attacks
- Firewalls, intrusion detection and protection system, network segmentation, anti-virus, least privilege, strong password, path management

- Least privilege
  - Users and programs should only access the data and resources required to perform its function

  - A torch application does not need access to your location, photos, camera, microphone, wifi, device id, to perform its intended task!

- Privilege separation



  - Segment the system into components to which we can limit access
  - Will limit the damage caused by a security break of any individual component
- Open design
  - The security of a mechanism should not depend on its secrecy

- The design and implementation details always get leaked

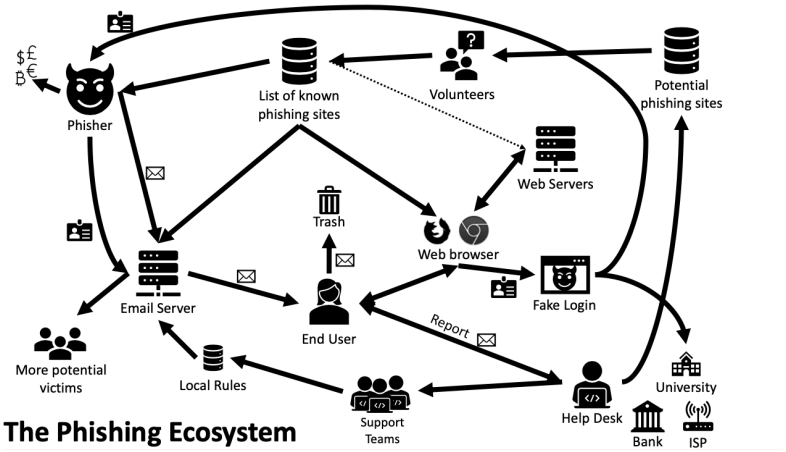
- Economy of mechanism
  - When designing a security mechanism keep it simple
  - It will facilitate the job of security researchers and allow verification
  - It will facilitate the task of developers and avoid bugs
  - It will facilitate the life of users and avoid misuses
- Fail-safe defaults
  - default configuration should be conservative, eg. new user should be granted least privileges by default
- Complete mediation
  - every access to a resource must be checked for compliance with security policy
- Usable security
  - UIs and security mechanisms should be designed with the ordinary user in mind – the users should be supported in interacting in a secure way with the system – you can't blame users !

- What we learned today
  - Many tasks handled by the OS relate to fundamental security problems
  - 1. OS concepts
    - basic tasks of the OS
    - security concerns arise from multiple users and multiple processes
    - processes and process management
    - x86 runtime memory
  - 2. Security design principle
    - Defence-in-depth
    - Least privilege

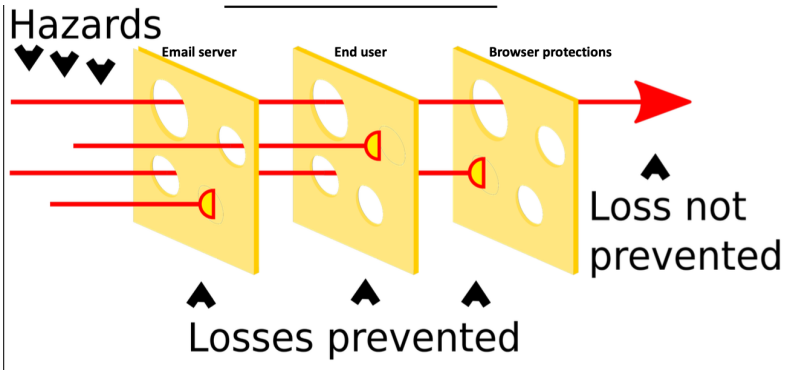
        - Privilege separation
        - Open design
        - Economy of mechanism

Lecture 18 - Usable Security and Phishing

- Outline
    - What is "usable security"?
    - An explanation of phishing
    - Authentication in brief
    - Passive vs. active indication
- Usable security is challenging because:
    - Users are unmotivated to care about security over their current task
    - Complex configurations make sense to computer scientists but not to end users
    - Good feedback/advice is very hard to give to users
    - Barn door - once security or privacy is lost, its gone. There is no undo
- Phishing
    - Phishing is very common and very disruptive to UK businesses
    - Also, it really annoys those of us who are just trying to get our work done.
- Phishing Ecosystem

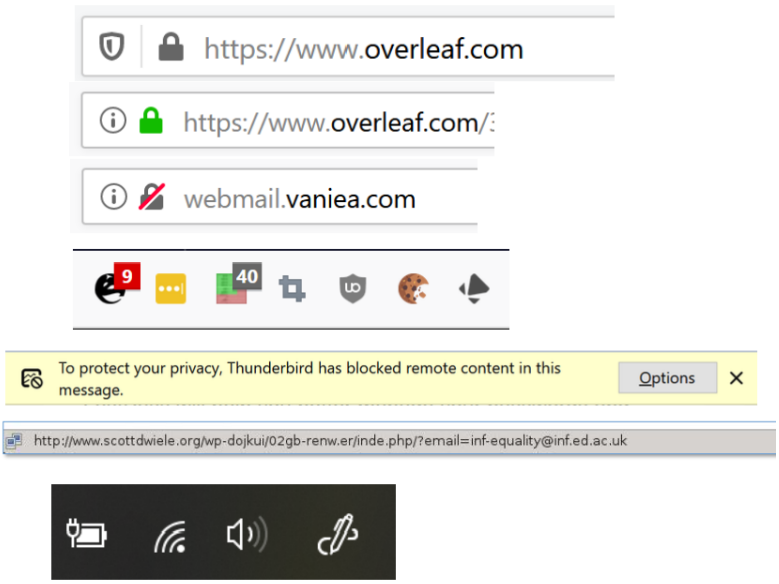


- Swiss Cheese Model



- Main Solutions
  - **Automatically block attacks using filters**
    - Stop email from even arriving in inboxes
    - Block people from visiting known bad websites
  - **Train users**
    - Provide users with training on how to identify phishing attacks
  - **Support users**
    - Show UI indicators to help users tell the difference between real and fake sites
      - Also known as “passive indicators”, like the lock icon
    - Provide feedback when phishing is reported or blocked
  - **Improve protection of authentication credentials**
    - Make it harder to impossible for a user to give away credentials

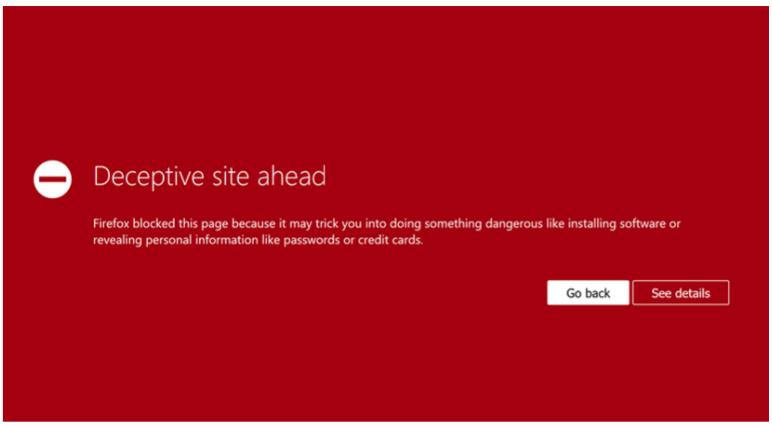
        - Limit the damage of credential sharing to one transaction
        - Let users authenticate websites
- Why does phishing work? Authentication is very broken
    - Authentication is how Entity A proves their identity to Entity B
    - We normally think of authentication as one directional
    - But it is actually two directional
        - The user must first make sure they are interacting with the “correct” website. Then the website must make sure that they are interacting with the “correct” user.
- Phishing Support(a history lesson)
    - Passive indicator



        - A UI element that provides information, but the user is not forced to look at or interact with.
    - Phishing moves to email
        - Phishing moved off AOL and onto the less secure email. Directing people to fake sites, particularly fake financial sites.
        - Massive rise in identity theft
        - Financial loss skyrocketing

- Low conviction rate with "1-in-700 chance of escaping capture"
- Burdon falling on consumers

- Recommend:
  - Businesses should take security seriously
  - Financial organizations should auto identify fraudulent applications
  - Reduce impact on consumers
- Passive security indicators were not working at the level researchers wanted.
- So indicators started getting more obvious and intrusive.
- Active Indicator



  - A UI element that interrupts the user's activity and demands a response.
  - Active indicators work better than passive ones in terms of helping people avoid phishing.
  - **Click through rates**
    - "Click through" – when a user sees a warning and chooses to proceed anyway.
    - Willingness to use Linux or use nightly builds of browsers indicates users are more willing to click through warnings.
    - A huge downside of active indicators is "habituation"where the user starts learning that the

        - warnings always happen and starts ignoring them.
    - **Active indicators are alive and well in 2022**
        - Screenshot of Santander payment page
        - Asks payment purpose, then gives specific advice based on answer
        - More customized to user needs, but still likely ignored
- Where are users learning about security?
    - Users follow advice from sources they trust
    - Users self-evaluate advice they feel they understand, like password advice
    - Marketing material in the advice results in less trust
    - High socioeconomic users get phishing training at work and tend to follow it
    - Low socioeconomic users tend to get and follow advice from friends and service providers (ISPs)
- Many different types of advice given in guidance
    - **Lookout for:**
        - Requests for sensitive data
        - Poor grammar
        - Unusual senders
        - Use of an alarming tone
        - Has a link to a website
        - Content is account related
        - Content too good to be true
    - **Protection actions:**
        - Check the URL in the address bar
        - Check for HTTPS
        - Type the URL yourself
        - Check for a lock icon
        - Bookmark sensitive websites

- Inclusion
  - Usable security
    - Harder than it looks
  - Phishing only requires one side of the two way authentication to fail
  - Passive indicators
    - Show information but do not block user tasks
    - Are easily ignored by users
  - Active indicators
    - Block the user till they interact with the dialog in some way
    - Much more effective than passive indicators
    - Lead to habituation if a user sees the warning frequently, they stop reading it

以上内容整理于 幕布文档