CSCI-620

Introduction

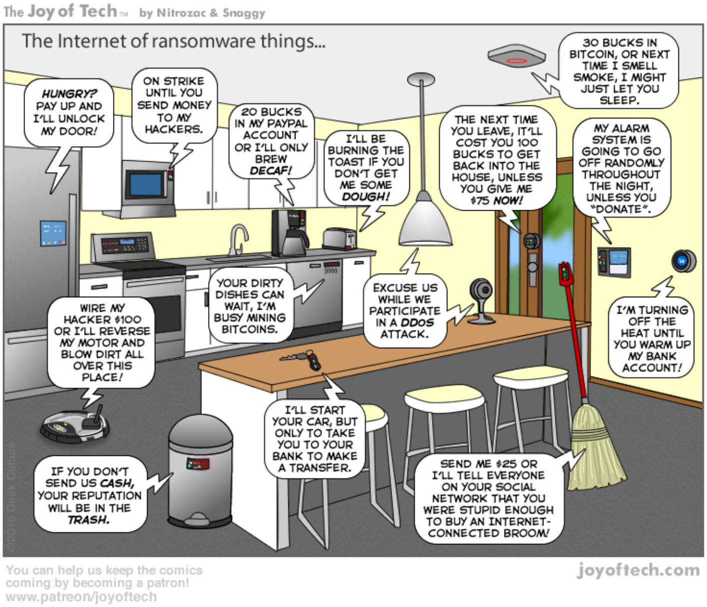
Operating Systems

- Operating system security forms the foundation of the secure operation of computer systems.

Operating Systems

- Operating system security forms the foundation of the secure operation of computer systems.
- Operating systems:
  - Provide access to system resources such as CPU, memory, and devices
  - Mediate interactions between programs
  - Mediate user interaction with the system
  - . . .

Everything Runs Software These Days...


The Joy of Tech™ by Nitrozac & Snaggy
The Internet of ransomware things...
HUNGRY? PAY UP AND I'LL UNLOCK MY DOOR!
ON STRIKE UNTIL YOU SEND MONEY TO MY HACKERS.
20 BUCKS IN MY PAYPAL ACCOUNT OR I'LL ONLY BREW DECAF!
I'LL BE BURNING THE TOAST IF YOU DON'T GET ME SOME DOUGH!
THE NEXT TIME YOU LEAVE, IT'LL COST YOU 100 BUCKS TO GET BACK INTO THE HOUSE, UNLESS YOU GIVE ME $75 NOW!
30 BUCKS IN BITCOIN, OR NEXT TIME I SMELL SMOKE, I MIGHT JUST LET YOU SLEEP.
MY ALARM SYSTEM IS GOING TO GO OFF RANDOMLY THROUGHOUT THE NIGHT, UNLESS YOU "DONATE".
YOUR DIRTY DISHES CAN WAIT, I'M BUSY MINING BITCOINS.
EXCUSE US WHILE WE PARTICIPATE IN A DDOS ATTACK.
I'M TURNING OFF THE HEAT UNTIL YOU WARM UP MY BANK ACCOUNT!
WIRE MY HACKER $100 OR I'LL REVERSE MY MOTOR AND BLOW DIRT ALL OVER THIS PLACE!
I'LL START YOUR CAR, BUT ONLY TO TAKE YOU TO YOUR BANK TO MAKE A TRANSFER.
IF YOU DON'T SEND US CASH, YOUR REPUTATION WILL BE IN THE TRASH.
SEND ME $25 OR I'LL TELL EVERYONE ON YOUR SOCIAL NETWORK THAT YOU WERE STUPID ENOUGH TO BUY AN INTERNET-CONNECTED BROOM!
©2016 Geek Culture
You can help us keep the comics coming by becoming a patron! www.patreon/joyoftech
joyoftech.com

Operating Systems

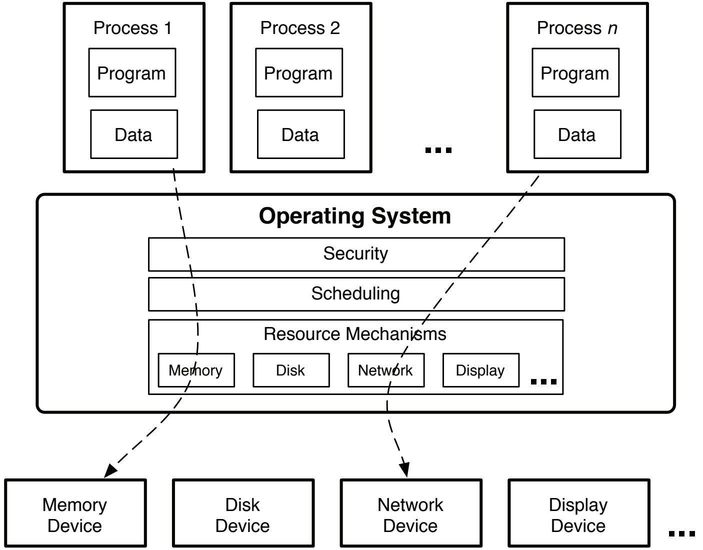
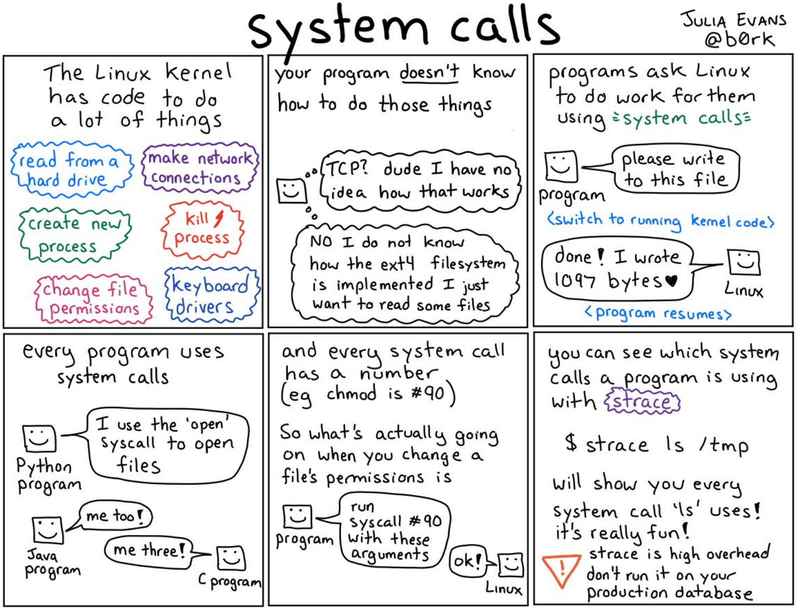


Figure 1.1: An operating system runs *security*, *scheduling*, and *resource mechanisms* to provide *processes* with access to the computer system's resources (e.g., CPU, memory, and devices).

Operating Systems



system calls

JULIA EVANS
@b0rk

The Linux kernel has code to do a lot of things

read from a hard drive

make network connections

create new process

kill process

change file permissions

keyboard drivers

your program doesn't know how to do those things

TCP? dude I have no idea how that works

NO I do not know how the ext4 filesystem is implemented I just want to read some files

programs ask Linux to do work for them using system calls

please write to this file

program

<switch to running kernel code>

done! I wrote 1097 bytes ♥

Linux

<program resumes>

every program uses system calls

I use the 'open' syscall to open files

Python program

me too!

Java program

me three!

C program

and every system call has a number (eg chmod is #90)

So what's actually going on when you change a file's permissions is

run syscall #90 with these arguments

program

ok!

Linux

you can see which system calls a program is using with strace

$ strace ls /tmp

will show you every system call 'ls' uses! it's really fun!

strace is high overhead don't run it on your production database

Why Operating Systems Security

- Originally: one process at a time (e.g., batch systems)
- 1960s: concurrent/timasharing systems
  - Once CPU, multiple processes running at the same time
  - Processes might be from different users
  - Processes might work on *confidential* data
  - Processes might (accidentally) try to interfere with each other
- Additionally, operating systems are now:
  - Highly interconnected
  - Might run code with different levels of trust (Web browser, Facebook app, . . .)

Operating Systems Major Tasks and Responsibilities

1. Provide various mechanisms that enable high performance use of computer resources
   - Provide efficient resource mechanisms, such as file systems, memory management systems, network protocol stacks, etc., that define how processes use the hardware resources
2. Switch among the processes fairly, such that the user experiences good performance from each process in concert with access to the computer's devices
3. Control access to resources, such that one process cannot (inadvertently or maliciously) impact the execution of another

Operating Systems Major Tasks and Responsibilities

- Operating Systems must provide proper boundaries between processes
  - E.g., a file system must not allow a process request to access one file to overwrite the disk space allocated to another file
- Scheduling mechanisms must ensure availability of resources to processes to prevent denial of service attacks
  - All processes are eventually scheduled for execution

Operating Systems Major Tasks and Responsibilities

- Using a computer is ultimately about various processes interacting with each other
  - User writes program's source code with an editor
  - Source code is then read by compiler
  - Output of compiler is processed by linker
  - Debugger oversees the execution of the program
- The challenge in developing operating systems security is to design security mechanisms that protect process execution despite complex interactions.

Operating Systems Security

In the context of this course, *security* relates to computing or communicating in the presence of adversaries

Current operating systems address this in one of two ways

1. Constrained systems that can enforce security goals with a high degree of assurance
   - Support few applications
   - Applications often have limited functionality and lower performance requirements
   - Security is the top priority
2. General-purpose systems that can enforce limited security goals with a low to medium degree of assurance

Computer Security Concepts

Computer Security Concepts

The NIST Computer Security Handbook defines the term computer security as follows:

> *Computer Security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).*

Computer Security Goals

1. **Confidentiality**
   - **Data confidentiality**: Assures that private or confidential information is not made available or disclosed to unauthorized individuals
   - **Privacy**: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed
2. **Integrity**
   - **Data integrity**: Assures that information and programs are changed only in a specified and authorized manner
   - **System integrity**: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system
3. **Availability**
   - Assures that systems work promptly and service is not denied to authorized users

Computer Security Goals

1. **Authenticity**: The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.
2. **Accountability**: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action. Because truly secure systems are not yet an achievable goal, we must be able to trace a security breach to a responsible party. Systems must keep records of their activities to permit later forensic analysis to trace security breaches or to aid in transaction disputes.

Computer Security

1. Not simple
2. Must consider potential attacks
3. Procedures used to provide service might be counter-intuitive
4. Must decide where to deploy mechanisms and algorithms
5. Involve secret information
6. Attacker only needs to find a single weakness (defender must find all weaknesses)
7. Users do not see benefits until system fails
8. Requires regular monitoring
9. Too often an after-thought
10. Regarded as impediment to using system

Vulnerabilities and Attacks

System resource may become:

- Corrupted (loss of integrity)
- Leaky (loss of confidentiality)
- Unavailable (loss of availability)

Vulnerabilities and Attacks

Threats

- Capable of exploiting vulnerabilities
- Represent potential security harm to an asset

Attacks (threats carried out)

- **Passive:** does not affect system resources
- **Active:** attempt to alter system resources or affect their operation
- **Insider:** initiated by an entity inside the security parameter
- **Outsider:** initiated from outside the perimeter

Countermeasures

- Means used to deal with security attacks
  - Prevent
  - Detect
  - Recover
- May result in new vulnerabilities
- Will have residual vulnerability
- Goal is to minimize risk given constraints

The Ten Immutable Laws of Computer Security

- Law #1: If a bad guy can persuade you to run his program on your computer, it's not solely your computer anymore.
- Law #2: If a bad guy can alter the operating system on your computer, it's not your computer anymore.
- Law #3: If a bad guy has unrestricted physical access to your computer, it's not your computer anymore.
- Law #4: If you allow a bad guy to run active content in your website, it's not your website any more.
- Law #5: Weak passwords trump strong security.
- Law #6: A computer is only as secure as the administrator is trustworthy.
- Law #7: Encrypted data is only as secure as its decryption key.
- Law #8: An out-of-date anti-malware scanner is only marginally better than no scanner at all.
- Law #9: Absolute anonymity isn't practically achievable, online or offline.
- Law #10: Technology is not a panacea.

`https://technet.microsoft.com/en-us/library/hh278941.aspx`

Threat Consequences

- Unauthorized disclosure
  - Exposure, interception, inference, intrusion
- Deception
  - Masquerade, falsification, repudiation
- Disruption
  - Incapacitation, corruption, obstruction
- Usurpation, misappropriation, misuse

Secure Operating Systems

Secure Operating Systems

A secure operating system provides security mechanisms that ensure that the system's security goals are enforced despite the threats faced by the system.

Trusted Computing Base (TCB)

- Minimal amount of software necessary to enforce the security goals correctly
  - Software that **defines** the security goals, and software that **enforces** the security goal
  - Software that bootstraps this software must also be trusted
- Ideal TCB: bootstrapping mechanism that enables the security goals to be loaded and then enforced for lifetime of the system

Trusted Computing Base (TCB)

- TCB must mediate all security-sensitive operations
- Verification of the correctness of the TCB software and its data
- Verification that the software's execution cannot be tampered by processes outside the TCB
- TCB Codebase should be very small
  - Enables formal verification
  - Minimizes number of bugs

Trusted Computing Base (TCB)

In practice, TCB consists of wide variety of code:

- Enforcement mechanisms are part of the OS
  - In monolithic OSs, enforcement mechanisms must trust the entire OS code, which is therefore part of the TCB
- Further, a variety of other software running outside the operating system must also be trusted.
  - Login, SSH, graphics subsystem, windowing system, clipboard managers, various libraries, . . .

Secure OS Trust Model

- An OS cannot trust processes outside of the TCB to behave as expected
- This seems pretty obvious

Secure OS Trust Model

- An OS cannot trust processes outside of the TCB to behave as expected
- This seems pretty obvious
- However, this is not how most modern systems work
  - When a user runs a program, OS assumes that the program behaves as the user would

Secure OS Trust Model

- An OS cannot trust processes outside of the TCB to behave as expected
- This seems pretty obvious
- However, this is not how most modern systems work
  - When a user runs a program, OS assumes that the program behaves as the user would
  - This is not true if, e.g., a program has been subverted by a remote vulnerability, or via a macro virus

Actual (Commercial) Operating Sysrtems

- Two kinds of operating systems
  1. Constrained, very secure systems
  2. General purpose systems with a low level of security assurance
- Recent operating systems (e.g., Android, iOS) try to strike a better balance, primarily because of limited OS legacy
- However, all complex operating systems have bugs, which typically undermine their security