OPERATING SYSTEMS

Input - Output Management and Security - 7

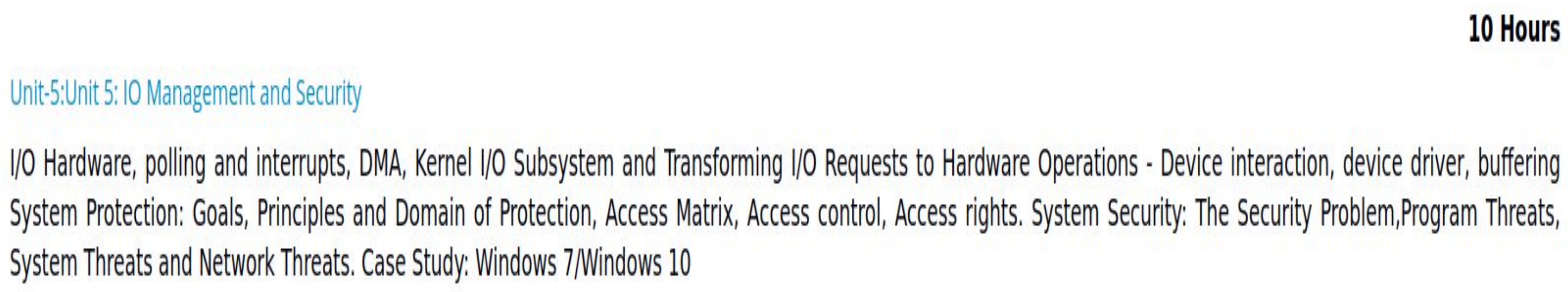
Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

The Program Threats

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

Course Syllabus - Unit 5



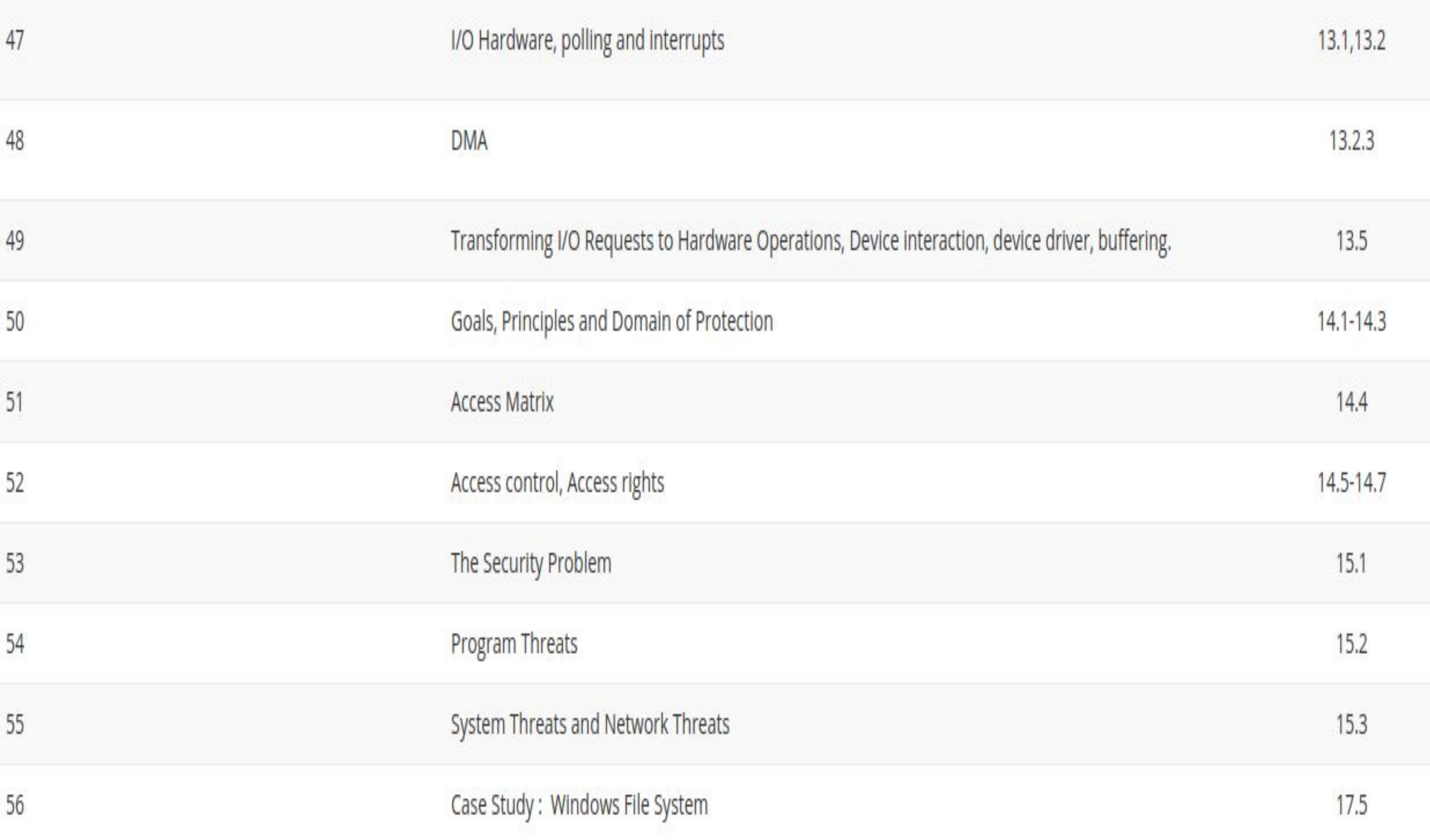
10 Hours

Unit-5:Unit 5: IO Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10

OPERATING SYSTEMS

Course Outline



47	I/O Hardware, polling and interrupts	13.1,13.2
48	DMA	13.2.3
49	Transforming I/O Requests to Hardware Operations, Device interaction, device driver, buffering.	13.5
50	Goals, Principles and Domain of Protection	14.1-14.3
51	Access Matrix	14.4
52	Access control, Access rights	14.5-14.7
53	The Security Problem	15.1
54	Program Threats	15.2
55	System Threats and Network Threats	15.3
56	Case Study : Windows File System	17.5

Topic Outline

- The Program Threats

The Program Threats

- Processes, along with the kernel, are the only means of accomplishing some appropriate task on a computer.

- Writing a program that creates a breach of security, or causing a normal process to change its behavior and create a breach, is a common goal of crackers.

- While it is useful to log in to a system without authorization, it is quite a lot more useful to leave behind a **back-door** daemon that provides information or allows easy access even if the original exploit is blocked.

The Program Threats: Trojan Horse

- A code segment that misuses its environment is called a **Trojan horse**.

- Long search paths, such as are common on UNIX systems, **exacerbate** the Trojan horse problem.

- All the directories in such a search path must be secure, or a Trojan horse could be slipped into the user's path and executed accidentally.

- If a user has "." in his / her search path, has set his / her current directory to a friend's directory, and enters the name of a normal system command, the command may be executed from the friend's directory.

- The program will run within the user's domain, allowing the program to do anything that the user is allowed to do, including deleting the user's files

The Program Threats: Trojan Horse

- A variation of the Trojan horse is a program that emulates a login program.

- An unsuspecting user starts to log in at a terminal and notices that has apparently mistyped his password and tries again and is successful.

- Thus authentication key and password have been stolen by the login emulator, which was left running on the terminal by the thief.

- The emulator stored away the password, printed out a login error message, and exited; the user was then provided with a genuine login prompt.

The Program Threats: Trojan Horse

- Variation on the Trojan horse is **Spyware**.

- Spyware sometimes accompanies a program that the user has chosen to install.

- Most frequently, it comes along with freeware or shareware programs, but sometimes it is included with commercial software.

- The goal of spyware is to download ads to display on the user’s system, create pop-up browser windows when certain sites are visited, or capture information from the user’s system and return it to a central site.

- This latter practice is an example of a general category of attacks known as covert channels, in which surreptitious communication occurs.

The Program Threats: Trojan Horse

- The installation of an innocuous seeming program on a Windows system could result in the loading of a spyware daemon.
- The spyware could contact a central site, be given a message and a list of recipient addresses, and deliver a spam message to those users from the Windows machine.
- This process continues until the user discovers the spyware.
- Frequently, the spyware is not discovered.
- In 2010, it was estimated that 90 percent of spam was being delivered by this method.
- This theft of service is not even considered a crime in most countries

The Program Threats: Trojan Horse

- Spyware is a micro example of a macro problem: violation of the principle of least privilege.

- Under most circumstances, a user of an operating system does not need to install network daemons.

- Such daemons are installed via two mistakes.
  - First, a user may run with more privileges than necessary for example, as the administrator, allowing programs that he / she runs to have more access to the system than is necessary. This is a case of human error—a common security weakness.
  - Second, an operating system may allow by default more privileges than a normal user needs. This is a case of poor operating-system design decisions.

The Program Threats: Trap Door

- The designer of a program or system might leave a hole in the software that only he / she is capable of using.
- This type of security breach (or **trap door**) was shown in the movie **War Games**.
- Programmers have been arrested for embezzling from banks by including rounding errors in their code and having the occasional half-cent credited to their accounts.
- This account crediting can add up to a large amount of money, considering the number of transactions that a large bank executes
- A clever trap door could be included in a compiler.
- The compiler could generate standard object code as well as a trap door, regardless of the source code being compiled.
- Trap doors pose a difficult problem because, to detect them, we have to analyze all the source code for all components of a system.
- Given that software systems may consist of millions of lines of code, this analysis is not done frequently, and frequently it is not done at all!

The Program Threats: War Games Movie Trailer

- **WarGames** is a **1983** American Cold War science fiction techno-thriller film written by Lawrence Lasker and Walter F. Parkes and directed by John Badham.

- The film stars Matthew Broderick, Dabney Coleman, John Wood, and Ally Sheedy.

- The film was a box-office success, **costing $12 million** and grossing **$79 million**, after five months, in the United States and Canada

The Program Threats: Logic Bomb

- Consider a program that initiates a security incident only under certain circumstances.

- It would be hard to detect because under normal operations, there would be no security hole.

- However, when a predefined set of parameters was met, the security hole would be created. This scenario is known as a **logic bomb**.

- A programmer, for example, might write code to detect whether he/she was still employed; if that check failed, a daemon could be spawned to allow remote access, or code could be launched to cause damage to the site.

The Program Threats: Under Siege 2 Movie Trailer

- **Under Siege 2: Dark Territory** is a **1995** American action thriller film directed by Geoff Murphy, starring Steven Seagal as the ex-Navy SEAL, Casey Ryback

- The title refers to the railroading term that the subject train was travelling through dark territory, a section of railroad track that has no train signals and in which communications between train dispatchers and the railroad engineers were impossible

The Program Threats: Stack and Buffer Overflow

- The attack exploits a bug in a program.
- The bug can be a simple case of poor programming, in which the programmer neglected to code bounds checking on an input field.
- In this case, the attacker sends more data than the program was expecting.
- By using trial and error, or by examining the source code of the attacked program if it is available, the attacker determines the vulnerability and writes a program to do the following
    - Overflow an input field, command-line argument, or input buffer—for example, on a network daemon—until it writes into the stack.
    - Overwrite the current return address on the stack with the address of the exploit code loaded in step 3.
    - Write a simple set of code for the next space in the stack that includes the commands that the attacker wishes to execute —for instance, spawn a shell.

The Program Threats: Stack and Buffer Overflow

- The result of this attack program’s execution will be a root shell or other privileged command execution.

- For instance, if a web-page form expects a username to be entered into a field, the attacker could send the user name, plus extra characters to overflow the buffer and reach the stack, plus a new return address to load onto the stack, plus the code the attacker wants to run.

- When the buffer-reading subroutine returns from execution, the return address is the exploit code, and the code is run.

The Program Threats: Viruses

- This vast variety of viruses has continued to grow.
- For example, in 2004 a new and widespread virus was detected.
- It exploited three separate bugs for its operation.
    - This virus started by infecting hundreds of Windows servers (including many trusted sites) running Microsoft Internet Information Server (IIS).
    - Any vulnerable Microsoft Explorer web browser visiting those sites received a browser virus with any download.
    - The browser virus installed several back-door programs, including a **keystroke logger**, which records everything entered on the keyboard (including passwords and credit-card numbers).
    - It also installed a daemon to allow unlimited remote access by an intruder and another that allowed an intruder to route spam through the infected desktop computer.

The Program Threats: Viruses

- A **Virus** is a fragment of code embedded in a legitimate program.
- Viruses are self-replicating and are designed to “infect” other programs.
- They can wreak havoc in a system by modifying or destroying files and causing system crashes and program malfunctions.
- Viruses are very specific to architectures, operating systems, and applications.
- Viruses are a particular problem for users of PCs.
- UNIX and other multiuser operating systems generally are not susceptible to viruses because the executable programs are protected from writing by the operating system.
- Even if a virus does infect such a program, its powers usually are limited because other aspects of the system are protected.

The Program Threats: Viruses

- Viruses are usually borne via email, with spam the most common vector.

- They can also spread when users download viral programs from Internet file-sharing services or exchange infected disks.

- Once a virus reaches a target machine, a program known as a **Virus Dropper** inserts the virus into the system.

- The virus dropper is usually a Trojan horse, executed for other reasons but installing the virus as its core activity.

- There are literally thousands of viruses, but they fall into several main categories.

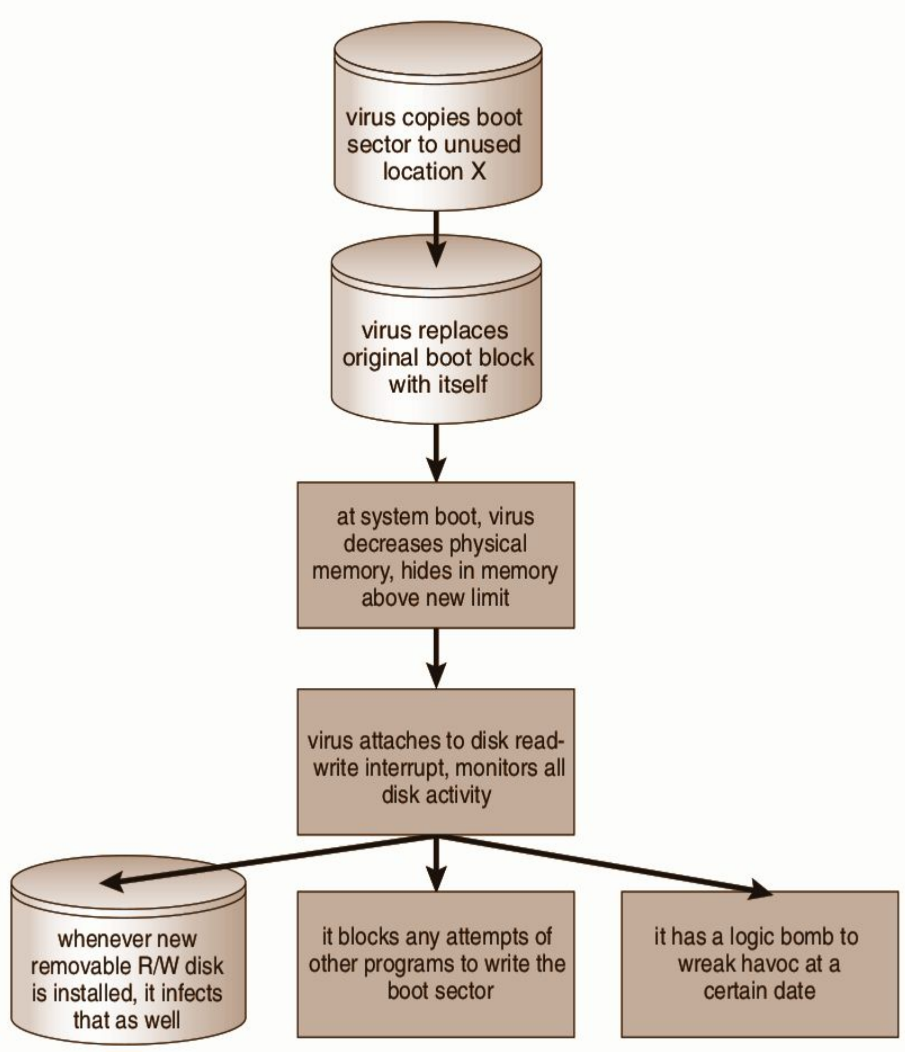
- Note that many viruses belong to more than one category.

The Program Threats: Viruses

- **File:**
  - A standard file virus infects a system by appending itself to a file.
  - It changes the start of the program so that execution jumps to its code.
  - After it executes, it returns control to the program so that its execution is not noticed.
  - File viruses are sometimes known as **Parasitic Viruses**, as they leave no full files behind and leave the host program still functional.

The Program Threats: Viruses

- **Boot:**
  - A boot virus infects the boot sector of the system, executing every time the system is booted and before the operating system is loaded.
  - It watches for other bootable media and infects them.
  - These viruses are also known as **memory viruses**, because they do not appear in the file system.


virus copies boot sector to unused location X
virus replaces original boot block with itself
at system boot, virus decreases physical memory, hides in memory above new limit
virus attaches to disk read-write interrupt, monitors all disk activity
whenever new removable R/W disk is installed, it infects that as well
it blocks any attempts of other programs to write the boot sector
it has a logic bomb to wreak havoc at a certain date

A boot-sector computer virus

The Program Threats: Viruses

- **Macro:**
  - Most viruses are written in a low-level language, such as assembly or C.
  - Macro viruses are written in a high-level language, such as Visual Basic.
  - These viruses are triggered when a program capable of executing the macro is run.
  - For example, a macro virus could be contained in a spreadsheet file.

The Program Threats: Viruses

- **Source code:**
  - A source code virus looks for source code and modifies it to include the virus and to help spread the virus.
- **Polymorphic**:
  - A polymorphic virus changes each time it is installed to avoid detection by antivirus software.
  - The changes do not affect the virus’s functionality but rather change the virus’s signature.
  - A virus signature is a pattern that can be used to identify a virus, typically a series of bytes that make up the virus code.

The Program Threats: Viruses

- **Encrypted:**
  - An encrypted virus includes decryption code along with the encrypted virus, again to avoid detection.
  - The virus first decrypts and then executes.
- **Stealth:**
  - This tricky virus attempts to avoid detection by modifying parts of the system that could be used to detect it. For example, it could modify the read system call so that if the file it has modified is read, the original form of the code is returned rather than the infected code.
- **Tunneling:**
  - This virus attempts to bypass detection by an antivirus scanner by installing itself in the interrupt-handler chain.
  - Similar viruses install themselves in device drivers.

The Program Threats: Viruses

- **Multipartite:**
  - A virus of this type is able to infect multiple parts of a system, including boot sectors, memory, and files.
  - This makes it difficult to detect and contain.
- **Armored:**
  - An armored virus is coded to make it hard for antivirus researchers to unravel and understand.
  - It can also be compressed to avoid detection and disinfection.
  - In addition, virus droppers and other full files that are part of a virus infestation are frequently hidden via file attributes or unviewable file names.

The Program Threats: Viruses

- Viruses are the most disruptive security attacks, and because they are effective, they will continue to be written and to spread.

- An active security-related debate within the computing community concerns the existence of a **monoculture**, in which many systems run the same hardware, operating system, and application software.

- This monoculture supposedly consists of Microsoft products.

- One question is whether such a monoculture even exists today.

- Another question is whether, if it does, it increases the threat of and damage caused by viruses and other security intrusions.

Topic Uncovered Today

- **The Program Threats**

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com