OPERATING SYSTEMS

Input - Output Management and Security - 8

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

OPERATING SYSTEMS

System and Network Threats

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

OPERATING SYSTEMS

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 System Threats
- The Network Threats

The System and Network Threats

- Program threats typically use a breakdown in the protection mechanisms of a system to attack other programs.
- In contrast, **System** and **Network Threats** involve the abuse of services and network connections.
- System and network threats create a situation in which operating-system resources and user files are misused
- A system and network attack is used to launch a program attack, and vice versa
- Operating Systems strive to be secure by default

The System and Network Threats

- Changes in policy and mechanisms reduce the system's attack surface the set of ways in which an attacker can try to break into the system.
- It is important to note that masquerading and replay attacks are also commonly launched over networks between systems.
- These attacks are more effective and harder to counter when multiple systems are involved. For example, within a computer, the operating system usually can determine the sender and receiver of a message.
- Even if the sender changes to the ID of someone else, there may be a record of that ID change.
- When multiple systems are involved, especially systems controlled by attackers, then such tracing is much more difficult

The System and Network Threats

- Sharing secrets to prove identity and as keys to encryption is required for authentication and encryption, and sharing secrets is easier in environments such as a single operating system in which secure sharing methods exist.

- These methods include shared memory and interprocess communications.

The System and Network Threats: Worms

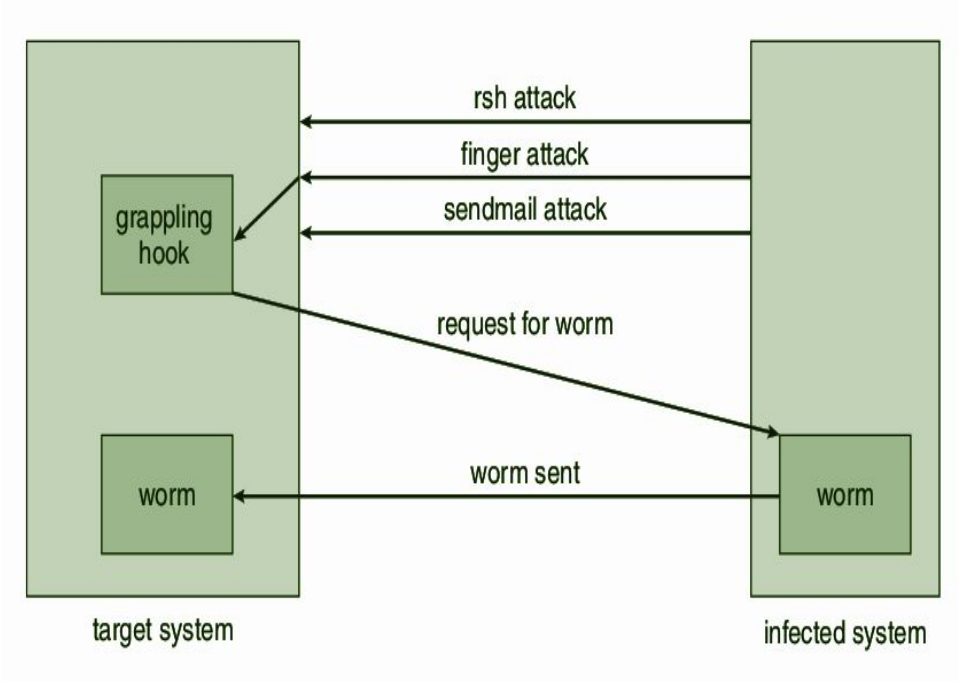
- A **worm** is a process that uses the spawn mechanism to duplicate itself.
- The worm spawns copies of itself, using up system resources and perhaps locking out all other processes.
- An event occurred in 1988 to UNIX systems on the Internet, causing the loss of system and system-administrator time worth millions of dollars
- At the close of the workday on November 2, 1988, Robert Tappan Morris, Jr., a first-year Cornell graduate student, unleashed a worm program on one or more hosts connected to the Internet.
- Targeting Sun Microsystems' Sun 3 workstations and VAX computers running variants of Version 4 BSD UNIX , the worm quickly spread over great distances.
- Within a few hours of its release, it had consumed system resources to the point of bringing down the infected machines.

The System and Network Threats: Worms

- The worm was made up of two programs, a grappling hook also called a bootstrap or vector program and the main program.

- Once established on the computer system under attack, the grappling hook connected to the machine where it originated and uploaded a copy of the main worm onto the hooked system

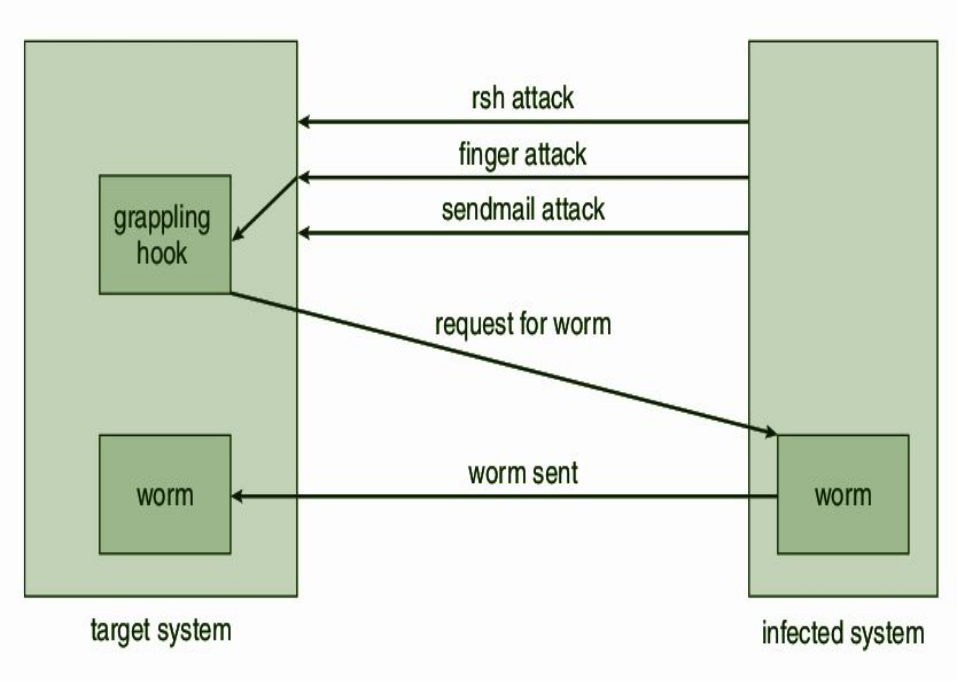
- The main program proceeded to search for other machines to which the newly infected system could connect



The Morris Internet worm.

The System and Network Threats: Worms

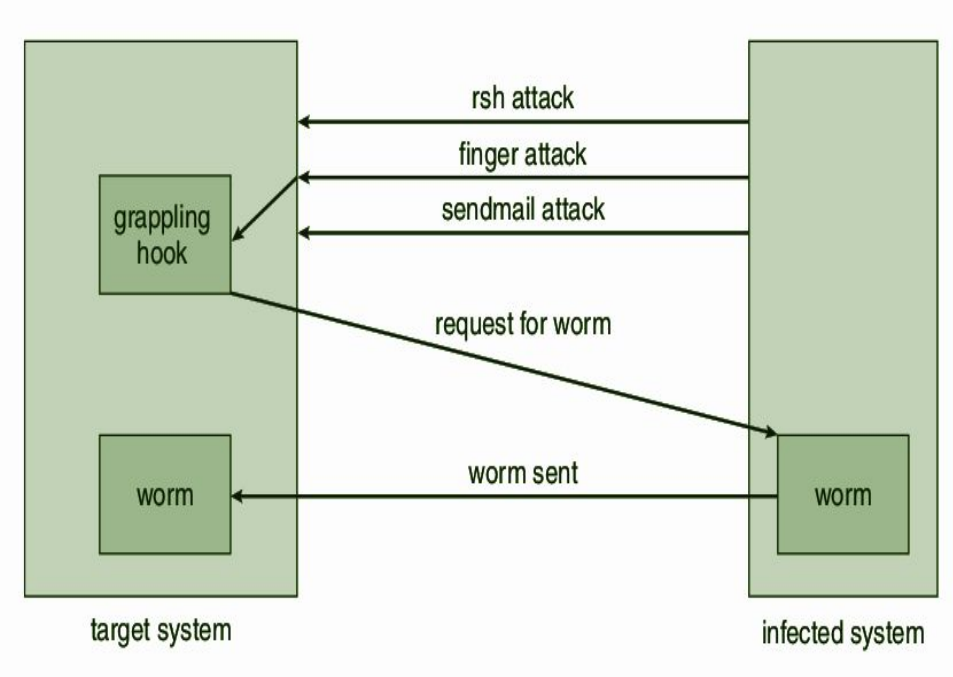
- In these actions, Morris exploited the UNIX networking utility rsh for easy remote task execution.
- By setting up special files that list host–login name pairs, users can omit entering a password each time they access a remote account on the paired list
- The worm searched these special files for site names that would allow remote execution without a password.
- Where remote shells were established, the worm program was uploaded and began executing as new.



The Morris Internet worm.

The System and Network Threats: Worms

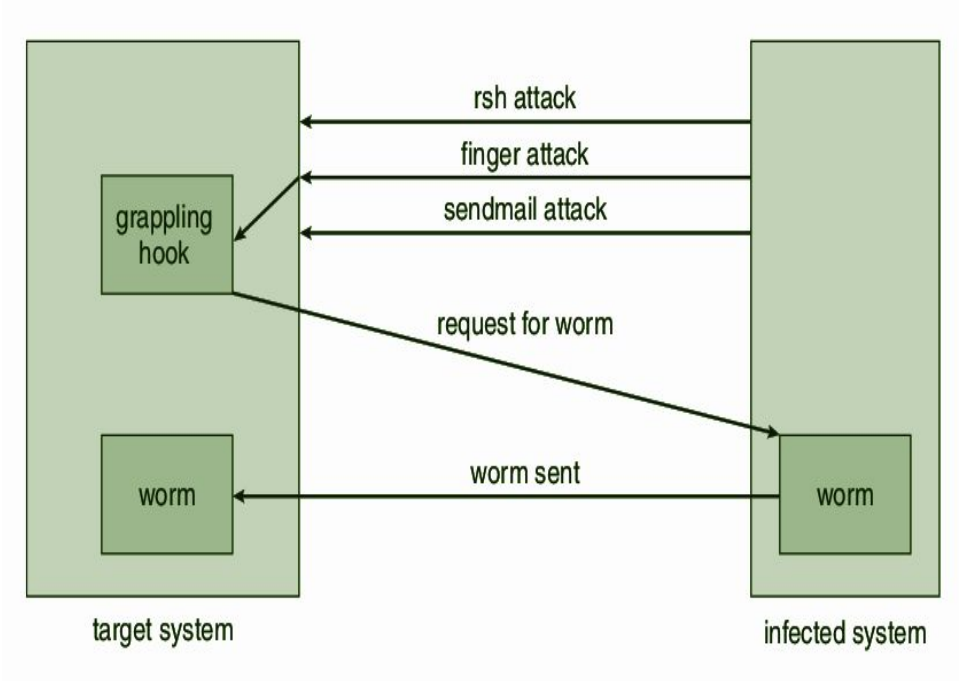
- The finger utility functions as an electronic telephone directory.
- The command finger user-name@hostname returns a person's real and login names along with other information that the user may have provided, such as office and home address and telephone number, research plan, or clever quotation.
- Finger runs as a background process (or daemon) at each BSD site and responds to queries throughout the Internet.



The Morris Internet worm.

The System and Network Threats: Worms

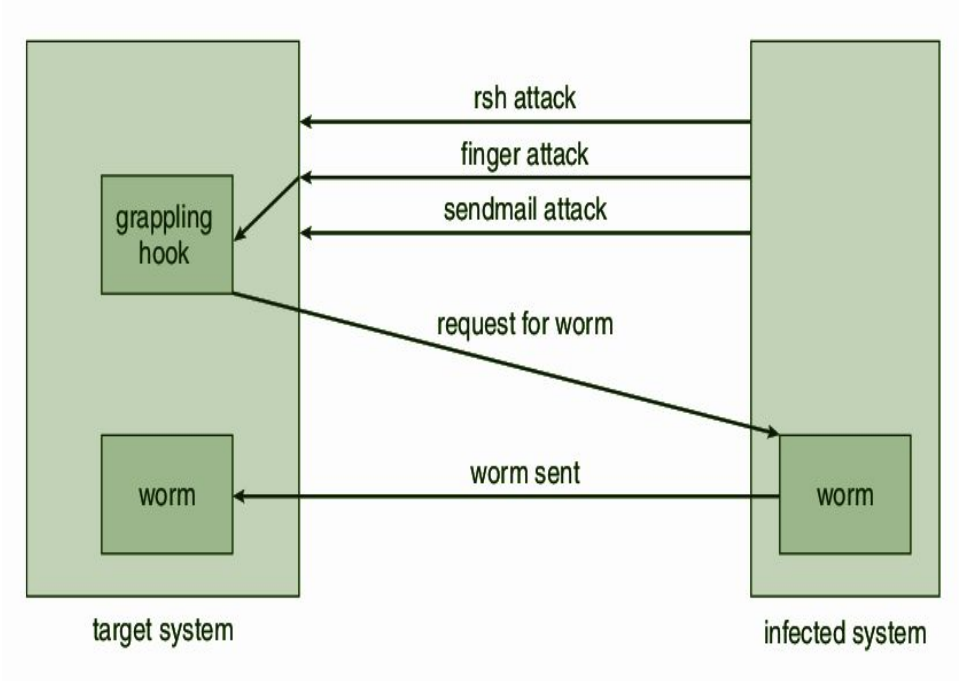
- The program queried finger with a 536-byte string crafted to exceed the buffer allocated for input and to overwrite the stack frame.
- Instead of returning to the main routine where it resided before Morris's call, the finger daemon was routed to a procedure within the invading 536-byte string now residing on the stack.
- The new procedure executed /bin/sh , which, if successful, gave the worm a remote shell on the machine under attack
- The bug exploited in sendmail also involved using a daemon process for malicious entry. sendmail sends, receives, and routes electronic mail.



The Morris Internet worm.

The System and Network Threats: Worms

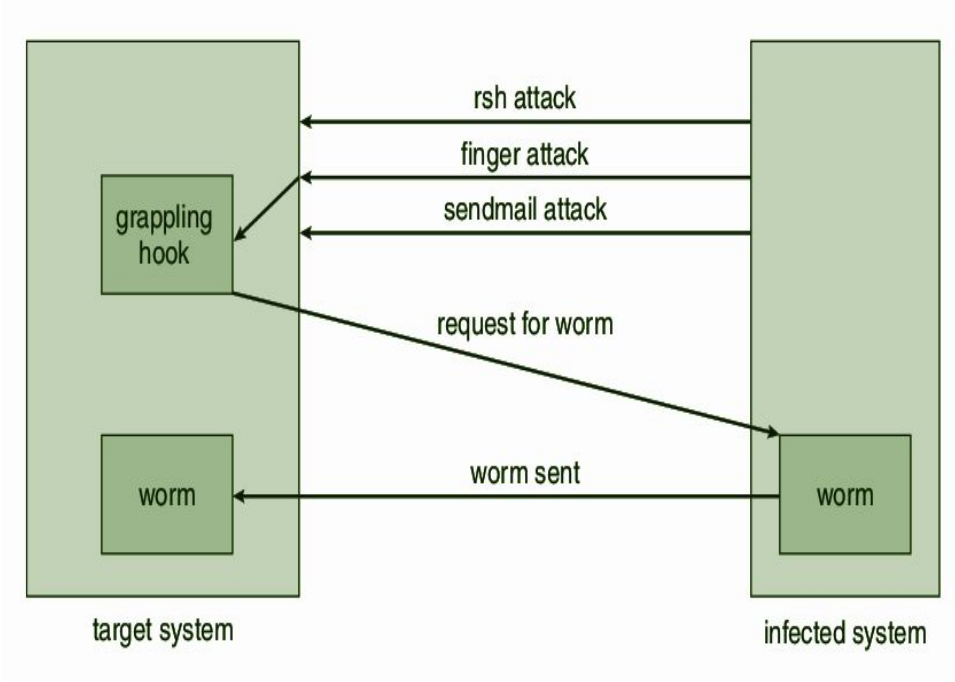
- Once in place, the main worm systematically attempted to discover user passwords.
- It began by trying simple cases of no password or passwords constructed of account–username combinations, then used comparisons with an internal dictionary of 432 favorite password choices, and then went to the final stage of trying each word in the standard UNIX online dictionary as a possible password.
- The very features of the UNIX network environment that assisted in the worm's propagation also helped to stop its advance.



The Morris Internet worm.

The System and Network Threats: Worms

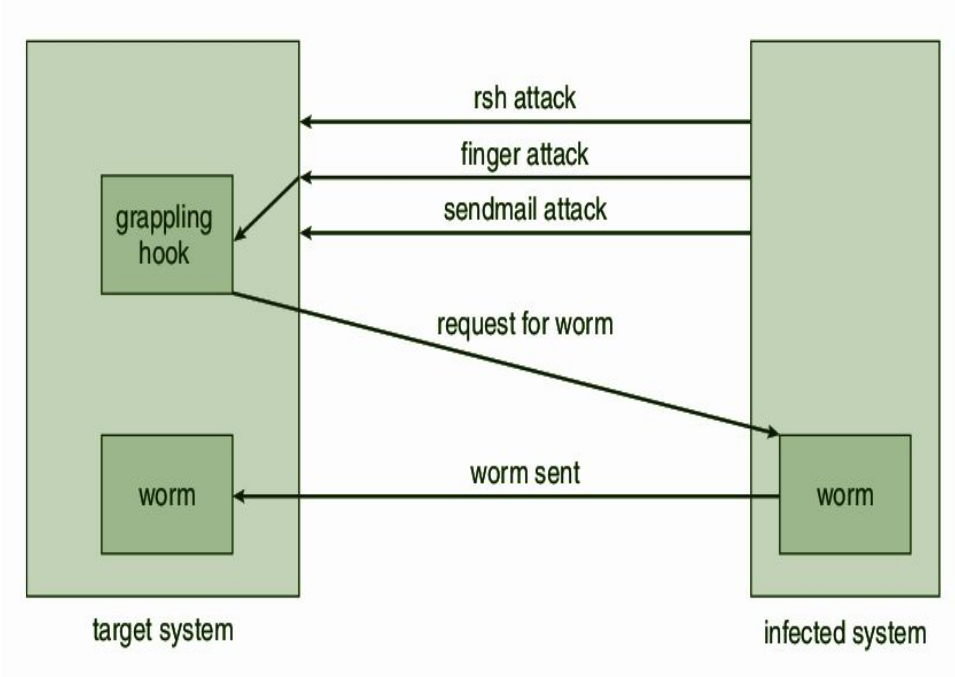
- By the evening of the next day, November 3, methods of halting the invading program were circulated to system administrators via the Internet.
- Within days, specific software patches for the exploited security flaws were available.
- The action has been characterized as both a harmless prank gone awry and a serious criminal offense.
- Based on the complexity of the attack, it is unlikely that the worm's release or the scope of its spread was unintentional.
- The worm program took elaborate steps to cover its tracks and to repel efforts to stop its spread.



The Morris Internet worm.

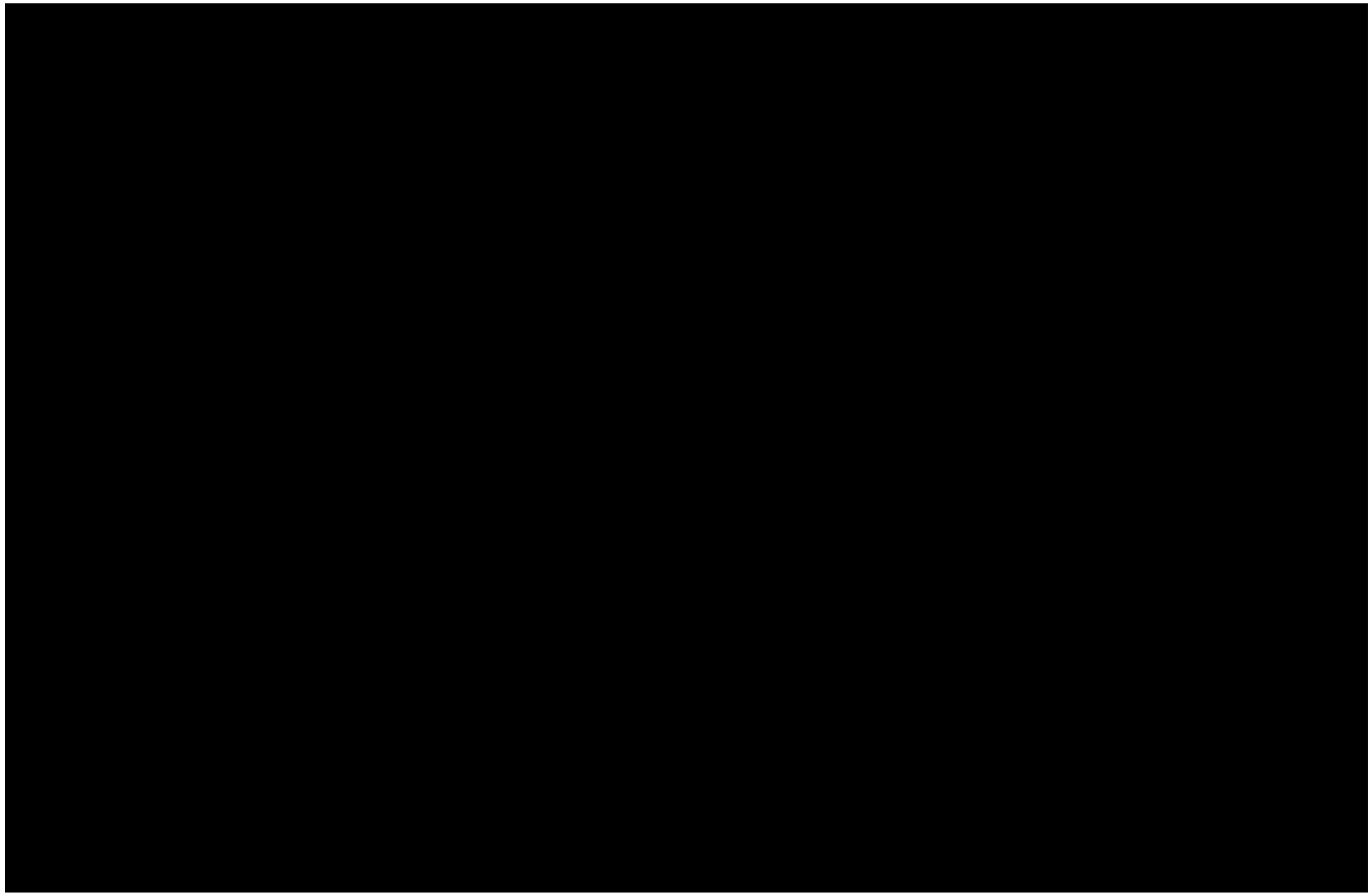
The System and Network Threats: Worms

- What is not open to speculation, however, is the legal outcome: a federal court convicted Morris and handed down a sentence of three years' probation, 400 hours of community service, and a **$10,000 fine**. Morris's legal costs probably exceeded **$100,000**



The Morris Internet worm.

The World's First Cyber Crime: The Morris Worm [KERNEL PANIC]



The System and Network Threats: Port Scanning

- Port scanning is not an attack but rather a means for a cracker to detect a system's vulnerabilities to attack.
- Port scanning typically is automated, involving a tool that attempts to create a TCP/IP connection to a specific port or a range of ports.
- A cracker could launch a port scanner to try to connect, say, to port 25 of a particular system or to a range of systems.
- If the connection was successful, the cracker (or tool) could attempt to communicate with the answering service to determine if the service was indeed sendmail and, if so,if it was the version with the bug
- A tool in which each bug of every service of every operating system was encoded.
- The tool could attempt to connect to every port of one or more systems.

The System and Network Threats: Port Scanning

- Frequently, the bugs are buffer overflows, allowing the creation of a privileged command shell on the system.
- From there, of course, the cracker could install Trojan horses, back-door programs, and so on
- Because port scans are detectable, they frequently are launched from **zombie systems**.
- **Zombies** make crackers particularly difficult to prosecute because determining the source of the attack and the person that launched it is challenging

The System and Network Threats: Denial of Service

- Denial-of-service attacks are aimed not at gaining information or stealing resources but rather at disrupting legitimate use of a system or facility.
- Launching an attack that prevents legitimate use is frequently easier than breaking into a machine or facility
- Denial-of-service attacks are generally network based, which typically fall into two categories
- Attacks in the **first category** use so many facility resources that, in essence, no useful work can be done.
- The **second category** involves disrupting the network of the facility.

The System and Network Threats: Denial of Service

- The attacks are usually stopped at the network level until the operating systems can be updated to reduce their vulnerability
- Generally, it is impossible to prevent denial-of-service attacks.
- The attacks use the same mechanisms as normal operation.
- Even more difficult to prevent and resolve are distributed denial-of-service (DDOS) attacks
- These attacks are launched from multiple sites at once, toward a common target, typically by zombies.
- DDOS attacks have become more common and are sometimes associated with blackmail attempts.
- A site comes under attack, and the attackers offer to halt the attack in exchange for money.

The System and Network Threats: Denial of Service

- There are other interesting aspects of DOS attacks.
- For example, if an authentication algorithm locks an account for a period of time after several incorrect attempts to access the account, then an attacker could cause all authentication to be blocked by purposely making incorrect attempts to access all accounts.
- Similarly, a firewall that automatically blocks certain kinds of traffic could be induced to block that traffic when it should not.
- These examples suggest that programmers and systems managers need to fully understand the algorithms and technologies they are deploying.
- Finally, computer science classes are notorious sources of accidental system DOS attacks.
- Consider the first programming exercises in which students learn to create sub processes or threads.
- A common bug involves spawning subprocesses infinitely.
- The system's free memory and CPU resources don't stand a chance.

Topic Outline

- The System Threats
- The Network 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