Lesson 5 – Hardware architecture for security & Malicious Code

Risk matrix review

- It is a tool to manage security risks
  - What could happen
    - Brainstorm potential incidents
  - How likely it is to happen
    - How often we expect those incidents
  - What to do when it happen
    - Plan of action

Managing risk with a “risk matrix” -- Review

1. Identify possible incidents and consequences
   - Brainstorm what incidents could happen and how serious are those incidents
   - Use a risk impact matrix
2. Identify likelihood of the events
   - Use a likelihood score matrix
3. Use a 5x5 scoring matrix to calculate risk
4. Identify the appropriate action based on the risk
   - Describe what to do when an incident happen
   - Use a mitigation/action matrix

Risk impact matrix

- Risk impact matrixes are easy to understand
- Columns represent severity of a security incident
- Rows represent the affected area

	1	2	3	4	5
Descriptor	**Insignificant**	**Minor**	**Moderate**	**Major**	**Catastrophic**
Objectives/Projects	Insignificant cost increase/schedule slippage. Barely noticeable reduction in scope or quality	<5% over budget/ schedule slippage. Minor reduction in quality/scope	5-10% over budget/schedule slippage. Reduction in scope or quality.	10-25% over budget/schedule slippage. Failure to meet secondary objectives.	>25% over budget/ schedule slippage. Doesn't meet primary objectives.
Injury (Physical/Psychological)	No apparent injury or minor injury not requiring first aid.	Minor injury or illness requiring first aid treatment.	RIDDOR/NPSA reportable	Major injury, or long term incapacity/ disability (loss of limb)	Death or major permanent incapacity
Patient Experience/Outcome	Unsatisfactory patient experience not directly related to patient care	Unsatisfactory patient experience – readily resolved	Mismanagement of patient care, short term effects (less than a week)	Serious mismanagement of patient care, long term effects (more than a week)	Totally unsatisfactory patient outcome or experience

Likelihood Score

- Likelihood Scores indicates how often an incident is expected
- Should be based on probability or frequency

Likelihood Score	Description
1 - Rare	The event may never occur. The event is not expected for years.
2 - Unlikely	The event is not expected, but it may happen. Expected at least yearly.
3 - Possible	The event will occasionally happen. Expected at least every semester.
4 - Likely	The event will probably happen. Expected at least monthly.
5 - Almost certain	The event will certainly happen. Expected at least weekly.

5x5 Risk scoring matrix

- Columns correspond to risk-impact scores
- Rows correspond to likelihood scores
- The risk score is calculated by multiplying the column impact score with the row likelihood score

		Rare	Unlikely	Possible	Likely	Certain
		1	2	3	4	5
Insignificant	1	1	2	3	4	5
Minor	2	2	4	6	8	10
Moderate	3	3	6	9	12	15
Major	4	4	8	12	16	20
Catastrophic	5	5	10	15	20	25

Mitigation/Action matrix

- Describes what to do when a security incident occurs
- Scores come from the 5x5 risk scoring matrix
- Risk is a categorization of the scores

Score	Risk	Actions	Reporting Requirements
1-3	Low	Where appropriate carry out local investigation for low risk incidents. Local managers would be expected to monitor trends associated with this grade of incident and identify where causal factors are generic to the service/area and take appropriate action to address any local systems failures. Any identified low level risks should be managed through existing control measures and assessments kept under review.	Report to manager/supervisor of the area in which the accident/incident occurred or where the risk has been identified.
4-6	Moderate	Investigate to determine underlying causes. Where appropriate review any existing risk assessment and consider the effectiveness of the controls. Implement any further treatment plans as required.	Report to manager/supervisor of the area in which the accident/incident occurred or where the risk has been identified.
8-12	High	**Make safe the situation**. Carry out a full RCA. Action plans to be developed, implemented and monitored. Where a high level risk is identified review effectiveness of existing controls. If adequate control cannot be implemented immediately, an action plan must be developed to indicate how the risk will be reduced, who will be responsible for implementation and the time-scale involved.	**Report immediately** to the appropriate Director/Senior Manager. (See Incident Reporting Policy) Enter onto Corporate Risk Register
15-25	Extreme	**STOP activity and make safe the situation.** Immediate action must be taken to either eliminate or adequately control the risk before any further activity is undertaken. Carry out a full RCA and where appropriate develop, implement and monitor further action plans.	**Report immediately** to the appropriate Director/Senior Manager. (See Incident Reporting Policy) Report to Trust Board. Enter onto Corporate Risk Register

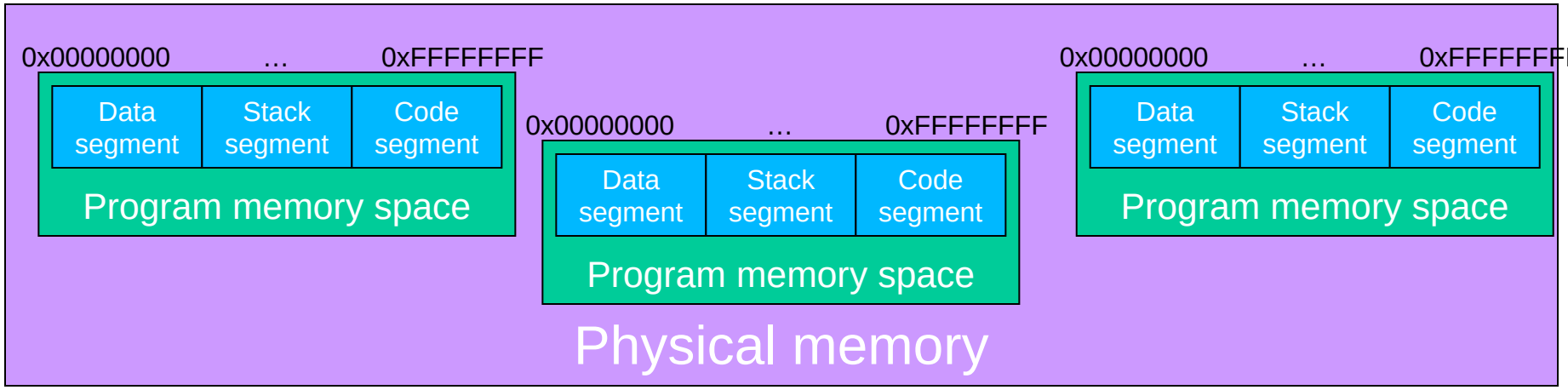
Hardware architecture for security

Operating System (OS) definitions

- Users:
  - Are materialized in the system by programs executed on their behalf.
- Data:
  - Information or instructions encoded in memory. Data should remain related to their owners via a pointer.
- Programs:
  - Are actually data, and should be considers as such until they are fed into memory for execution.

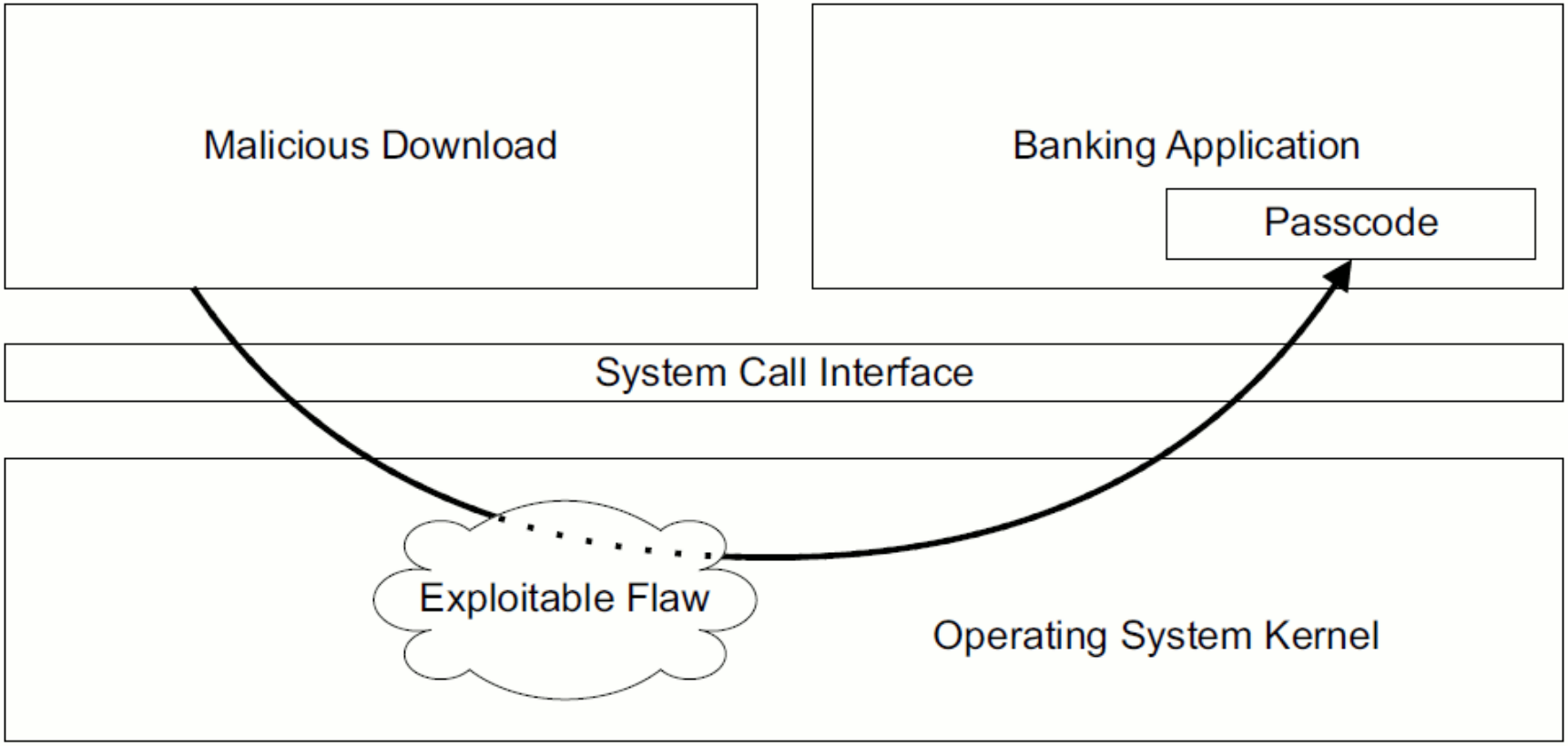
Program memory isolation

- Modern architectures
  - Map program memory space to the physical memory space
  - Protect a process memory space from being accessed by another process
    - A pointer from one process cannot access outside its memory space



Cross application memory access vulnerability

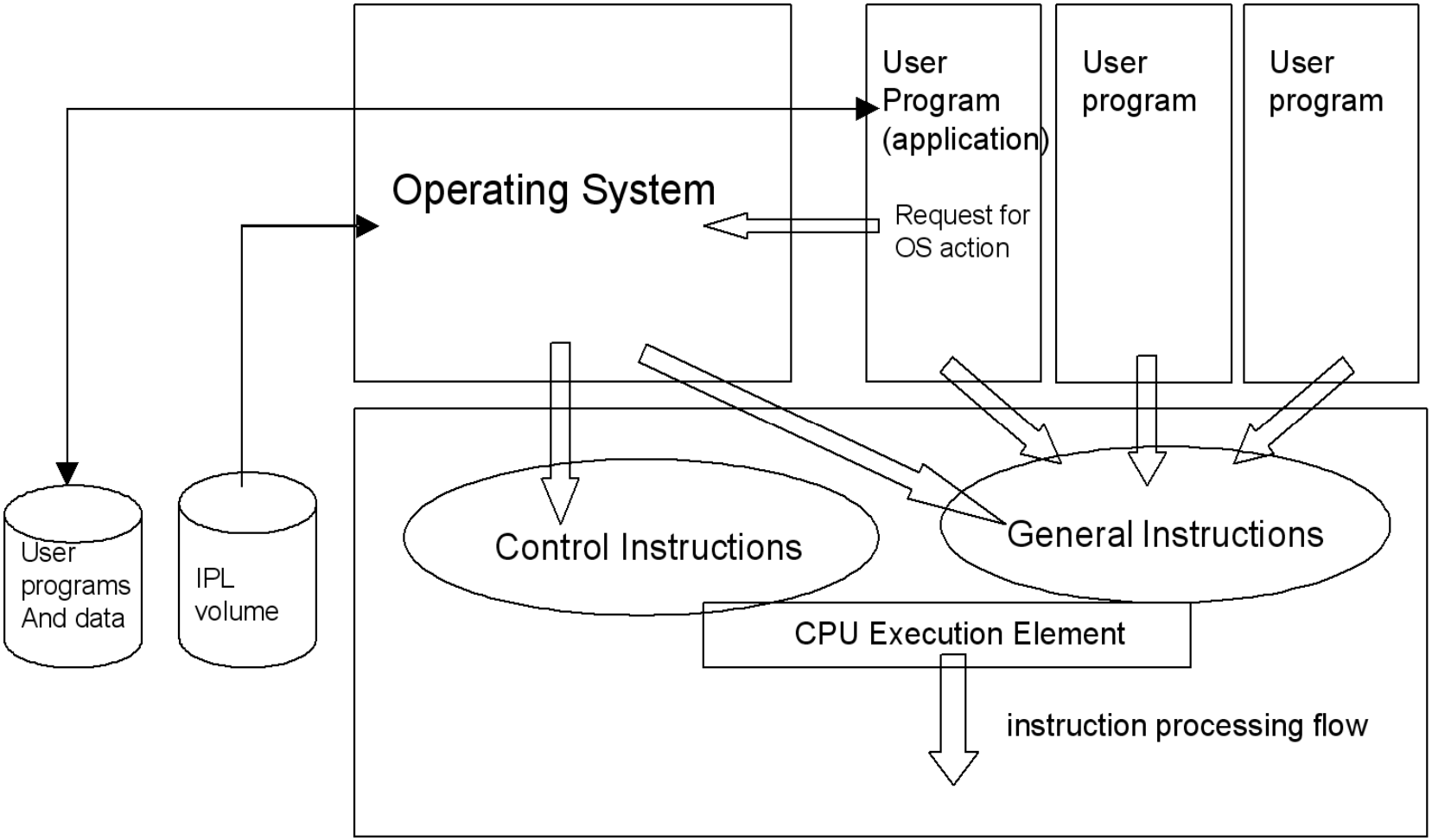
- Exploiting a system flaw to access other process memory space



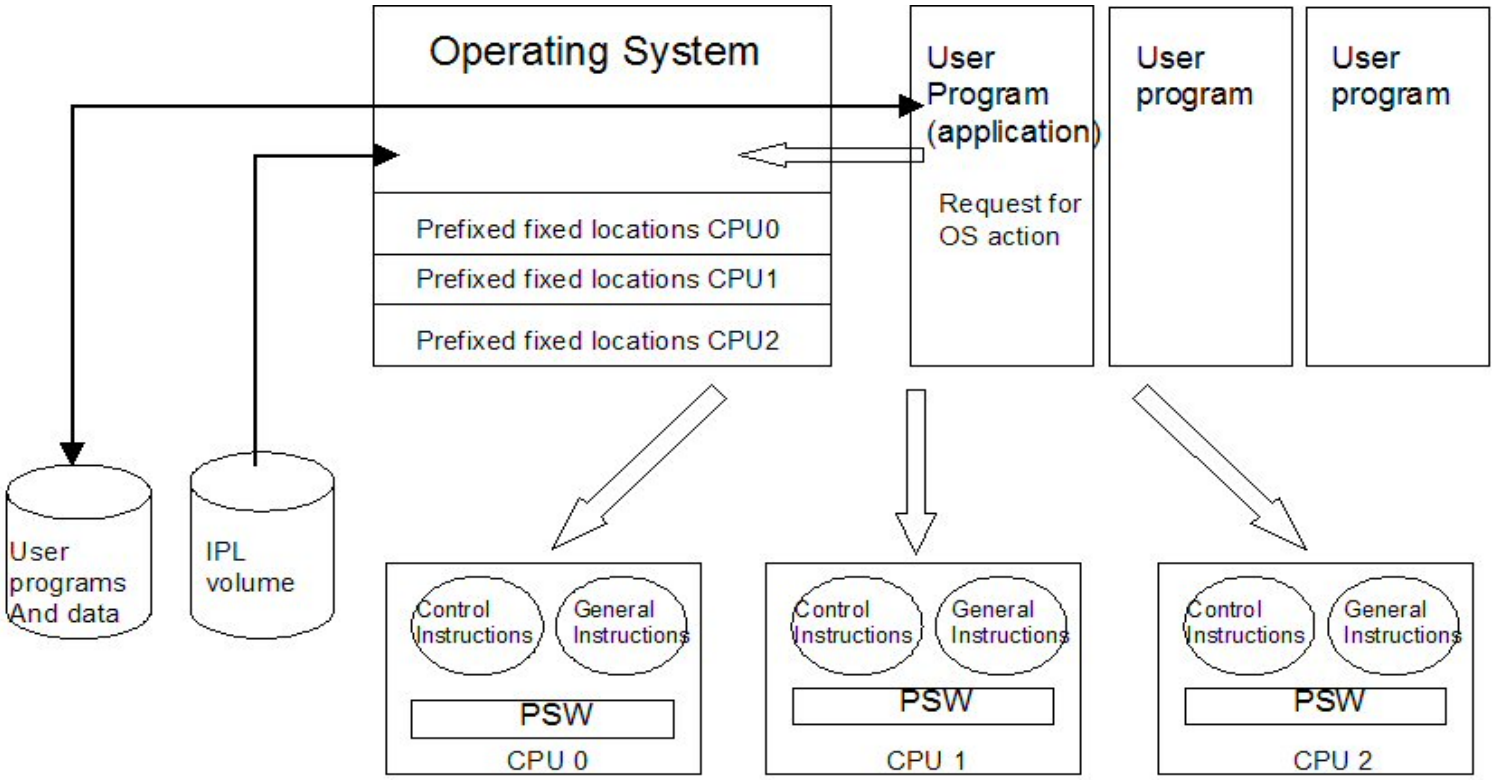
Hardware instructions

- User mode or general instructions:
  - Instructions that can be executed by any program.
- Kernel mode or control instructions:
  - Have the capability of affecting the user execution environment.
  - Should be made available to the OS only

zOS Instruction Execution

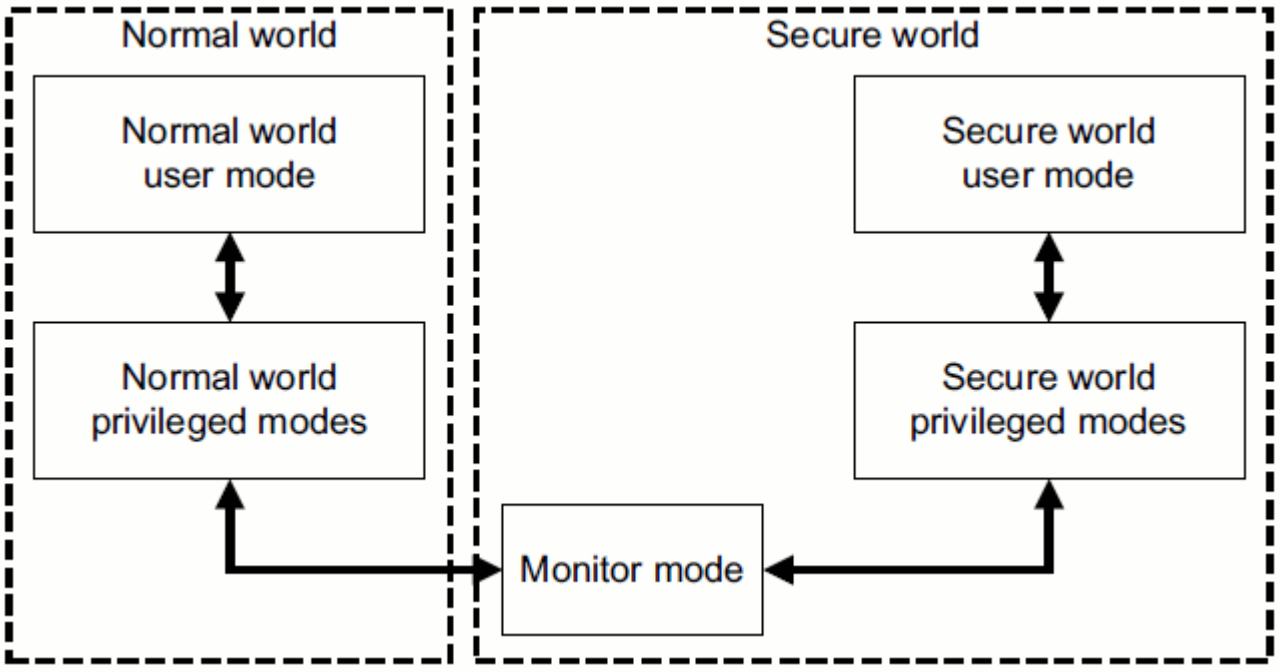


Multiprocessing zOS architecture



ARM processor TrustZone hardware architecture

- Two cores
  - Non-secure core (normal world) can only access non-secure system resources
  - Secured core (secured world) can access all the resources
- Entering monitor mode is tightly controlled



Summary

- Hardware security protection is achieved by
  - Memory isolation
  - Segregating user mode and kernel mode instructions
    - Only secure programs (operating system) get to execute kernel mode instruction
  - Segregation of system resources into non-secured and secured
- In general, hardware security is based on preventing malicious code from accessing unauthorized resources

Malicious Code (Malware)

Some material was adapted from: Mihai Christodorescu, Malicious Code for Fun and Profit, 3/29/2007

Malicious code = Malware

- Malicious software, designed to secretly access a computer system without the owner's informed consent
- Code that breaks the security policies of the victim

Malware characteristics

- Delivers a payload
  - How the malware affects its target
- Uses an attack vector
  - How the malware infects or spread to its targets
- May use a replicating algorithm
  - How the malware makes copies of itself

Payload

- The function or malicious action taken by the malware
- Examples
  - Displaying a message in the screen
  - Erasing files
  - Damaging the boot sector
  - Infecting other programs or data files

Attack Vectors

- Social engineering
  - “Make them want to run it”
- Vulnerability exploitation
  - “Force your way into the system”
- Piggybacking
  - “Make it run when other programs run”

Social engineering

- Trick the user to execute the program
- Examples:
  - Executable email attachment
  - Make it look like a game, movie, important document, etc.

Vulnerability exploitation

- Exploit software design flaws
- Examples
  - Buffer overflow
  - SQL injection

Piggybacking

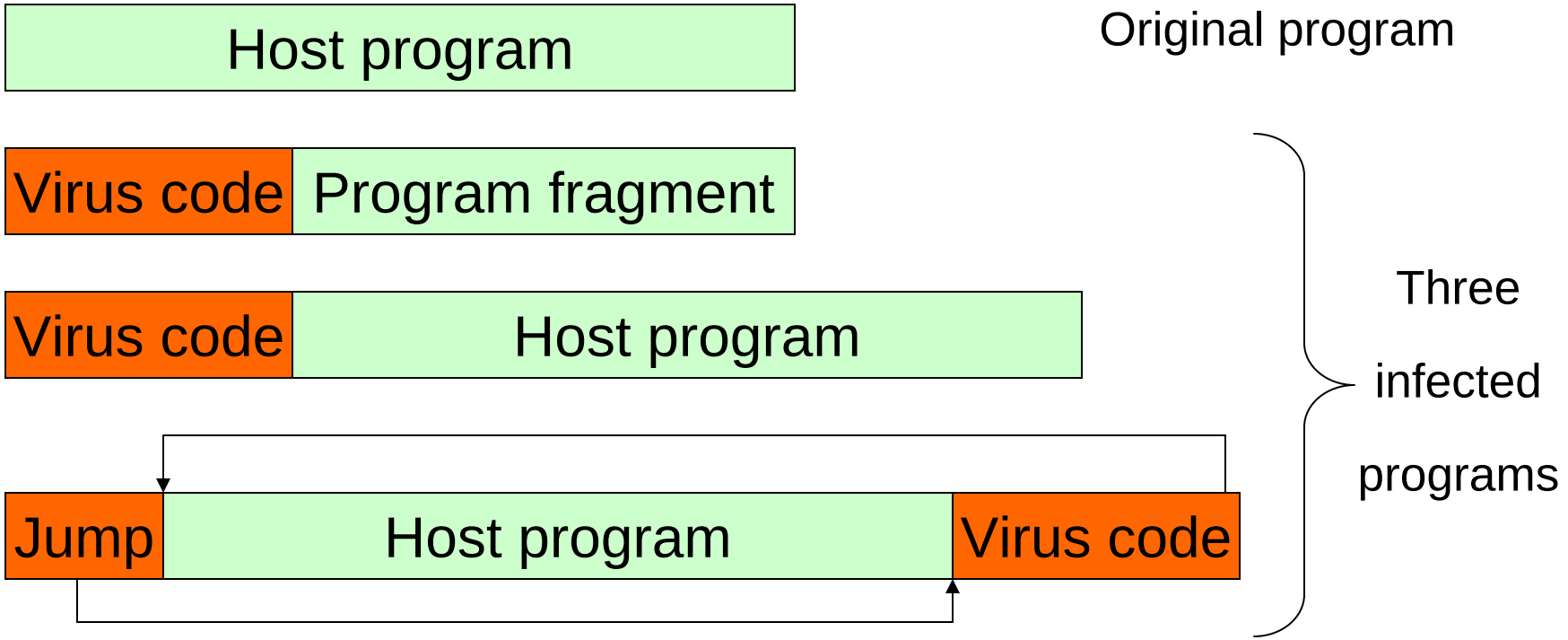
- Malicious code inserted into a program or data file
- Examples
  - infecting an executable file
  - Inserting a malicious macro into a document or spreadsheet

Replicating algorithm

- Not all malware replicates
- The malware that replicates depend on the attack vector
  - Email based social engineering
    - need email addresses
  - Vulnerability-based
    - need IP addresses of hosts running the vulnerable service
  - Piggybacking
    - need more files to infect

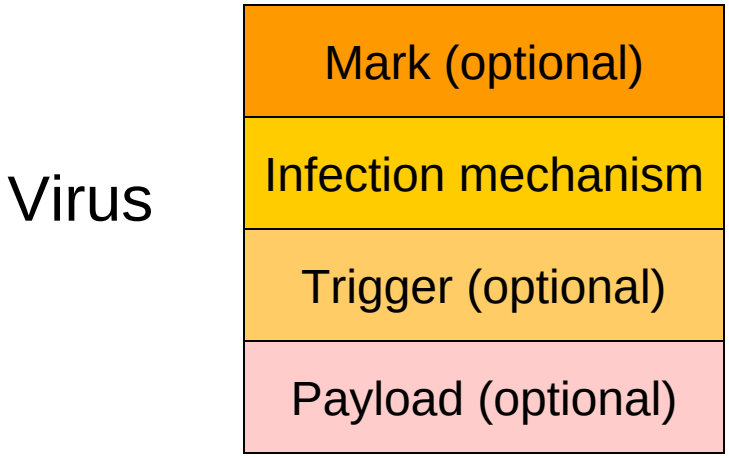
Virus

- Attaches itself to a program and executes secretly when the host is run



Anatomy of a virus

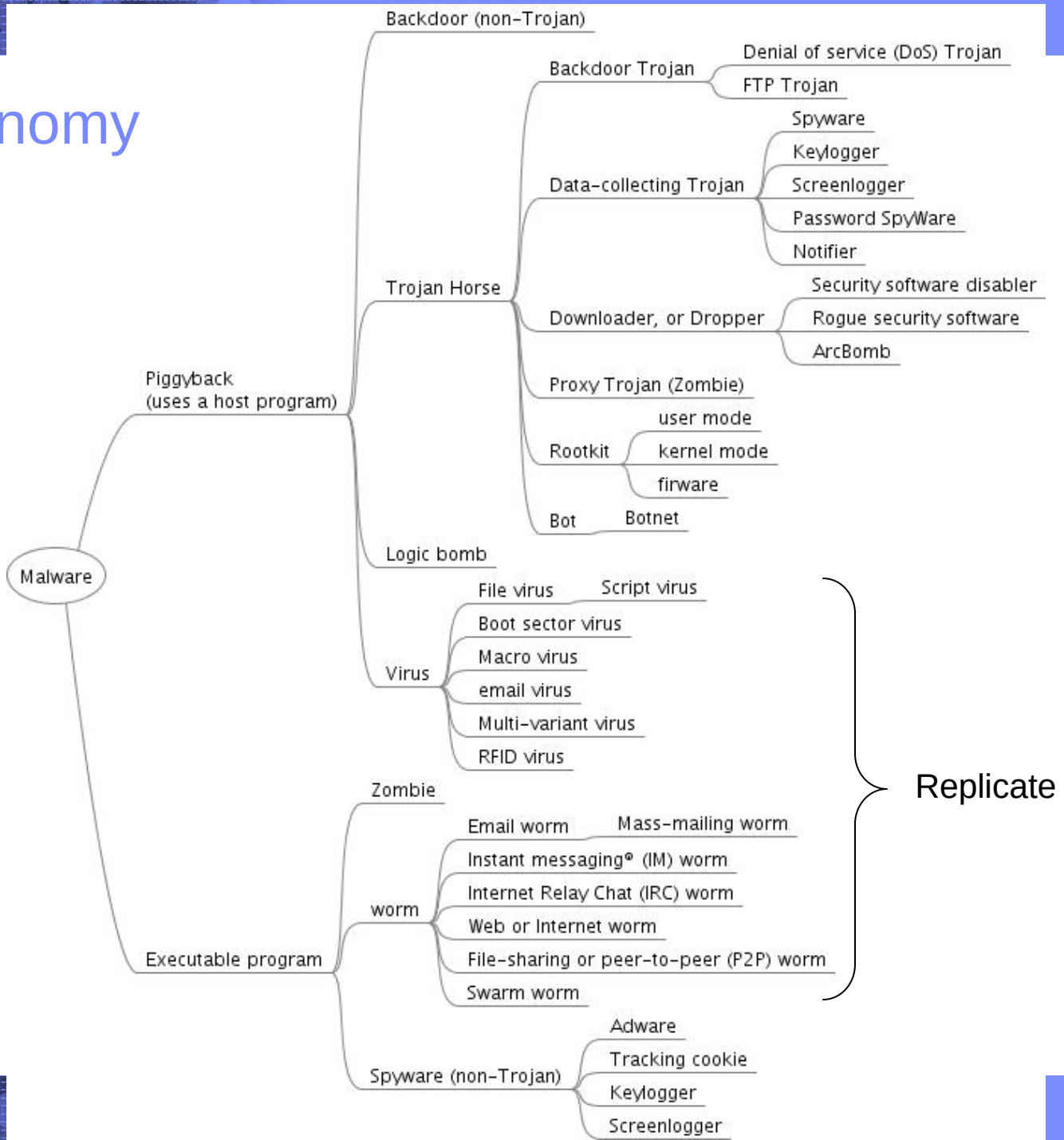
- A virus may have four components
  - Mark to prevent re-infection attempts (optional)
  - Infection mechanism to spread to other files
  - Trigger or condition to deliver the payload (optional)
  - Payload is the malicious function (optional)



Virus lifecycle

- Most virus will evolve in four phases
  - Dormant phase (optional phase)
    - Idle waiting for an event to wakeup
  - Propagation phase
    - Replicate itself into other programs or system structures in disk
  - Triggering phase
    - Waits for an event to move into the execution phase
  - Execution phase
    - Executes the function it was designed to do

Malware taxonomy



Types of malicious code

- Backdoor
  - Allows unauthorized access to a system or application
- Rootkit
  - Allow remote access to a computer
  - Hides itself from detection
- Logic bomb
  - Trigger a malicious function when a specified condition is meet
  - May be part of a virus, worm, or Trojan horse

Types of malicious code

- Trojan horse
  - Appears to execute an useful function
- Adware
  - Provide (play, display, or download) advertisement to the computer
- Spyware
  - Collects information about the user without his or her knowledge
- Virus
  - Self-replicating, infects program or document
- Worm
  - Self-replicating, spreads across the network

Types of malicious code

- Zombie computer
  - Self-replicating, performs controlled remote attacks
  - Commonly used to send email spam
    - It is cheaper for the spammers to use other computers, email accounts, and bandwidth

The End