جامـــــعـة المـــــسـتـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الأمن السيبراني

Lecture: (5+6)

Threat Model , Attack Vectors

Subject: software security

second Stage

Lecturer: Asst. Lecturer. Suha Alhussieny

Threat Model

A Threat Model in software security is a structured approach used to identify, evaluate, and address potential threats that could harm a software system. The purpose of threat modeling is to understand the security risks to a system, prioritize those risks, and develop strategies to mitigate them. This process is crucial in building secure software, as it helps in anticipating and countering potential attacks before they occur.

Threat modeling is the process of systematically identifying security threats and vulnerabilities, assessing their potential impact, and planning mitigations to protect the system.

The primary goal is to improve the security posture of a system by proactively identifying and addressing potential threats, ensuring that security is integrated throughout the software development lifecycle. Key Components of Threat Modeling

- **Assets:** The valuable components of the system that need protection, such as data.
- **Threats:** Potential actions or events that could compromise the confidentiality, integrity, or availability of assets.
- **Vulnerabilities:** Weaknesses or flaws in the system that could be exploited by a threat to cause harm.
- **Attack Vectors:** The paths or methods that attackers use to exploit vulnerabilities and carry out threats.

- **Mitigations:** The security controls or countermeasures implemented to reduce or eliminate the risks posed by threats.

Bug versus Vulnerability

In software security, "bug" and "vulnerability" are terms that refer to different aspects of software flaws. While they are related, they have distinct meanings and implications.

1. Bug

- **Definition:** A bug is a flaw, mistake, or unintended behavior in the software's code that causes the software to operate incorrectly or produce unexpected results. Bugs can arise from errors in logic, incorrect assumptions, or programming mistakes.
- **Scope:** Bugs can affect the functionality, performance, or usability of software. They are not necessarily security-related and may not have any impact on the security of the system.
- **Examples:**

O A calculation error in a financial application that leads to incorrect results.

O A typo in a user interface string that displays incorrect information to the user.

O A crash in a software program due to improper memory management.

2. Vulnerability

- **Definition:** A vulnerability is a specific type of bug or flaw in software that can be exploited by an attacker to compromise the security of the system. Vulnerabilities create opportunities for unauthorized access, data breaches, or other malicious activities.

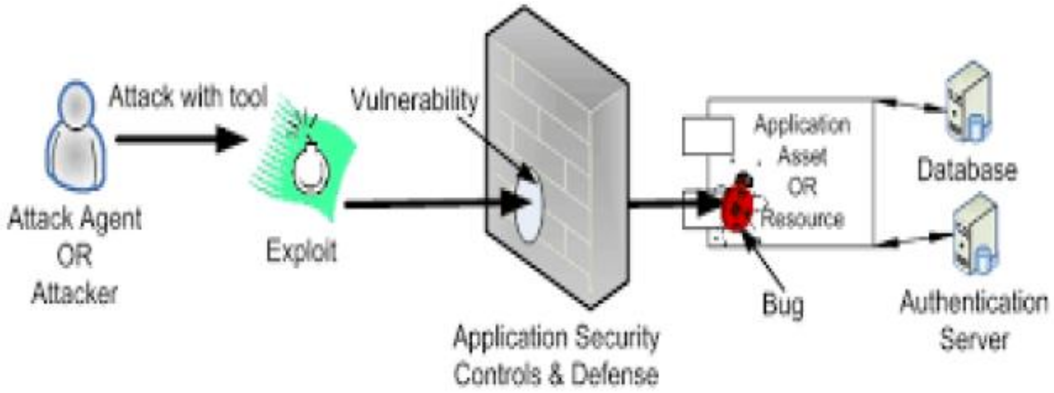
- **Scope:** Vulnerabilities directly impact the confidentiality, integrity, or availability of a system. They can be exploited to perform actions that should not be possible, such as gaining unauthorized access, executing arbitrary code, or causing a denial of service.

- **Examples:**

  O A buffer overflow vulnerability that allows an attacker to execute arbitrary code on a system.

  O An SQL injection vulnerability that enables an attacker to manipulate a database.

  O An authentication bypass vulnerability that lets an attacker access a system without proper credentials.



Section two: Attack Vectors

An attack vector, or threat vector, is a way for attackers to enter a network or system. Common attack vectors include social engineering attacks, credential theft, vulnerability exploits, and insufficient protection against insider threats.

Suppose a security firm is tasked with guarding a rare painting that hangs in a museum. There are several ways that a thief could enter and exit the museum — front doors, back doors, elevators, and windows. A thief could enter the museum in some other way too, perhaps by posing as a member of the museum's staff. All of these methods represent attack vectors, and the security firm may try to eliminate them by placing security guards at all doors, putting locks on windows, and regularly screening museum staff to confirm their identity.

Similarly, digital systems all have areas attackers can use as entry points. Because modern computing systems and application environments are so complex, closing off all attack vectors is typically not possible. But strong security practices and safeguards can eliminate most attack vectors, making it far more difficult for attackers to find and use them.

H.W

Q1/ what is Bug , Vulnerability ?

Q2/

Threat Model Questions

1. **What is the primary goal of threat modeling in software security?**

○ A. To identify all bugs in the software

○ B. To improve the security posture of a system by proactively identifying threats

○ C. To increase the performance of the software

○ D. To reduce the cost of software development

○ E. To eliminate all vulnerabilities in the system

2. **Which of the following is NOT a key component of threat modeling?**

○ A. Assets

○ B. Threats

○ C. Vulnerabilities

○D. Attack Vectors

○E. Debugging Tools

3. **What are "assets" in the context of threat modeling?**

○A. The tools used by developers

○B. The valuable components of the system that need protection

○C. The bugs in the software

○D. The methods used by attackers

○E. The programming languages used in development

4. **Which of the following best describes a "vulnerability"?**

○A. A flaw in the software that causes it to crash

○B. A weakness in the system that can be exploited by a threat

○C. A feature that improves system performance

○D. A tool used to detect bugs

○E. A type of attack vector

5. **What is the purpose of "mitigations" in threat modeling?**

○A. To increase the complexity of the system

○B. To reduce or eliminate the risks posed by threats

○C. To introduce new vulnerabilities

○D. To improve the user interface

○E. To speed up the software development process

Bug vs. Vulnerability Questions

6. **What is a "bug" in software?**

○A. A feature that enhances security

○B. A flaw or mistake in the software's code that causes incorrect behavior

○C. A type of attack vector

○D. A tool used to detect vulnerabilities

○E. A security control implemented to protect the system

7. **Which of the following is an example of a bug?**

○A. A buffer overflow vulnerability

○B. A calculation error in a financial application

- C. An SQL injection vulnerability
- D. An authentication bypass vulnerability
- E. A denial of service attack

8. **What is the main difference between a bug and a vulnerability?**

- A. Bugs are always security-related, while vulnerabilities are not
- B. Vulnerabilities are always bugs, but bugs are not always vulnerabilities
- C. Bugs affect performance, while vulnerabilities affect security
- D. Vulnerabilities are intentional, while bugs are accidental
- E. Bugs are easier to fix than vulnerabilities

9. **Which of the following is an example of a vulnerability?**

- A. A typo in a user interface string
- B. A crash due to improper memory management
- C. A buffer overflow that allows arbitrary code execution
- D. A calculation error in a financial application
- E. A feature that improves usability

10. **Which of the following statements is true about vulnerabilities?**

○A. They always cause the software to crash

○B. They are always intentional

○C. They can be exploited to compromise the security of a system

○D. They are unrelated to bugs

○E. They only affect the usability of the software

Attack Vectors Questions

11. **What is an "attack vector"?**

○A. A tool used to detect vulnerabilities

○B. A method used by attackers to enter a system or network

○C. A type of bug in the software

○D. A security control implemented to protect the system

○E. A feature that improves system performance

12. **Which of the following is NOT a common attack vector?**

○A. Social engineering attacks

○ B. Credential theft

○ C. Vulnerability exploits

○ D. Debugging tools

○ E. Insider threats

13. **In the museum analogy, what do the "front doors" and "windows" represent?**

○ A. Vulnerabilities

○ B. Attack vectors

○ C. Mitigations

○ D. Assets

○ E. Bugs

14. **Why is it difficult to close off all attack vectors in modern computing systems?**

○ A. Because attackers are always one step ahead

○ B. Because modern systems are too simple

○ C. Because modern systems and application environments are complex

○ D. Because there are no tools to detect attack vectors

○ E. Because attack vectors are always intentional

15. **What can strong security practices and safeguards achieve regarding attack vectors?**

○ A. Eliminate all attack vectors

○ B. Make it more difficult for attackers to find and use attack vectors

○ C. Introduce new vulnerabilities

○ D. Increase the complexity of the system

○ E. Reduce the performance of the system