**Abuse Case Template and Instruction**

Associated Use Case/Functionality:

Keyword(s) that was used to search for the attack pattern:

Abuse Case Name:

Attack Pattern Name and ID:

Abuse Case Objective:

Precondition/Prerequisite:

Resources:

Flow of Events:

Post Condition:

Solution & Mitigation:

**Abuse Case Description Template Instruction**

**Associated Use Case/Functionality: <***The use case or functionality that the abuse case affects>*

**Abuse Case Name:** *<The abuse case name should be based on the Use Case Name (from the Attacker perspective) and threats.* The abuse case should include attacker and/or user. For example, *instead of using "Steals motorcycle" as an abuse case name, it is suggested to use "Attacker steals owner motorcycle" as the abuse case name.*

**Attack Pattern Name and ID:** *<The name of the attack pattern used for developing this abuse case along with its ID.>*

**Keyword(s) that was used to search for the attack pattern** :< *List the keyword(s) that gave this attack pattern* >

**Abuse Case Objective:** *<What is the Attacker trying to achieve from this attack? Information from the summary of the attack pattern can be used in writing the objective. >*

**Precondition/Prerequisite:** *<The state that the system should be in for this abusive interaction.>*

**Resources:** *< The information or tools the Attacker should have for this attack to be carried out.>*

**Flow of Events:***<How is the attack executed?> Should have the following components:*

1. Explore: How to check if the use case can be exploited
2. Experiment: Indication of possible vulnerability
3. Exploit: Execute a complete attack.

**Post Condition:***<What would happen to the system if this attack is carried out.>*

**Solution & Mitigation:** *<How can the risk of this abuse case be minimized/removed?>*

***Information from relevant CAPEC attack patterns can be used to fill out the above template. The following shows the mapping of the CAPEC attack pattern fields to the above template fields.***

|  |  |
| --- | --- |
| **Abuse Case Template Fields** | **CAPEC Attack Pattern Fields** |
| Abuse Case Objective | Summary of attack pattern. |
| Precondition/Prerequisite | Attack Prerequisites |
| Resources | Resources Required |
| Flow of Events | Attack Execution Flow |
| Post Condition | Base this on the Summary of the attack pattern or any other information of the attack pattern. |
| Solution & Mitigation | Solutions and Mitigations of attack pattern, and the potential mitigation of the related CWE associated with the attack pattern. |

**Abuse Case Example**

**Associated Use Case/Functionality:** Check Out

**Keyword(s) that was used to search for the attack pattern**: Shopping

**Abuse Case Name:** Attacker makes fraudulent transaction

**Attack Pattern Name and ID:** 74, Manipulating User State

**Abuse Case Objective:** The abusive interaction is to modify state information maintained by the Check Out process. State information, such as username, payment information, and other browser history that are used to complete a valid checkout transaction is manipulated by an adversary to conduct fraudulent transactions.

**Precondition/Prerequisite:** Knowledge of the application state conditions

**Resources:** No special resources are required. An attacker can choose to use a data tampering tool to aid in the attack.

**Flow of Events:**

1. The customer clicks the check-out button.
2. The application displays the items in the shopping cart of the customer.
3. The attacker can add or remove items during this step.
4. The attacker clicks the “continue to payment” button.
5. The attacker enters the payment information.
6. The attack then clicks the OK button to complete the order.
7. The application checks that the credit card is valid.
8. The attacker uses a tool to modify the information that will be sent to the server, for example, modify the price.
9. The application processes the transaction, and displays the confirmation page.
10. The application sends a confirmation via customer’s email.

**Post Condition:** Fraudulent transaction is made. For example, the attack buys the products with a cheaper price.

**Solution & Mitigation:**

1. Do not rely solely on user-controllable locations, such as cookies or URL parameters, to maintain user state.
2. Do not store sensitive information, such as usernames or authentication and authorization information, in user-controllable locations.
3. At all times, sensitive information that is part of the user state must be appropriately protected to ensure confidentiality and integrity at each request.