EU-Rent Car (misuse case and mitigation template)

## Get unauthorized access via SQLi

|  |  |
| --- | --- |
| Misuse case ID | MUC-01 |
| Misuse case Name | Get unauthorized access via SQLi |
| Brief Description | A malicious actor attempts to gain unauthorized access to a system or application by exploiting a SQL Injection (SQLi) vulnerability. |
| Precondition | 1. System has at least one registered user. 2. The system's network security is inadequate for detecting spoofed IP addresses |
| Basic Threat Flow | 1. The MALICIOUS user PROVIDES SQLI VALUES IN input fields of the url. 2. The MALICIOUS user BYPASS the validation REQUEST TO server program. 3. The system executes the query provided in the url. 4. The system evaluates the query in the database. 5. The system VALIDATES THAT the query is successful. 6. The system SENDS the welcome message TO the MALICIOUS user. |
| Alternate threat Flow | RFS 5   1. **DO** 2. The system SEND the database error message DATA to the MALICIOUS user. 3. The MALICIOUS user EXPOLITES the database error message DATA from the system. 4. RESUME STEP 1 5. **UNTIL** the query is successful. 6. RESUME STEP 6 |
| Postcondition | 1. The attacker successfully get unauthorized access to the database |

## Intercept communication via SSL stripping

|  |  |
| --- | --- |
| Misuse case ID | MUC-02 |
| Misuse case Name | Intercept communication via SSL stripping |
| Brief description | A malicious actor employs SSL stripping techniques to intercept communication between a client and a server, despite the use of HTTPS (HTTP over SSL/TLS) encryption. |
| Precondition | 1. The user application is not configured to enforce the use of HTTPS connection. 2. The MALCIOUS user positioned between the user and server in the network. 3. The system's network security is inadequate for detecting spoofed IP addresses |
| Threat basic flow | 1. The MALICIOUS user intercepts the initial connection attempt in the network. 2. The MALICIOUS user modifies the request with http connection. 3. The MALICOUS user forward the http request connection to server. 4. The MALICOUS user BYPASS the validation of http connection TO the server. 5. The system response with the http connection. 6. The MALICIOUS user intercepts the http connection response. 7. The MALICIOUS user modifies the http connection response. 8. The MALICIOUS user forwards the http connection response to the user. 9. The customer SEND sensitive DATA over the insecure http connection. 10. The MALCIOUS user intercept the customer DATA on the network. |
| Threat alternate flow | RFS 5   1. The system VALIDATES THAT the connection is insecure. 2. ABORT.   RFS 10   1. IF the intercepted information is encrypted THEN 2. DO 3. The MALICIOUS user attempts the encrypt keys to decrypt the customer DATA. 4. UNTIL The MILICIOUS user successfully gets the data. 5. ENDIF 6. RESUME STEP 10. |
| Post condition | 1. The MALICIOUS user intercepts the user sensitive DATA without authorization. |

## Modify Persistent data via IDOR

|  |  |
| --- | --- |
| Misuse case ID | MUC-03 |
| Misuse case Name | Modify persistent data via IDOR |
| Brief description | A malicious actor exploits an Insecure Direct Object Reference (IDOR) vulnerability to modify persistent data within a web application, allowing unauthorized users to access or manipulate sensitive data by directly referencing object identifiers (IDs) in requests. |
| Precondition | 1. The data is stored in database. 2. The MALCIOUS user get unauthorized access to the database through network. 3. The system's network security is inadequate for detecting spoofed IP addresses |
| Threat basic flow | 1. The MALICIOUS user identifies the sensitive data in the system. 2. The MALICIOUS user GETS object reference id FROM the url. 3. The MALICIOUS user tamper with the object reference id. 4. The MALICIOUS user BYPASS the validation REQUEST TO the system. 5. The system VALIDATES THAT the url is valid. 6. The MALCIOUS user get access to the user DATA in database. 7. The MALICIOUS user reads the user DATA in database. 8. The MALCIOUS user modify the DATA in the database. |
| Threat alternate flow | RFS 5   1. The system detects the URL invalid. 2. ABORT   RFS 7   1. IF the customer DATA is encrypted THEN 2. **DO** 3. The MALICIOUS user attempts the encrypt keys to decrypt the customer DATA. 4. **UNTIL** The MILICIOUS user successfully gets the data. 5. ENDIF 6. RESUME STEP 8. |
| post condition | The MALICIOUS user modifies the persistent data in database. |

## Mask a page

|  |  |
| --- | --- |
| Aspect Misuse case ID | AMUC-04 |
| Aspect Misuse case Name | Mask a page |
| Brief description | Obscure or hide the content of a web page from the user's view. |
| Precondition | 1. The system follows insecure coding practices. 2. The system has misconfigurations. 3. The system's network security is inadequate for detecting spoofed IP addresses |
| Threat Basic Flow | 1. The MALICIOUS user crafts a payload with code snippet. 2. The MALICIOUS user EXPOLITES the code injection vulnerability of server. 3. The MALICIOUS user GETS code FROM the code repository. 4. The MALICIOUS user PROVIDES REC VALUES IN the server code. 5. the MALICIOUS user SENDS a MALICIOUS payload TO the server program. 6. The MALICIOUS code BYPASS the secure code validation REQUEST TO server. 7. The system VALIDATES THAT the payload is valid. 8. The MALCIOUS user modify the server code with desired content. |
| Post condition | 1. The MALICIOUS user successfully defaces the target website. |

## Create Malicious Ads

|  |  |
| --- | --- |
| Aspect Misuse case ID | AMUC-05 |
| Aspect Misuse case Name | Create malicious ads |
| Brief description | A malicious user designs and Malvertising with the intent to deceive, exploit, or harm users who interact with them. |
| Precondition | 1. insecure coding practices 2. misconfigurations, or outdated software. 3. The system's network security is inadequate for detecting spoofed IP addresses |
| Basic Threat Flow | 1. The MALICIOUS user crafts a MALCIOUS ad with code snippet. 2. The MALICIOUS user EXPOLITES the code injection vulnerability of server. 3. The MALICIOUS user GETS code FROM the code repository. 4. The MALICIOUS user PROVIDES REC VALUES IN the server code. 5. the MALICIOUS user SENDS a MALICIOUS ad payload TO the server program. 6. The MALICIOUS ad BYPASS the secure code validation REQUEST TO server. 7. The system VALIDATES THAT the MALICIOUS ad is valid. 8. The system displays a MALICIOUS ad in the url. |
| Alternate Threat Flow | RFS 7   1. The system VALIDATES THAT the MALICIOUS ad is invalid. 2. ABORT |
| Post condition | The MALICIOUS user successfully creates a malicious pop up in the system. |

## mitigation: Validate input.

|  |  |
| --- | --- |
| Mitigation ID | 01 |
| Mitigation Name | Validate input |
| Brief Description | Mitigate the threat by validating user-supplied data to ensure that it conforms to the expected format, structure, and constraints defined by the application. |
| Precondition | The system has received malicious script. |
| Basic Mitigation Flow | 1. The system sanitizes the inputs according to the input speciﬁcation. 2. The system VALIDATES THAT the inputs are valid. |
| Alternate Mitigating Flow | RFS 2  1 The system displays an error message.  2 ABORTS. |
| Postcondition | The system has successfully validated the inputs. |

## mitigation: Encrypt Data.

|  |  |
| --- | --- |
| Mitigation ID | 02 |
| Mitigation Name | Encrypt data |
| Brief Description | Involves transforming plaintext data into ciphertext using an encryption algorithm and a cryptographic key. |
| Precondition | 1. Identification of sensitive data that needs to be protected. 2. Selection of appropriate encryption algorithms and keys. |
| Basic Mitigation Flow | 1. The system identifies the sensitive data entered by the user. 2. The system applies the encryption algorithm to secure the transmission of secure data. 3. The system applies the encryption algorithm on the data stored in database. |
| Alternate Mitigating Flow | RFS 2   * 1. The system fails to apply the encryption during transit.   RFS 3   1. un-encrypted data stored in data base |
| Post condition | Sensitive data is encrypted |

## mitigation: Monitor Network.

|  |  |
| --- | --- |
| Mitigation ID | 05 |
| Mitigation Name | Monitor network |
| Brief Description | Involves analysing network traffic patterns, protocols, and data flows to identify abnormal behaviour and detect security threats |
| Precondition | 1. The developer Deploy an Intrusion Prevention The system (IPS) within the network infrastructure. |
| Basic Mitigation Flow | 1. The IPS continuously monitors incoming and outgoing network traffic, analysing packet headers and payload content. 2. The system utilizes signature-based detection mechanisms within the IPS to match patterns of known traffic patterns. 3. IPS detect deviations from normal network behaviour. 4. IPS decrypt and inspect encrypted traffic. 5. detect Address Resolution Protocol (ARP) spoofing or MAC address anomalies within the network, 6. the IPS generates real-time alerts or triggers predefined actions. 7. IPS isolate the affected segments of the network. 8. IPS block the malicious traffic |
| Alternate Mitigation Flow |  |
| Postcondition | 1. The IPS logs all detected MitM-related activities, including details of identified threats, actions taken, and traffic analysis results. |

# Mitigation:IPS.

|  |  |
| --- | --- |
| AFMUC ID | AFMUC-05 |
| AFMUC name | Monitor network |
| Brief Description | Involves analysing network traffic patterns, protocols, and data flows to identify abnormal behaviour, spoofed IPS and detect security threats |
| Precondition | 1. The Intrusion Prevention (IPS) installed within the network infrastructure. 2. The network is operational and monitoring traffic. |
| Basic Mitigation Flow | 1. The IPS continuously monitors incoming and outgoing network traffic, analysing packet headers and payload content. 2. The system utilizes signature-based detection mechanisms within the IPS to match patterns of known traffic patterns. 3. IPS detect deviations from normal network behaviour. 4. IPS decrypt and inspect encrypted traffic. 5. IPS detect Address Resolution Protocol spoofing or MAC address anomalies within the network, 6. IPS generates real-time alerts or triggers predefined actions. 7. IPS isolate the affected segments of the network. 8. IPS block the malicious traffic |
| Alternate Mitigation Flow | RFS 2   1. The IPS mistakenly identifies legitimate traffic as spoofed. 2. The IPS drops legitimate packets.   RFS 5   1. The IPS does not identify the spoofed IP 2. The spoofed IP packets are not flagged and continue to the network. 3. The network is compromised |
| Postcondition | 1. Spoofed IP addresses are detected and mitigated. 2. The network remains secure from the intrusion attempt. |