**Interaction: Authentication Request**

**1. Spoofing the Third Party Application External Entity  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | Third Party Application may be spoofed by an attacker and this may lead to unauthorized access to Key. Consider using a standard authentication mechanism to identify the external entity. |
| **Justification:** | <no mitigation provided> |

**2. Elevation Using Impersonation  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Elevation Of Privilege |
| **Description:** | Key may be able to impersonate the context of Third Party Application in order to gain additional privilege. |
| **Justification:** | <no mitigation provided> |

**3. Cross Site Request Forgery  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Elevation Of Privilege |
| **Description:** | Cross-site request forgery (CSRF or XSRF) is a type of attack in which an attacker forces a user's browser to make a forged request to a vulnerable site by exploiting an existing trust relationship between the browser and the vulnerable web site. In a simple scenario, a user is logged in to web site A using a cookie as a credential. The other browses to web site B. Web site B returns a page with a hidden form that posts to web site A. Since the browser will carry the user's cookie to web site A, web site B now can take any action on web site A, for example, adding an admin to an account. The attack can be used to exploit any requests that the browser automatically authenticates, e.g. by session cookie, integrated authentication, IP whitelisting, … The attack can be carried out in many ways such as by luring the victim to a site under control of the attacker, getting the user to click a link in a phishing email, or hacking a reputable web site that the victim will visit. The issue can only be resolved on the server side by requiring that all authenticated state-changing requests include an additional piece of secret payload (canary or CSRF token) which is known only to the legitimate web site and the browser and which is protected in transit through SSL/TLS. See the Forgery Protection property on the flow stencil for a list of mitigations. |
| **Justification:** | <no mitigation provided> |

**4. Elevation by Changing the Execution Flow in Keycloak  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Elevation Of Privilege |
| **Description:** | An attacker may pass data into Key in order to change the flow of program execution within Key to the attacker's choosing. |
| **Justification:** | <no mitigation provided> |

**5. Keycloak May be Subject to Elevation of Privilege Using Remote Code Execution  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Elevation Of Privilege |
| **Description:** | Third Party Application may be able to remotely execute code for Key. |
| **Justification:** | <no mitigation provided> |

**6. Data Flow Generic Data Flow Is Potentially Interrupted  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Denial Of Service |
| **Description:** | An external agent interrupts data flowing across a trust boundary in either direction. |
| **Justification:** | <no mitigation provided> |

**7. Potential Process Crash or Stop for Keycloak  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Denial Of Service |
| **Description:** | Key crashes, halts, stops or runs slowly; in all cases violating an availability metric. |
| **Justification:** | <no mitigation provided> |

**8. Data Flow Sniffing  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | Data flowing across Authentication Request may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow. |
| **Justification:** | <no mitigation provided> |

**9. Potential Data Repudiation by Keycloak  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Repudiation |
| **Description:** | Key claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. |
| **Justification:** | <no mitigation provided> |

**10. Potential Lack of Input Validation for Keycloak  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Tampering |
| **Description:** | Data flowing across Authentication Request may be tampered with by an attacker. This may lead to a denial of service attack against Key or an elevation of privilege attack against Key or an information disclosure by Key. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach. |
| **Justification:** | <no mitigation provided> |

**11. Spoofing the Keycloak Process  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | Key may be spoofed by an attacker and this may lead to information disclosure by Third Party Application. Consider using a standard authentication mechanism to identify the destination process. |
| **Justification:** | <no mitigation provided> |

**Interaction: Authentication Response**

**26. External Entity End User Potentially Denies Receiving Data  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Repudiation |
| **Description:** | End User claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data. |
| **Justification:** | <no mitigation provided> |

**27. Spoofing of the End User External Destination Entity  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | End User may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of End User. Consider using a standard authentication mechanism to identify the external entity. |
| **Justification:** | <no mitigation provided> |

**Interaction: JDBC Query**

**42. Spoofing of Destination Data Store Database  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of Database. Consider using a standard authentication mechanism to identify the destination data store. |
| **Justification:** | <no mitigation provided> |

**43. Potential Excessive Resource Consumption for Keycloak or Database  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Denial Of Service |
| **Description:** | Does Key or Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout. |
| **Justification:** | <no mitigation provided> |

**Interaction: Result Set**

**44. Spoofing of Source Data Store Database  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Spoofing |
| **Description:** | Database may be spoofed by an attacker and this may lead to incorrect data delivered to Key. Consider using a standard authentication mechanism to identify the source data store. |
| **Justification:** | <no mitigation provided> |

**45. Weak Access Control for a Resource  [State: Not Started]  [Priority: High]**

|  |  |
| --- | --- |
| **Category:** | Information Disclosure |
| **Description:** | Improper data protection of Database can allow an attacker to read information not intended for disclosure. Review authorization settings. |
| **Justification:** | <no mitigation provided> |