|  |  |
| --- | --- |
| **Application Name** | Mojaloop Portals |
| **Application Version** | 1.0 |
| **Description** | A review of the portal’s application using the STRIDE threat model |
| **Document Owner** | Victor Akidiva |
| **Participants** | Victor Akidiva |
| **Reviewer** | Godfrey Kutumela |

**Executive Summary**

The Portals application will serve as the landing page for Mojaloop clients to interact with the backend Mojaloop deployment for the following key purposes:

1. Portlet sandboxing
2. Single click configurations
3. Certificate exchange via CM backend
4. Reporting and Searching
5. Workflows
6. User access control e.t.c

The Portals frontend will provide access to the following back end services:

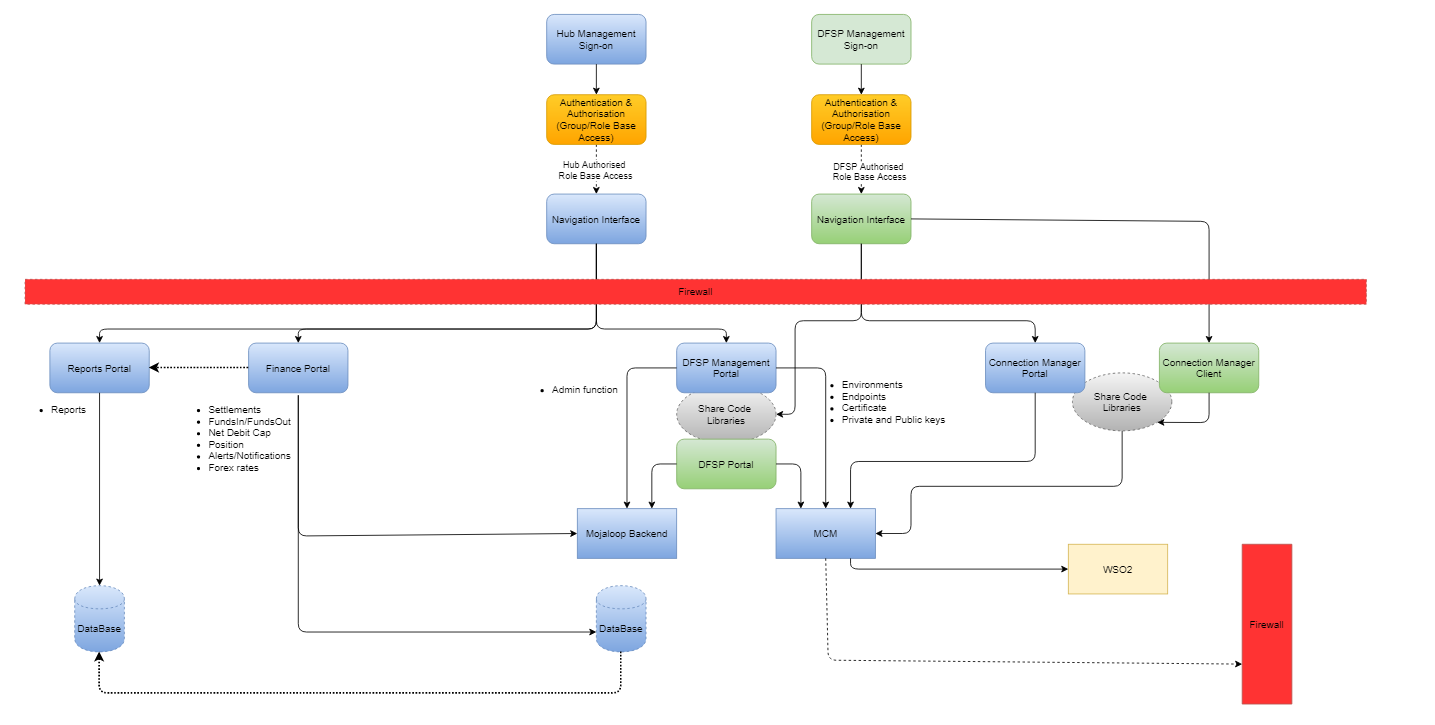
1. Reports portal
2. Finance portal
3. DFSP Management portal
4. Connection manager

**Risk and objective**

The portals application acts as single pane gateway to critical services in Mojaloop backend, on this note it is classified as a **high risk** application requiring robust controls around its deployment, management and operations.

The purpose of this report was to review the Portals proposed design and review inherent controls, making recommendations on any additional controls to be implemented.

**Portals proposed design**



**Security Requirements summary**

The following are high level requirements for implementation within the portals application. These are high level recommendations based on OWASP Top 20 controls for web applications as well as PCI-DSS guidelines for PII data protection.

1. Web application firewall / reverse proxy - This will act as a filter to protect portals application from web application attacks and depending on capabilities offer intrusion detection.
2. Data Protection controls
   1. Restrict access to sensitive data
   2. Encrypt data at rest and in motion
3. Identity and Access management - using a proper identity management solution, Portals can enforce strong authentication and authorization procedures including:
   1. User access management and workflows with least privilege controls
   2. API authentication and authorization
   3. Multi Factor Authentication with configurable policies
   4. Session management and control
4. Input validation within Portals as well as back end APIs
   1. Secure database communication and queries
5. Secure communication
   1. Use SSL/TLS for all communication
   2. Secure PKI artifact exchange
   3. Secure cookies, headers
6. Logging controls as prescribed by secure logging standard:
   1. Authentication and authorization calls
   2. Privilege changes
   3. Administrative actions
   4. Access to sensitive data
   5. Sanitise error messages

## **External Dependencies**

External dependencies are items external to the code of the application that may pose a threat to the application. These items are typically still within the control of the organization, but possibly not within the control of the development team.

|  |  |
| --- | --- |
| **Dependency / Components** | **Description** |
| Web Server | Details of selected web server |
| Operating System | Details of OS to be used |
| Database | Details of back end database used |
| UI Framework | Details of selected framework to be used to code application |
| Application Server | Will the application have a web server and application server? |
| Application Server | Application server to sit in DMZ behind a firewall |
| API Calls | API calls to back end applications via HTTPS |
| SSL encryption | DFSP users access portal via HTTPS url |
| Report SQL Queries | Optimised queries to be used for reporting and updates |
| SSL Certificate | Certificate for use in portals procured from a valid CA provider |
| Database tuning | Backend database to be tuned to optimise query responses, avoid timeouts and validate all inputs before execution |
| IAM server | We currently assume its using WSO2 to manage identities for users and APIs communicating with the portals. |

**Threats and Countermeasures**

|  |  |  |
| --- | --- | --- |
| **Threat** | **Stride Mapping** | **Countermeasure** |
| Compromised credentials | Spoofing | Strong authentication controls   * IP binding * IP Whitelisting * MFA * SSL authentication |
| Privilege escalation - Users | Elevation of privilege | Access controls   * RBAC within portal * Audit logs |
| Privilege escalation - Devices and Services | Elevation of privilege | Access controls   * MTLS authentication for servers * Least privilege server access for hub admins * Audit logs |
| Privilege escalation - APIs | Elevation of privilege | Access controls   * API Authentication and authorisation * API gateway baseline configs * Audit logs |
| Insecure configuration | Information disclosure | Baseline setup and configuration for:   * Firewalls * API gateway * Servers * Database * Web Server * Application Server * SQL Optimisation * Audit logging |
| Insecure communication - exposed sensitive data | Information disclosure | Encryption and data access controls |
| Exposed services are prone to intrusion attacks on public IPs | Spoofing  Information disclosure  Denial of service | Baseline setup and configuration for:   * Firewalls * Intrusion detection rules * Audit logging |
| Application availability | Denial of service | Configurations to ensure uptime   * High availability |
| Data exfiltration | Information Disclosure | Authentication and intrusion detection controls   * Database access controls * API authentication and authorisation * Query optimisation * Input validation |
| Zero day vulnerabilities | All | Baseline setup and configuration for:   * Firewalls * API gateway * Servers * Database * Web Server * Application Server * SQL Optimisation * Audit logging * Patch Management * Change Management |

**Components and Entry Points**

Entry points define the interfaces through which potential attackers can interact with the application or supply it with data. In order for a potential attacker to attack an application, entry points must exist. Entry points in an application can be layered, for example each web page in a web application may contain multiple entry points.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Trust Level** |
| HTTPS Port | The Portals application will only be accessible via TLS. All pages within the Portals application are layered on this entry point. | (14) Valid API DFSP token  (15) Valid MTLS authenticated user  (16) Valid DFSP IP (Whitelisted) |
| Portal landing page | The splash page for the Portals application is the entry point for all users. | (14) Valid API DFSP token  (15) Valid MTLS authenticated user  (16) Valid DFSP IP (Whitelisted) |
| Portal login page | Login page for all users | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User  (16) Whitelisted IP |
| Navigation page | Page allowing users to navigate various features of portals application | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| APi Endpoint | API endpoint that will receive requests from Portal for processing | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |

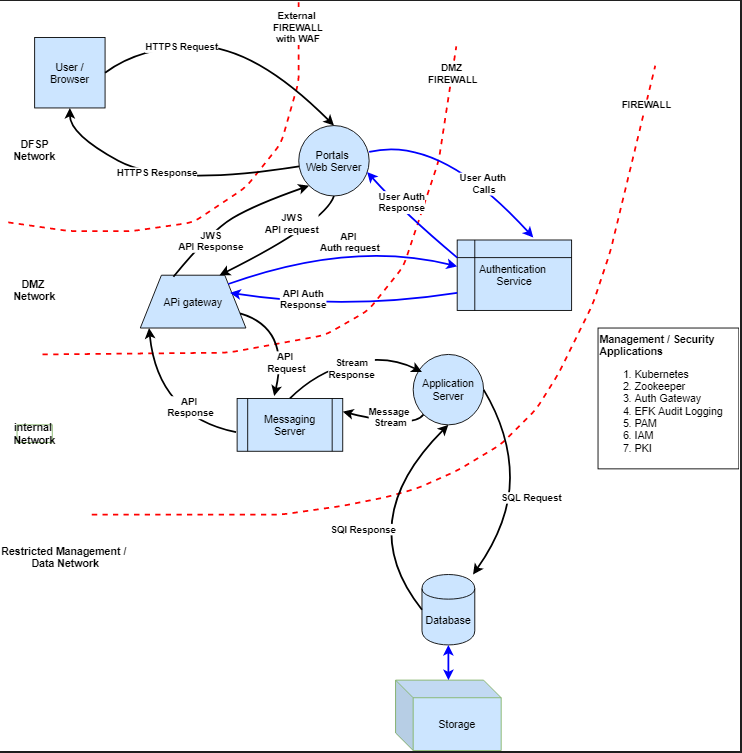
**Assets/ Components**

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Trust Level** |
| Portal Users | Portal users at Hub and DFSP. | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Hub User login | Portal user within the hub | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| DFSP User login | Portal user within the valid DFSP | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Customer data | Any customer data sent to and / or from the portal | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Hub User data | Details about a portal hub user profile | (2) User with Valid Login Credentials  (4) Hub User  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| DFSP user data | Details about a DFSP user profile | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| DFSP data | Details about a DFSP entity profile | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Web Server code execution | Permissions to execute code in web server | (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Database read execution | Permissions to execute code in web server | (7) Web Server User Process (8) Database Read User  (9) Database Read User |
| Database Read/Write execution | Permissions to execute SQL code in database server | (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| API calls | Permission to execute API calls on exposed endpoint | (14) Valid API DFSP Token  (15) Valid MTLS authenticated user |
| User session | Data on user session | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Create Users | Ability to create users - DFSP | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Create users - hub | Ability to create users - hub | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Access audit data - DFSP | Access to view audit log data | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Access audit data - Hub | Access to view audit log data | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Search DFSP information | Access to DFSP search functionality | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Print report data | Access to DFSp print report | (2) User with Valid Login Credentials  (4) Hub user  (5) Database Server Administrator  (7) Web Server User Process (8) Database Read User  (9) Database Read/Write User |
| Endpoint URL / IP / Hostname | Access to published services | (16) Valid DFSP IP |

**Trust Levels**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Description** |
| 1 | Anonymous Hub user | Unauthenticated user to the portal from within hub |
| 2 | Anonymous DFSP user | Unauthenticated user to the portal from within DFSP network |
| 3 | Valid hub user | User with valid credentials within hub |
| 4 | Valid DFSP user | User with valid credentials within DFSP |
| 5 | Hub Officers | User with valid credentials within hub |
| 6 | Hub IT administrator - web application | User with valid credentials within hub to manage Infrastructure and applications |
| 7 | Hub IT Administrator - application server | User with valid credentials within hub to manage applications |
| 8 | Hub IT administrator - database | User with valid credentials within hub to manage the database server |
| 9 | Hub IT administrator - API gateway | User with valid credentials within hub to manage Infrastructure - APi gateway |
| 10 | Hub IT administrator - WSO2 | User with valid credentials within hub to manage Infrastructure - WSO2 |
| 11 | Web server process user | Web server process |
| 12 | Database user read only | Read only DB user |
| 13 | Database user read / write | Read/Write DB user |
| 14 | Valid API DFSP token | User with valid API token to push API calls to hub |
| 15 | Valid MTLS authenticated user | Authenticated DFSP using valid SSL certificates |
| 16 | Valid DFSP IP (Whitelisted) | Whitelisted DFSP IP |

**Data Flow Diagram**



**STRIDE Threat descriptions**

|  |  |  |
| --- | --- | --- |
| **Type** | **Examples** | **Security Controls** |
| Spoofing (S) | Threat action aimed to illegally access and use another user's credentials, such as username and password. | Strong authentication |
| Tampering (T) | Threat action aimed to maliciously change/modify persistent data, such as persistent data in a database, and the alteration of data in transit between two computers over an open network, such as the Internet. | Integrity |
| Repudiation (R) | Threat action aimed to perform illegal operations in a system that lacks the ability to trace the prohibited operations. | Non repudiation |
| Information disclosure (I) | Threat action to read a file that one was not granted access to, or to read data in transit. | Confidentiality |
| Denial of service (D) | Threat aimed to deny access to valid users, such as by making a web server temporarily unavailable or unusable. | Resilience and business continuity |
| Elevation of privilege (E) | Threat aimed to gain privileged access to resources for gaining unauthorized access to information or to compromise a system. | Authorisation |

**STRIDE in applications and networks**

The following table summarises how key application components are affected by threats as viewed from a STRIDE perspective:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Component** | **S** | **T** | **R** | **I** | **D** | **E** |
| External Entity | **X** |  | **X** |  |  |  |
| Process | **X** | **X** | **X** | **X** | **X** | **X** |
| Data Flow |  | **X** |  | **X** | **X** |  |
| Data Store |  | **X** | **X** | **X** | **X** | **X** |

**Threats and Countermeasures**

|  |  |  |
| --- | --- | --- |
| **Threat** | **Stride Mapping** | **Countermeasure** |
| Compromised credentials | Spoofing | Strong authentication controls   * IP binding * IP Whitelisting * MFA * SSL authentication |
| Privilege escalation - Users | Elevation of privilege | Access controls   * RBAC within portal * Audit logs |
| Privilege escalation - Devices and Services | Elevation of privilege | Access controls   * MTLS authentication for servers * Least privilege server access for hub admins * Audit logs |
| Privilege escalation - APIs | Elevation of privilege | Access controls   * API Authentication and authorisation * API gateway baseline configs * Audit logs |
| Insecure configuration | Information disclosure | Baseline setup and configuration for:   * Firewalls * API gateway * Servers * Database * Web Server * Application Server * SQL Optimisation * Audit logging |
| Insecure communication - exposed sensitive data | Information disclosure | Encryption and data access controls |
| Exposed services are prone to intrusion attacks on public IPs | Spoofing  Information disclosure  Denial of service | Baseline setup and configuration for:   * Firewalls * Intrusion detection rules * Audit logging |
| Application availability | Denial of service | Configurations to ensure uptime   * High availability |
| Data exfiltration | Information Disclosure | Authentication and intrusion detection controls   * Database access controls * API authentication and authorisation * Query optimisation * Input validation |
| Zero day vulnerabilities | All | Baseline setup and configuration for:   * Firewalls * API gateway * Servers * Database * Web Server * Application Server * SQL Optimisation * Audit logging * Patch Management * Change Management |

**Best Practice Standard mapping - OWASP (2017 release)**

|  |  |  |  |
| --- | --- | --- | --- |
| **OWASP Application Risk** | **Description** | **STRIDE mapping** | **Control / Feature** |
| A1:2017- Injection | Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization. | Spoofing  Elevation of privilege | Strong authentication controls   * Input validation * Query optimisation * IP binding * IP Whitelisting * MFA * SSL authentication * RBAC within portal * Firewall * WAF filtering |
| A2:2017-Broken Authentication | Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users’ identities temporarily or permanently. | Spoofing  Elevation of privilege | Strong authentication controls   * IP binding * IP Whitelisting * MFA * SSL authentication * RBAC within portal * API token authorisation |
| A3:2017- Sensitive Data Exposure | Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. | Information disclosure | Authentication and intrusion detection controls   * Database access controls * API authentication and authorisation * Query optimisation * Input validation |
| A4:2017-XML External Entities (XXE) | External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks. | Information disclosure  Denial of service | Data input validation controls   * API authentication and authorisation * Query optimisation * Input validation * WAF filtering |
| A5:2017-Broken Access Control | Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users’ data, change access rights, etc. | Privilege escalation  Information disclosure | Data input validation controls   * RBAC controls * Audit logs |
| A6:2017-Security Misconfiguration | Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. | All | SecOps processes. Baseline setup and configuration for:   * Firewalls * API gateway * Servers * Database * Web Server * Application Server * SQL Optimisation * Audit logging * Patch Management |
| A7:2017-  Cross-Site  Scripting (XSS) | XSS flaws occur whenever an application includes untrusted data in a new web page without  proper validation or escaping, or updates an existing web page with user-supplied data using a  browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the  victim’s browser which can hijack user sessions, deface web sites, or redirect the user to  malicious sites | Spoofing  Repudiation |  |
| A8:2017-  Insecure  Deserialization | Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks. | Spoofing  Elevation of Privilege |  |
| A9:2017-Using  Components  with Known  Vulnerabilities | Components, such as libraries, frameworks, and other software modules, run with the same  privileges as the application. If a vulnerable component is exploited, such an attack can facilitate  serious data loss or server takeover. Applications and APIs using components with known  vulnerabilities may undermine application defenses and enable various attacks and impacts. | Information Disclosure  Elevation of Privilege |  |
| A10:2017-  Insufficient  Logging &  Monitoring | Insufficient logging and monitoring, coupled with missing or ineffective integration with incident  response, allows attackers to further attack systems, maintain persistence, pivot to more systems,  and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over  200 days, typically detected by external parties rather than internal processes or monitoring | Elevation of privilege |  |
|  |  |  |  |

**Attacker Stories / Abuse Cases**

|  |  |  |
| --- | --- | --- |
| **Story** | **Description** | **Stride** |
| As an attacker I want to access the published portals using a spoofed IP address | Attempt to spoof a valid whitelisted IP | Spoofing |
| As an attacker I want to use malicious SQL code to manipulate the portal backend database so it reveals information. | Attempt SQL injection | Information disclosure  Privilege escalation |
| As an attacker I want to attempt to load invalid URLs or force error input so I can get information on backend services based on error response | Attempted error generation for information disclosure | Information Disclosure  Denial of service |
| As an attacker I want to attempt to launch an XSS injection attack targeting users in order to access accounts, activate Trojans or modify page content. | Attempted user redirection, injection of malicious code and exposing fake pages to users to harvest information | Information Disclosure  Spoofing |
| As an attacker I want to inject malicious files into the portals application via exposed service endpoints | Malicious file uploads | Privilege escalation |
| As an attacker I want to attempt to perform tasks outside what my user is authorised to perform | Attempt to escalate privileges and perform illegal tasks not configured for my user | Privilege escalation  Spoofing |
| As an attacker I want to attempt to log in multiple times using a valid user with invalid credentials so I test if the Portals application has robust authentication mechanism | Attempt to brute force attack the Portals application using numerous invalid logins. Check how application will respond. This also applies to API tokens | Spoofing  Privilege escalation |
| As an attacker I want to attempt to log in to the Portals application and exposed supporting services / infrastructure using default credentials | Testing use of default credentials in exposed applications / infrastructure | Privilege escalation |
| As an attacker I want to attempt to eavesdrop traffic to and from Portals server using a man in the middle attack | Testing data protection in motion using encryption. | Information Disclosure |
| As an attacker I want to attempt to enumerate and take advantage of weak / old cryptographic algorithms in use by the portals to break encryption and steal data | Enumerate encryption algorithms used and attempt to breach data encryption | Information disclosure |
| As an attacker, I want to exploit vulnerable areas of the application where the user or system can upload XML to extract data, execute a remote request from the server, scan internal systems, perform a denial-of-service attack, as well as execute other attacks. | Scan application components for vulnerabilities that are exploitable and attempt to breach system | Information Disclosure  Privilege escalation |
| As an attacker, I want to bypass access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool. | URL breach, session management controls bypass. | Information Disclosure  Privilege escalation |
| As an attacker, I want to leverage metadata manipulation, such as replaying or tampering with a JSON Web Token (JWT) access control token or a cookie or hidden field manipulated to elevate privileges or abusing JWT invalidation. | Input validation breach as well as access controls integrity testing | Information Disclosure  Privilege escalation |
| As an attacker, I force browsing to authenticated pages as an unauthenticated user or to privileged pages as a standard user. | Session validation and access control testing | Privilege escalation |
| As an attacker, I want to access APIs with missing access controls for POST, PUT and DELETE. | API access control testing | Privilege escalation |
| As an attacker, I find areas where the user agent (e.g. app) does not verify if the received server certificate is valid and perform attacks where I get unauthorized access to data. | Test user agent and server certificate validation | Information Disclosure |
| As an attacker, I want to find and exploit missing appropriate security hardening configurations on any part of the application stack, or improperly configured permissions on cloud services. | Test patch management and zero day vulnerabilities | Information Disclosure  Privilege escalation |
| As an attacker, I want to find unnecessary features which are enabled or installed (e.g. unnecessary ports, services, pages, accounts, or privileges) and attack or exploit the weakness. | Identify unnecessary services and applications that are exposed and attempt to exploit them. | Information Disclosure  Privilege escalation |
| As an attacker, I want to use default accounts and their passwords to access systems, interfaces, or perform actions on components which I should not be able to. | Test for default settings and configurations | Information Disclosure  Privilege Escalation |
| As an attacker, I want to find areas of the application where error handling reveals stack traces or other overly informative error messages I can use for further exploitation. | Test error handling messages and information disclosure | Information Disclosure |
| As an attacker, I find security settings in the application servers, application frameworks (e.g. Struts, Spring, ASP.NET), libraries, databases, etc. not set to secure values. | Test insecure deployedcomponents and libraries | Information Disclosure |
| As an attacker, I find the server does not send security headers or directives or they are not set to secure values. | Test security of server headers | Information Disclosure |
| As an attacker, I attack an organization and the logs, monitoring systems, and teams do not see or respond to my attacks. | Test adequacy of logging and monitoring of security events | Information Disclosure |