Description of basic vulnerabilities

Cross-Site Scripting
Cross-Site Scripting (XSS) attacks are a type of injection problem, in which malicious scripts are injected into the otherwise benign and trusted web sites. An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any unprotected cookies, session tokens, or other sensitive information retained by your browser and used with that site. These scripts can even rewrite the content of the HTML page. A successful exploitation is considered a full compromise of the application.

Cross-Site Request Forgery
Cross-Site Request Forgery (CSRF) is an attack that tricks the victim into loading a page that contains a malicious request. It is malicious in the sense that it inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something. CSRF attacks generally target functions that cause a state change on the server but can also be used to access sensitive data.

Insecure Direct Object References
Insecure Direct Object References occur when an application provides direct access to objects based on user-supplied input. As a result of this vulnerability attackers can bypass authorization and access resources in the system directly, for example database records or files.

Injection vulnerabilities (such as SQL Injection, XML injection or ORM injection)
Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code, often found in SQL queries, LDAP queries, XPath queries, OS commands, program arguments, etc. Injection flaws are easy to discover when examining code, but more difficult via testing. Scanners and fuzzers can help attackers find them.

Authentication bypass
Authentication bypass is a vulnerability that allows an attacker to bypass the authentication mechanism and login without knowing user credentials such as password. An attacker is than able to access resources that are normally inaccessible for unauthenticated user.

Code injection
Code Injection is the general term for attack types which consist of injecting code that is then interpreted/executed by the application. This type of attack exploits poor handling of untrusted data. These types of attacks are usually made possible due to a lack of proper input/output data validation.

Privilege escalation
An attacker is able to gain access to resources that are normally protected from an application or user. There are two types of privilege escalation:

- vertical privilege escalation – This type of privilege escalation occurs when the user or process is able to obtain a higher level of access
- horizontal privilege escalation – The attacker can access resources granted to a

similarly configured account (e.g., in an online banking application, accessing information related to a different user)

Significant Security misconfiguration
Security misconfiguration can happen at any level of an application stack, including the platform, web server, application server, database, framework, and custom code. Developers and system administrators need to work together to ensure that the entire stack is configured properly. Automated scanners are useful for detecting missing patches, misconfigurations, use of default accounts, unnecessary services, etc.

Upload of malicious file
The attacker is able to upload a malicious file without being detected. This malicious file can then be distributed to users and affect their systems.

File Inclusion
The File Inclusion vulnerability allows an attacker to include a file, usually exploiting a "dynamic file inclusion" mechanisms implemented in the target application. The vulnerability occurs due to the use of user-supplied input without proper validation. This can lead to custom code execution, denial of service or sensitive information disclosure.

SQL injection attack
SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploits can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

Command injection
Command injection results in arbitrary command being executed on the server side from untrusted user input. We identified that product ID attribute is passed to Ruby eval function and is possibly executed in the Ruby context when a client with the associated product is loaded from the OSB service. Only administrator is able to specify a malicious product ID when creating a new product.

Open Redirect
Unvalidated redirects and forwards are possible when a web application accepts untrusted input that could cause the web application to redirect the request to a URL contained within untrusted input. By modifying untrusted URL input to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials or other sensitive data.

Authorization bypass
Authorization bypass vulnerability allows an attacker to access resources granted to another account.

Path Traversal
A path traversal attack aims to access files and directories that are stored outside the web root folder. By manipulating variables that reference files with "dot-dot-slash (../)" sequences and its variations or by using absolute file paths, it may be possible to access

arbitrary files and directories stored on file system including application source code or configuration and critical system files.

Session Fixation
Session Fixation is an attack that permits an attacker to hijack a valid user session. The attacker has to provide a legitimate Web application session ID and try to make the victim's browser use it.

Internal SSRF
Server Side Request Forgery (SSRF) is a vulnerability that appears when an attacker has the ability to create requests from the vulnerable server. With SSRF it’s also possible to access services listening on the loopback interface.

Server-side code execution bugs (Stack traces)
Stack traces are not vulnerabilities by themselves, but they often reveal information that is interesting to an attacker. Attackers attempt to generate these stack traces by tampering with the input to the web application with malformed HTTP requests and other input data.