INDUSFACE™

WAS SCAN

Vulnerabilities that Indusface WAS Scans

Disclaimer

Indusface has prepared this document for internal audience. Neither this document nor its content may be copied or distributed outside Indusface, without prior written approval from Indusface.

Notice of Ownership

Vulnerabilities Scanned

S.no	Category	Description	Severity
1	HTTP DELETE Method Enabled	HTTP 'DELETE' method allows a client to delete a file on the web server. An attacker can exploit it as a very simple and direct way to deface a web site or to mount a DoS attack.	Low
2	HTTP Response Splitting	HTTP response splitting is a form of web application attack where unsafe characters are inserted into user-controllable fields which are later inserted into the HTTP header being used for 302 redirects. As per RFC standard, HTTP request headers are separated by one carriage return and line feed, and response headers are separated by two carriage return (CR) and line feed (LF). The response splitting attack consists of making the server print a carriage return line feed sequence followed by content supplied by the attacker in the header section of its response, typically by including them in input fields sent to the application. Response splitting can be used to perform CRLF injection that allows the attacker to set fake cookies, steal CSRF tokens, disclose user information by injecting a script (XSS) and perform a variety of other attacks. It also allows attackers to deactivate & bypass security measures like XSS filters & Same Origin Policy (SOP).	High
3	Microsoft IIS Internal IP Address Disclosure (CVE20020422)	Certain WebDAV methods (PROPFIND, MKCOL, WRITE), when requested with a blank Host field, will return the internal IP of the target host machine. This IP can be used in subsequent attacks to further exploit the target system.	Low

4	Source Code Disclosure	Source code disclosure allows a malicious user to obtain the source code of a server-side application from a webpage. The attacker can obtain deeper knowledge of the Web application logic . Disclosure of source code and configuration files can be devastating for a web application. They usually contain database connection information like IP address, port number and valid credentials. In certain cases, application test users	Medium
5	Cross-Site Scripting (XSS)	The Web application is vulnerable to cross-site scripting (XSS), which allows attackers to take advantage of Web server scripts to inject JavaScript or HTML code that is executed on the client-side browser.	High

		This vulnerability is often caused by server-side scripts written in languages such as PHP, ASP, .NET, Perl or Java, which do not adequately filter data sent along with page requests or by vulnerable HTTP servers. This malicious code appears to come from your Web application when it runs in the browser of an unsuspecting user. An attacker can do the following damage with an exploit script: - access other sites inside another client's private intranet - steal another client's cookie(s) - modify another client's cookie(s) - steal another client's submitted form data - modify another client's submitted form data before it reaches the server - submit a form to your Web application on the user's behalf that modifies passwords or other application data The two most common methods of attack are: - Having a user click a URL link sent in an e-mail	

		Having a user click a URL link while visiting a Web site	

		In both scenarios, the URL will generally link to the trusted site, but will contain additional data that is used to trigger the XSS attack. Note that SSL connectivity does not protect against this issue.	
6	Directory Listing	A web directory was found to be browsable, which means that anyone can see the contents of the directory. These directories can be found: - via page spidering (following hyperlinks), or - as part of a parent path (checking each directory along the path and searching for "Directory Listing" or similar strings), or - by brute forcing a list of common directories. Browsable directories could allow an attacker to view "hidden" files in the web root, including CGI scripts, data files, or backup pages.	Medium

7	SQL Injection	Web applications that do not properly sanitize user input before passing it to a database system are vulnerable to SQL injection. This type of attack potentially allows a malicious user to recover and/or modify any data that the application has access to.	Critical
8	TLS/SSL Server Certificate Expired	The server's HTTPS X.509 certificate is expired.	Critical

9	HTTP TRACE Method Enabled	The HTTP TRACE method is normally used to return the full HTTP request back to the requesting client for proxy-debugging purposes. An attacker can create a webpage using XMLHTTP, ActiveX, or XMLDOM to cause a client to issue a TRACE request and capture the client's cookies. This effectively results in a Cross-Site Scripting attack.	Low
10	Sensitive Form Data Submitted In Cleartext	A web form contains fields with data that is probably sensitive in nature. This form data is submitted over an unencrypted connection, which could allow hackers to sniff the network and view the data in plaintext.	Medium
11	ASP.NET Debug Feature Enabled	The ASP.NET application is running in debug mode which allows a remote user to glean information about an application by using the DEBUG verb in an HTTP request. This can leak information including source code, hidden filenames, and detailed error messages.	Medium
12	HTTP PUT Method Enabled	The Web server contains a flaw that may allow a remote attacker to upload arbitrary files by using the HTTP method 'PUT'. Existing files may be overwritten, resulting in a loss of integrity.	Low

13	Possible Physical Path Disclosure	The web page may disclose the physical path of the web root. While physical path disclosure is not a severe vulnerability by itself, this information can be leveraged by an attacker in combination with other vulnerabilities such as directory traversal.	Medium
14	Missing Secure Flag From Cookie Header	The Secure attribute tells the browser to only send the cookie if the request is being sent over a secure channel such as HTTPS. This will help protect the cookie from being passed over unencrypted requests. If the application can be accessed over both HTTP and HTTPS, then there is the potential that the cookie can be sent in clear text.	Low
15	Sensitive HTML Form Fields With auto-complete	The Web form contains passwords or other sensitive text fields for which the browser auto-complete feature is enabled. Auto-complete stores completed form field and passwords locally in the browser, so that these fields are filled	Low

	Enabled	automatically when the user visits the site again. Sensitive data and passwords can be stolen if the user's system is compromised. Note, however, that form auto-complete is a non-standard, browser-side feature that each browser handles differently. Opera, for example, disregards the feature, requiring the user to enter credentials for each Web site visit.	
16	HTTP Basic Authenticati on Enabled	The HTTP Basic Authentication scheme is not considered to be a secure method of user authentication (unless used in conjunction with some external secure system such as TLS/SSL), as the user name and password are passed over the network as cleartext.	Medium
17	OS Command Injection	An OS command injection vulnerability occurs when a developer uses invalidated user controlled parameters to execute operating system commands. OS command injection vulnerabilities allow attackers to run arbitrary commands on the remote server. This is one of the flaws under the category of Code Injection, in the OWASP Top Ten.	Critical

18	Remote File Inclusion (RFI)	Malicious file execution vulnerabilities are found in many applications. Developers will often directly use or concatenate potentially hostile input with file or stream functions, or improperly trust input files. On many platforms, frameworks allow the use of external object references, such as URLs or file system references. When the data is insufficiently checked, this can lead to arbitrary remote and hostile content being included,	Critical

		processed or invoked by the Web server. This is one of the flaws under the category of Injection, in the OWASP Top Ten.	
19	ASP.NET Unencrypted "__VIEWSTATE" Parameter	The application uses the ASP.NET framework viewstate (__VIEWSTATE) feature without encryption to maintain application state. The viewstate can be protected from tampering by using either encryption or signing. If only signing is used (without encryption), then the internal value of the parameter can be exposed simply by Base64decoding it. In a well-designed application, this parameter should never contain any sensitive information. However, application designers have been known to put passwords and other sensitive data inside the viewstate. Therefore, it is a good idea to always use viewstate encryption in ASP.NET applications.	Medium

20	XPath Injection	XPath is a query language used to select data from XML data sources. It is increasingly common for web applications to use XML data files on the backend, using XPath to perform queries much the same way SQL would be used against a relational database. XPath injection, much like SQL injection, exists when a malicious user can insert arbitrary XPath code into form fields and URL query parameters in order to inject this code directly into the XPath query evaluation engine. Doing so would allow a malicious user to bypass authentication (if an XML-based authentication system is used) or to access restricted data from the XML data source.	High
21	Missing HttpOnly	HTTP Only is an additional flag included in a Set-Cookie HTTP response header. If supported by the browser, using	Low

	Flag From Cookie	the HTTP Only flag when generating a cookie helps mitigate the risk of client side script accessing the protected cookie. If a browser that supports HTTP Only detects a cookie containing the HTTP Only flag, and client side script code attempts to read the cookie, the browser returns an empty string as the result. This causes the attack to fail by preventing the malicious (usually XSS) code from sending the data to an attacker's website.	

22	Unvalidated Redirects And Forwards/Open Redirection	An open redirect vulnerability is an application that takes a parameter and redirects a user to the parameter value, such a Web site, without validation. Attackers exploit this vulnerability with phishing e-mails that cause users to visit malicious sites inadvertently. This is one of OWASP Top Ten flaws in the Code Injection category.	Medium
23	Invalid TLS/SSL Server Certificate	The server's TLS/SSL certificate signature is invalid. This could indicate an attacker is actively attempting to eavesdrop on the connection.	Critical
24	Untrusted TLS/SSL Server Certificate	The server's TLS/SSL certificate is signed by a Certification Authority (CA) that is not a well-known, trusted one. It could indicate that a TLS/SSL man-in-the-middle is taking place and is eavesdropping on TLS/SSL connections.	Critical
25	Application Error Message	An attacker can try to force the target website to produce error messages by passing different attack vectors to different parameters and then analyse the errors to get target information. These errors have no direct security impact, most of the time they indicate a programming error, quality issue, or a potential vulnerability in the application. Many of these types of errors also leak information about the logic or the implementation of the application which can help an attacker to identify or exploit weaknesses in the application.	Medium
26	Email Address Disclosure	There are number of crawlers running across the Internet to search email addresses from all the publicly available websites.	Info

		Such crawlers crate a mailing list to keep sending spam emails. If your email address (example: sales@yourwebsite.com) gets listed in one of such mailing list, your inbox will receive dozens of spam on a daily basis. This may lead to missing out an important email.	
27	Password Field Submitted Using GET Method	The page contains a form with a password field, which submits the password and other user data using the GET method. The contents of the password field will appear in the URL. Sensitive information should not be passed through the URL. URLs could be logged or leaked via the Referrer header.	Critical
28	SQL Statement In HTML Comment	An SQL Statement is found in a webpage. A hacker may use this information to obtain knowledge about your web application. If your website has other database related vulnerabilities like SQL Injection, the information can be very helpful to the hacker to gain access to your database.	Medium
29	Internal IP Address Disclosure	Subtle data may be used by an attacker to exploit the target hosting network, web application, or its users.	Low
30	Possible Backup File(s) Detected	A possible backup file has been found on your web server. These files are usually created by developers to backup their work or by administrators when making backups of the web server.	Medium

31	Possible Sensitive Directories/ Files Detected	These directory/files may expose sensitive information that could help a malicious user to prepare more advanced attacks. A possible sensitive directory has been found. These directory/files are not directly linked from the website. This check looks for common sensitive resources like backup directories, database dumps, administration pages, temporary directories. Each one of these directories could help an attacker to learn more about his target.	Medium
32	Local File Inclusion (LFI)	This script is possibly vulnerable to file inclusion attacks. It seems that this script includes a file which name is determined using user-supplied data. This data is not properly validated before being passed to the include function.	High
33	Permissive Crossdomain Policy File Detected	Permissive crossdomain.xml policy files allow external scripts to interact with your website. Depending on how authorization is restricted on your website, this could inadvertently expose data to other domains or allow invocation of functionality across domains. The cross-domain policy file should permit only domains that can be trusted to make requests that include the user's domain-specific cookies. See Cross-domain policy file usage recommendations for Flash Player	Low
34	Readable .htaccess File Detected	Hypertext Access, commonly shortened to htaccess, The htaccess file is a configuration file which is used on Apache based web servers to control many features of the server. It controls the directory it is placed in and all the subdirectories underneath it.	Medium

35	TLS/SSL Server Certificate Will Expire Soon	SSL Certificate is about to expire. However, the communication will be still encrypted, but the trust mechanism will be undermined. Most importantly, users will get ugly warning messages about the security of the site and they won't make informed judgements about the integrity of the connection which will result user leaving the site.	High
36	Web Server Info Disclosure	HTTP web server information is disclosed in HTTP headers. This information may revel software name, version etc. It may help an attacker to look for specific web server version related vulnerabilities.	Info
37	Robots.txt File Detected	Website owners use robots.txt file to give instructions about their site to web robots. Robots.txt file It is robot exclusion standard to prevent robots from accessing parts of website. Robots.txt file found is not vulnerability but it displays information about site web directory which may help an attacker to launch more sophisticated attacks.	Info
38	ASP.NET ViewState MAC Disabled	ViewState is one of the most important aspects of ASP.NET WebForms applications. ViewState is a technique for storing changes in dynamic web pages during user interaction with the application server. A view-state MAC is an encrypted version of the hidden variable that a page's view state is persisted to when the page is sent to the browser. With disabled message authentication code (MAC) applied to the VIEWSTATE and allows attackers to tamper the viewstate data.	Low
39	Programming Language And Version Information Disclosure	Programming language information is disclosed in HTTP headers. This information may reveal framework and version etc. It may help an attacker to look for specific version related vulnerabilities.	Info
40	HTML Injection	HTML injection attack is similar to Cross-site Scripting (XSS). In XSS vulnerability attacker is able to execute the injected	High

		javascript code while HTML injection allows only few tags to injected. If a user input is not handled correctly then valid HTML code will get rendered and injected in to the application which results in this vulnerability. Once this is exploited it can further be used by attacker to perform other attacks.	

41	Predictable Resource Location	Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.	Info
42	Insecure Content Security Policy (CSP)/X-FrameOptions	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. Setting the right values to X-Frame-Options and/or Content-Security-Policy headers will help to protect against Clickjacking.	Medium
43	Session ID In URL	Storing the HTTP session information in URL is a highly insecure practice and leaves the HTTP session information open to theft through packet sniffing or observation of proxy logs.	Medium
44	HTTP Host Header Injection	Host header is used by a web server to decide which website should process the received HTTP request. So whenever multiple websites are hosted on the same IP address, web server uses the value of this header to forward the HTTP request to the correct website for processing. If the application relies on the value of the Host header for writing links without HTMLencoding, importing scripts, deciding the location to redirect to or even generate password resets links with its value without proper filtering, validation and sanitization then it can lead to several vulnerabilities like Cache Poisoning, Cross Site Scripting etc.	Medium

45	Unencoded special characters	Unencoded characters is deficiency or bug which allows user to inject unsafe characters which alters HTML output and can generate other security vulnerabilities like XSS and HTML injection.	Low
46	Cross-Origin Resource Sharing (CORS)	The HTML5 cross-origin resource sharing policy controls whether and how content running on other domains can perform twoway interaction with the domain which publishes the policy. The policy is fine-grained and can apply access controls per-request based on the URL and other features of the request. If another domain is allowed by the policy, then that domain can potentially attack users of the application. If a user is logged in to the application, and visits a domain allowed by the policy, then any malicious content running on that domain can potentially retrieve content from the application, and sometimes carry out actions within the security context of the logged in user. Even if an allowed domain is not overtly malicious in itself, security vulnerabilities within that domain could potentially be leveraged by a thirdparty attacker to	Low

		exploit the trust relationship and attack the application which allows access.	
47	Missing Account Lockout Policy	Multiple unsuccessful login attempts with invalid passwords is suspicious behaviour as it may be caused by brute force password guessing attacks which are intend to steal sensitive information, get access to administrative panels to perform unauthorized transactions or assisting to perform further attacks. To mitigate this issue, account lockout mechanisms are used and such locked out accounts can only be unlocked after a predetermined period of time, via a self-service unlock mechanism, or intervention by an administrator.	High
48	HTTP Verb Tampering	HTTP Verb Tampering vulnerability allows an attacker to bypass authorization by manipulating HTTP method in the request. Also any arbitrary method can be tried for the same which may lead to other attacks once executed successfully.	Low

49	Old SSL/TLS Version Detected	If the connection to site is made using old SSL/TLS versions like SSLv3, TLSv1 & TLSv1.1 which are deprecated, then connection is prone to vulnerabilities like BEAST, POODLE, etc. Usage of old SSL/TLS version often results in information leakage and other attacks.	Medium
50	Database Error Message	The error message may disclose sensitive information and this information can be used by an attacker to mount new attacks or to enlarge the attack surface. In rare conditions this may be a clue for an SQL injection vulnerability.	Medium
51	CVS Web Repository Disclosure	CVS Web Repository was found on this webpage. The CVS directory is a special directory. CVS/Entries lists files and subdirectories registered into the server. CVS/Repository contains the path to the corresponding directory in the repository. CVS/Root contains the path to the repository.	Medium
52	Web Admin Homepage (webadmin. php) Script	webadmin.php is a simple Web-based file manager. This file manager should not be installed on production systems because it does not employ any user authentication in the default configuration. Therefore an attacker can read and create random files in your system.	High
53	AWS Metadata Server Side Request Forgery	Server Side Request Forgery known as SSRF is vulnerability which allows an attacker to access aws metadata from the instances hosted upon by querying in the url. Once exploited attacker would have access sensitive information like secret key, tokens etc.	High
54	X-XSSProtection Header Disabled	The X-XSS-Protection header is designed to prevent Cross-Site Scripting (XSS) vulnerabilities built into modern web browsers. It is supported by Internet Explorer 8+, Chrome, Safari, Opera and Android. This is usually enabled by default and it can be disabled by using the HTTP Header "X-XSS-Protection: 0". Websites would be at risk with disabled X-XSS-Protection header.	Low

55	Suspicious HTML Comments Detected	Comments embedded in HTML pages may disclose sensitive information like user credentials, connection strings, sensitive file locations, etc. can lead to internal system level details being revealed to the client. Such information can be used by the attacker to conduct fatal attacks.	Low

56	User Controllable HTML Attribute	HTML attributes provide additional information about HTML elements and are generally in the form of name/value pair. There are many techniques which could use HTML attributes to submit HTML content. Using untrusted, user-controlled or attacker-controlled input in such attributes of a sensitive HTML tag and successful submission can cause XSS or HTML injection vulnerabilities.	Medium
57	Form Action Hijacking	Form action hijacking allows an attacker to specify the action URL of a form via a parameter. An attacker can construct a URL that will modify the action URL of a form to point to the attacker's server. Form content including CSRF tokens, user entered parameter values, and any other of the forms content will be delivered to the attacker via the hijacked action URL.	Medium
58	Insecure Flash Parameter "AllowScript Access" Detected	The AllowScriptAccess parameter controls whether ActionScript in a .swf flash file can perform outbound scripting actions, such as calling JavaScript in the HTML page containing the Flash object. This parameter is set inside the PARAM or EMBED tag. When it is set to "always," the SWF file can communicate with the HTML page in which it is embedded even when the SWF file is from a different domain than the HTML page. That is, an attacker can execute arbitrary JavaScript in a user's browser session and it could allow to conduct crossdomain scripting attacks.	Medium
59	Web Server Default Web Page Detected	Default configuration of web servers disclose sensitive information about their platform, version in HTTP headers and on error pages, etc. Successful exploitation will allow remote attackers to obtain such sensitive information that could aid in further attacks.	Medium

60	HTTP OPTIONS Method Enabled	The OPTIONS method provides a list of the methods that are supported by the web server, it represents a request for information about the communication options available on the request/response chain identified by the Request-URI. It may expose sensitive information that may help an malicious user to prepare more advanced attacks.	Low
61	SSL Certificate Common Name Mismatch	SSL Certificate Common Name mismatch error occurs when there is a mismatch between the domain name and the Subject Alternative Name (SAN) or common name of the SSL certificate. SAN allows you to list multiple domain names and subdomain names in a single certificate. This mismatch error occurs when the SSL certificate does not have the name entered in the browser address bar. The name mismatch error indicates that the common name (domain name) in the SSL certificate doesn't match the address that is in the address bar of the browser. This may cause a misconfiguration, or an attacker intercepting your connection or steal information.	Critical

62	SSL Certificate Signed Using Weak Signature Algorithm	The server responded with a certificate which is part of certificate chain that is signed using a weak signature algorithm (MD2, MD4, MD5, or SHA1) which are known to be vulnerable to collision attacks. Successful exploitation allows an attacker to conduct phishing attacks or to impersonate legitimate sites by taking advantage of malicious certificates.	Medium

63	SSL Certificate Using Weak Public Key	SSL certificates signed using RSA keys less than 2048 bits are considered weak, as they are increasingly vulnerable to being broken in a reasonable time-frame. A successful attack of this nature would provide an attacker with clear text access to encrypted data as it	High
64	Apache Struts2 Developme nt Mode Enabled	Apache Struts 2 has a setting (which can be set to true or false in struts.properties) called devMode (= development mode). When this setting is enabled, Struts 2 will provide additional logging and debug information, which can significantly speed up development. An attacker can gain potential information which will assist in conducting further attacks and there is a known risk of arbitrary Java (OGNL) code execution.	Medium

65	Apache serverstatus Enabled	Apache has a functionality called server-status that allows administrators to easily find how well their servers are performing and its enabled via a Module mod_status. Its an HTML page is presented that gives the current server statistics in an easily readable form. Information disclosed such as Server uptime, Individual request-response statistics and CPU usage of the working processes, Current HTTP requests, client IP addresses, requested paths, processed virtual hosts, could give a potential attacker information about how to attack the web server.	Medium
66	ASP.NET Tracing Enabled	ASP.NET tracing enables to view diagnostic information related to a specific web page or application that is being executed on the web server. This information like session ID, execution path, etc. helps to investigate errors or unwanted results while ASP.NET processes a page request. Disclosing such sensitive information may allow users to conduct attacks.	Medium
67	ASP.NET Version Disclosure	ASP.NET version information is disclosed by the web server via HTTP response header. Successful information disclosure allows attackers to conduct specific vulnerabilities based on the identified versions.	Info
68	Browser Cache Enabled	Caching web application data may result in exposure of URL histories, HTTP headers, HTML form inputs, cookies, transaction history and other such web-based data easily being revealed via response browser cache headers. Successful disclosure of such sensitive information allows remote attackers to conduct attacks in conjunction with other vulnerabilities.	Low
69	Hidden Form Input "Price" Detected	Hidden form inputs are often written into an HTML page by the web server when it serves that page to the client and are not visible on the rendered web page. Web applications can use hidden form inputs to remember session data and allows remote users to alter the values to their benefit and resubmit to the application. A hidden form input called	Medium

70	Possible Web Form Spam Detected	Poorly written scripts in Web forms may allow the application to send spam messages. A hidden form input with an email address as value has been detected in a web application which will allow remote users to distribute emails across the Internet with server as the identifiable source of the spam.	Medium
71	Documentation File Detected	An application documentation file like readme.txt, changelog.txt, etc. may contain sensitive information like application name, version, user details etc. Successful disclosure of such documentation file allows attackers to exploit vulnerabilities based on the identified application details.	Low

72	Possible Slow Response Time Detected	Server response time is the amount of time required to load the HTML page of an application from a server so that the client (browser) can begin rendering the page. Without a good server response time, the HTML page will take longer to load. If the HTML page is not loaded, then browser won't know what other resources will be required in order to display the page properly. Web pages with slow response time can be targeted to be used in conducting DOS attacks to overload the servers and may result in an unresponsive application.	Low
73	Microsoft IIS Version Disclosure	Web Server IIS sets response headers that reveal its version information in default configurations. Successful version disclosure can assist a user to conduct further attacks by targeting vulnerabilities specific to application version identified.	Info
74	HTML Form Found In Redirect Page	An HTML form in a redirect page which does not terminate the response can let users to bypass authentication and provide access to sensitive information.	Low
75	Server Side Request Forgery Local File Inclusion	Server Side Request Forgery known as SSRF is vulnerability which allows an attacker to perform local file inclusion by querying in the url. Once exploited attacker would have access sensitive information like passwords, user groups, etc.	High

76	Unset/Insecure HSTS header	Adding HTTP Strict-Transport-Security (HSTS) response header enable web sites to declare themselves accessible only via secure connections and/or for users to be able to direct their user agent(s) to interact with given sites only over secure connections. Missing HSTS header allows remote attackers to conduct man-in-the-middle attacks and steal private data.	Medium
77	Cookie Scoped To Parent Domain	A cookie scoped to the parent domain will be available to all subdomains therefore increasing the chance of leakage. This may occur when the information is transmitted unencrypted or when a XSS vulnerability affected a subdomain is in place.	Low
78	WordPress XML-RPC Interface Detected	XML-RPC is a remote procedure call (RPC) protocol which uses XML to encode its calls and HTTP as a transport mechanism. "XML-RPC"	Medium
79	Improper Token Handling - Duplicate	Token based transactions are used to store the user state on the client and the user data is encrypted into a token with a secret and then sent back to the client. Improper handling of tokens like not generating unique per user session or improper validation before accepting and executing it may allow remote attackers to bypass authentication mechanisms, CSRF etc.	Medium
80	Apache Tomcat Remote Code Execution Vulnerabilit y (CVE20190232)	A vulnerability in the CGI Servlet of Apache Tomcat could allow an unauthenticated, remote attacker to execute arbitrary code on a targeted system. The vulnerability occurs when enableCmdLineArguments is enabled on a Windows system and the Java Runtime Environment (JRE) passes command-line arguments to the system. An attacker could exploit this vulnerability by passing command-line arguments to the affected system. A successful exploit could allow the attacker to execute code on the targeted system.	High
81	Credit Card Number Disclosure	This is intended to detect sensitive and financial information such as Credit Card Number in HTTP responses. Credit Card Number disclosure may allow remote attackers to steal other financial information and make unknown transactions.	Medium

82	Oracle WebLogic Server Deserializati on Remote Command Execution Vulnerabilit y (CVE20192725)	Oracle WebLogic servers includes wls9_async_response.war and wls-wsat.war packages by default which provides asynchronous communication for WebLogic Server service. These WAR packages can be misused when deserializing input information and an attacker can send a constructed malicious HTTP request to gain the permissions of the target server and execute the command remotely without authorization	Critical
83	Dot Net Insecure Deserializati on Remote Command Execution Vulnerabilit y	Dot Net Insecure Deserialization triggers when an attacker abuses deserialization features when the application is deserializing untrusted data which the user controls. Successful insecure deserialization attacks could allow an attacker to carry out denial-of-service (DoS) , authentication bypasses and remote code execution attacks.	Critical
84	Perl Deserializati on Remote Command Execution Vulnerabilit y	Perl Insecure Deserialization triggers when an attacker abuses deserialization features when the application is deserializing untrusted data which the user controls. Successful insecure deserialization attacks could allow attacker to perm authentication bypasses, denial-of-service(DOS) and remote code execution attacks.	High
85	Passive Mixed Content Vulnerabilit y	Passive/display content is content served over HTTP that is included in an HTTPS webpage, but that cannot alter other portions of the webpage. For instance, an HTTPS page which loads an image over HTTP. This allows an attacker to replace an image served over HTTP with an inappropriate image or message to the user, tampering page, etc.	Medium
86	Active Mixed Content	Mixed active content is content which loads script file including scripts, stylesheets, iframes, flash resources, or other code via HTTP that can alter the behaviour of the HTTPS page. This allows attackers to change anything about the page, including	Medium

	Vulnerabilit y	displaying entirely different content, stealing user passwords or other login credentials, stealing user session cookies, or redirecting the user to a different site entirely, even rewrite the response to include malicious JavaScript code.	

87	PHP Deserializati on Remote Command Execution Vulnerabilit y (CVE201717672)	PHP Deserialization triggers when an attacker abuses unauthenticated deserialization that leads to arbitrary file deletion or code execution, because of unsafe usage of PHP's unserialize() in publicly exposed API.	High
88	Ruby on Rails XML/JSON Processor YAML Deserializati on Code Execution Vulnerabilit y (CVE2013-0156)	Ruby Deserialization RCE vulnerability in the XML request processor vulnerability allows an attacker to instantiate a remote object, which in turn can be used to execute any ruby code remotely in the context of the application & can compromise the system with authentication bypass or DenialOf-Service attacks. This has been tested against 3.x & 2.x versions of RoR which are vulnerable.	High
89	Oracle WebLogic Server Deserializati on Remote Command Execution Bypass Vulnerabilit y (CVE2019-2729)	A vulnerability in the Web Services component of Oracle WebLogic Server could allow an unauthenticated, remote malicious user to execute arbitrary code on a targeted system. The vulnerability is due to a deserialization condition that exists when the affected software uses the XMLDecoder class. An attacker could exploit this vulnerability by sending a request that submits malicious input to the targeted system. A successful exploit could allow the malicious user to execute arbitrary code, which could be used to conduct further attacks	Critical
90	Possible Archive File or Compressio n File (s) Detected	A possible archive or compression file has been found on your web server directory which are usually created by developers/administrators to collect multiple data files together into a single file for easier portability and storage, or simply to compress files to use less storage space or backup purpose.	Low

91	Cookie Overly Broad Path Detected	The cookie 'path' attribute signifies the URL or path for which the cookie is valid. If an overly broad path like root '/' is specified in the cookie then it is accessible through other applications on the same domain. Exposing the cookie to all web applications on the domain can lead to sensitive information disclosure like session identifier, etc. and can cause one application to compromise another application.	Low
92	Session Cookie Manipulation	Cookie is piece of information sent by a web server to store on a web browser which stores some specific personal information. If misconfigured then it can lead to dangerous vulnerabilities such as xss, sql, session fixation etc.	Medium
93	Weak Session IDs	The cookie 'session-ids' attribute signifies the authentication of the user. If it's weak and predictable, then it may cause for session hijacking attacks where attacker and impersonate as authentic user and use application in malicious way.	Medium
94	HTTP TRACK Method Enabled	The HTTP TRACK method is normally used to return the full HTTP request back to the requesting client for proxy-debugging purposes. An attacker can create a webpage using XMLHTTP, ActiveX, or XMLDOM to cause a client to issue a TRACK request and capture the client's cookies. This effectively results in a CrossSite Scripting attack.	Low
95	Log Injection	Logs are a source of information that can be used for debugging, data collection and performance optimizations. Injecting in the logs allows attacker to insert malicious data and false entries into the logs and ultimately corrupt the file or use it for other penetration attempts	Medium

96	HTML Form Without Anti-CSRF Token Detected	Cross-Site Request Forgery (CSRF/XSRF) is a vulnerability where an attacker tricks the victim into making a request victim did not make. So, the attacker abuses the trust a web application has with a victim's browser. Mostly the HTML forms submitted have CSRF tokens embedded in them while submitting the request. If a form is without this preventive measure enabled then it's very much prone to CSRF attacks and other dependent attacks. We are checking for the existence of the known list of CSRF tokens, if none of the tokens are found the target will be flagged.	Medium
97	Web Administrati on Login Page Detected	An applications can be configured & controlled by an administrators who can access the admin panel through a login page or administration (admin) pages. A remote attacker can target such admin pages available in public to gain admin access of an application or compromise the sites via bruteforce attacks, SQL injection, etc.	Low
98	Web Server Content Sniffing Enabled	Content Sniffing is a technique used by the browsers to determine an asset	Medium
99	Apache Range Denial Of Service	Apache httpd server has denial of service vulnerability in few of their versions. This exists due to the range header that expresses multiple overlapping ranges. And this byte filter range allows remote attacker to cause a DOS attack resulting in memory and cpu consumption.	Medium
100	VBulletin Pre-Auth RCE Vulnerabilit y	vBulletin is a software for running forums on your website. A pre-authentication remote code execution vulnerability exists in this which allows attacker to execute commands and compromise your systems	Critical

101	Server Side Javascript Injection	Server-side JavasScript injection vulnerability arises when an application uses user-controllable data into a string that is processed by a server. An attacker can abuse such functionality to inject malicious code and in turn use system in malicious way.	Critical
102	Server-Side Template Injection	Server-side Template injection vulnerability arises when an application uses user-controllable data added to server side template which is then processed by template engine. An attacker can abuse such functionality to inject template directives/code and execute arbitrary code in system in-turn compromising it.	Critical
103	Remote XSL Inclusion	XSL (Extensible Stylesheet Language) is used to refer to a family of languages used to transform and render XML documents. The script/site is vulnerable to remote XSL inclusion when targeted XSL file is in control of attacker which will be pretty much malicious file. Once site successfully executed XSL file, it then in turn can be used to execute malicious code and compromise system with various other attacks.	High
104	PHP Nginx Remote Command Execution	This vulnerability is an extension to OS Command Injection where php sites hosted on ngnix servers are vulnerable to remote command execution. Once attacker gains successful rce, it can be used further to compromise the system or use it in malicious way	Critical
105	Default And Common Credentials Detected	Commonly used username and/or passwords combinations that are valid regardless of the type of application are called as Common credentials. Similarly known usernames and password combinations associated with a specific applications are called as Default credentials. A remote attacker can exploit these issues to gain access to the web application and take complete control of the application affecting the operation of the application and underlying system.	High

106	SQL Injection Authenticati on Bypass	Web applications with weak authentication controls & access control polices may allow remote attackers to bypass authentication by injecting crafted SQL queries during login attempts. Successful attacks result in unauthenticated, remote attackers to gain complete control of the account/admin privileges and conducts attacks further.	High
107	Login Username Enumeratio n	Web applications which fail to respond with consistent error messages when a user attempts to login with existing and nonexisting accounts can indicate the validity of the username submitted. A remote, unauthenticated attacker could use this to enumerate valid usernames, which could be used to mount further attacks.	Medium
108	Core Dump File(s) Detected	Core dump files contain an application's memory (including details are shared libraries, user's data, credentials, etc.) created by the system when a process was interrupted. Disclosing such core dump files allows remote attackers to access sensitive information of the application and assist in conducting attacks further.	Medium
109	JSF ClientSide	Java Server Faces (JSF) is a Java-based Web application framework that implements the Model-View-Controller	Medium

	ViewState Detected	pattern and simplifies the development of web interfaces for Java EE applications. If the client side viewstate is used rather than server side and it's not encrypted then it can be easily used to read the critical information and used in other attacks	
110	WAF/IPS Detected	The site/server is protected by packet filtering systems like WAF (Web Application Firewall) or IPS (Intrusion Protection System). As they filter traffic and drop/redirect the connection, our scanner will not be successful in determining exploitable environment and hence will not be able to get comprehensive list of vulnerabilities exists in the current application	Info
111	Insecure CacheControl Header Detected	Cache-Control header is used to control the behaviour of browser caches and proxy caches based on multiple directives. With max-age directive enabled, the browser may cache the page, but it must re-validate with the server when its value is	Low

		exceeded. Setting max-age to zero ensures that a page is never served from cache, but is always re-validated against the server. Thus reduces the performance of the server as it increases load.	
112	Old Cipher Suites Detected	SSL connection to the site is made using old ciphers and this are considered weak in the current time. These ciphers can be decoded to reveal the information and could lead to other potential attacks and vulnerabilities like SWEET32.	Medium
113	Insecure/Deprecated Cryptography Detected	Usage of a weak/deprecated hashing/crypto function has been detected in the site. It can be sniffed and easily decrypted to obtain sensitive information and conducting further attacks.	Medium
114	Apache Axis2 Local File Inclusion Vulnerability	Local File Inclusion (LFI) vulnerability in the Apache Axis2 service allows remote attackers to access arbitrary/sensitive files which are normally inaccessible. By sending a crafted request using xsd parameter, attacker can obtain the file requested which contains sensitive information which is further used to perform other attacks.	High
115	JSMOL2 Server Side Request Forgery Local File Inclusion	JSMOL2 Server Side Request Forgery known as SSRF is vulnerability which allows an attacker to perform local file inclusion by attacking the url with specific payload. Once exploited attacker would have access sensitive information like database usernames, passwords, etc.	High
116	Long Password Denial Of Service	A flaw in the password hashing process of a web server could exhaust memory and CPU leading to a denial of service and the website becoming unresponsive. Remote attackers can exploit it by sending a very long password which is improperly handled during hashing leading to Denial of Service.	High
117	Uncontrolled Format String	The vulnerability allows an attacker to read the stack trace or execute code or cause segmentation faults by attacking with format string like %f, %s which application/server uses externally.	Medium

		Attacker can read/access other memory spaces with such attacks.	
118	Google Chrome Logger	Chrome Logger is a Google Chrome extension for debugging server side applications in the Chrome console. It uses an HTTP header to send log data from the application server to the web	Low
	Information Disclosure	browser. Such log data can carry sensitive information to debug the server-side code which can be used by the remote attackers to conduct attacks further.	
119	Java Virtual Machine (JVM) Version Disclosure	JVM (Java Virtual Machine) is virtual machine that provides runtime environment in which java bytecode can be executed. JVM version information can be disclosed via a server header and enables remote attackers to conduct version specific attacks.	Low
120	Missing Subresource Integrity Check	Subresource Integrity (SRI) refers to security feature which helps browsers to make sure that 3rd party resources fetched either from a CDN or other source is not tampered with and integrity of resource is intact. This is verified by cryptographic hash value provided to that file when it fetched. Missing SRI implementation enables attackers to gain control of a CDN, can inject arbitrary malicious content into files on the CDN (or replace the files completely) and thus can also potentially attack all sites that fetch files from that CDN.	Info
121	HTTP Request Smuggling	HTTP Request Smuggling is a type of attack where specially crafted HTTP messages can be parsed and interpreted in different ways depending on the technology/agent that receives them. Leveraging this an attacker can bypass security controls, firewall checks and many more.	High

122	Possible Slowloris DOS Attack	Slowloris is a type of Denial of Service (DOS) attack tool that causes DOS by sending a very slow partial HTTP header requests. The tool sends simultaneous, multiple partial HTTP header requests slowly to force the target web server to keep the connections open and continue to wait for the end of header request. The serverâ€™s all resources is consumed completely by such open connections and soon legitimate requests will not be handled by the server. The plugin is intend to detect such slow-rate DOS attacks possible against the web server with slow header requests.	Medium
123	Link Injection	Link injection is a type of HTML/XSS Injection attack. The attack happens by inserting an link tags in to page/site content which may be used further for phising, redirection to malicious sites, credential stuffing, etc. kind of attacks.	Medium
124	Iframe Injection	Iframe injection is a type of HTML/XSS Injection attack. The attack happens by inserting iframe tags with ability to load third party links in to page/site content which may be used further for phishing, credential stuffing, backdoor download etc. kind of attacks.	Medium
125	XML External Entity DOS Attack	An XML External Entity (XXE) is a parameter parsed entity that can access local or remote content via a declared system identifier which is assumed to be a URI that can be accessed by the XML processor when processing the entity. An attacker can leverage the same features to endure heavy load on servers by expansion of entity and thus creating a DOS attack.	High
126	XML External Entity (XXE) Injection	An XML External Entity (XXE) is a parameter parsed entity that can access local or remote content via a declared system identifier which is assumed to be a URI that can be accessed by the XML processor when processing the entity. An XML input	High

	Vulnerabilit y	containing a reference to an external entity processed by a weakly configured XML parser can lead to disclosure of confidential data, denial of service, server-side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.	

127	oracle weblogic uri attack	Vulnerability identified in the oracle web logic server enables attacker to abuse uri path in order to gain control of server, get elevated privileges before the server's authentication kicks in, This can result in RCE, XSS, sensitive information exposure, data exfiltration to/from backend network.	High
128	web cache poisoning attack	Web Cache Poisoning is an attack against the integrity of web cache repository, users of the web cache repository will thus consume spoofed content instead of a genuine one. Combined with injection attacks such as XSS, This can lead to sensitive data exposure, XSS , cookie stealing, session hijack.	High
129	Microsoft Exchange Server Remote Code Execution Vulnerabilit y	The identified vulnerability is a SSRF flaw in Exchange servers. A remote attacker can send arbitrary HTTP requests to bypass authentication and abuse functionality on the exchange server to read or update internal resources, remotely execute arbitrary code and exploit various post authentication vulnerability as part of attack chain.	High
130	Possible BREACH (Browser Reconnaissa nce & Exfiltration via Adaptive Compressio n of Hypertext) Vulnerabilit y	The identified vulnerability is prone to MITM attack abusing flaw in http level compression. Exploit of such vulnerability is agnostic to the version of SSL/TLS protocol in use, as long as provided conditions are met. An MITM attacker can brute force secret contents from protected SSL/TLS traffic such as CSRF tokens, credit card numbers and execute attacks like XSS, CSRF or session hijacking.	Medium
131	HTTP.sys Remote Code Execution Vulnerabilit y	The identified vulnerability is a flaw in HTTP protocol stack 'HTTP.sys' that is triggered when parsing HTTP requests. Successful Exploitation can allow remote attacker to cause blue screen of death, crash, restart server.	High

132	GNU glibc Remote Heap Buffer Overflow Vulnerabilit y (CVE20150235)	The identified vulnerability is a buffer overflow flaw in a linux glibc library. The vulnerability is called GHOST because it is triggered by the GetHost function of glib which can be abused via vulnerable version of php. An attacker can exploit this vulnerability to perform remote code execution, restart or crash server, install backdoor and also gain complete control of system. It is recommended to either install the patch or update php to latest patched version.	Medium
133	Partial user controllable	HTML attributes provide additional information about HTML elements and are generally in the form of name/value pair.	Medium

	script source	Script src attribute specifies the URL of external JavaScript file. The attribute is user-controlled and exploiting it can cause XSS, Reverse Clickjacking attacks and exposes to other security issues.	

134	Incorrect Session Timeout	The Timeout property specifies the time-out period assigned to the Session object for the application, in minutes. If the user does not refresh or request a page within the time-out period, the session ends. It was observed that the application doesn't terminate the session automatically even if the user session is inactive for a prolonged period after login. An attacker can use the compromised/leaked token to perform unauthorized and unintended activities on behalf of the user pretending to be the legitimate.	Medium
135	JWT misconfigur ation	JSON Web Token (JWT) is a standard for creating tokens that assert some number of claims. For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client. The client could then use that token to prove that they are logged in as admin. An attacker can take over the account of the victim if authorization tokens are not configured properly.	Medium

136	Permissive Client Access Policy File Detected	ClientAccessPolicy.xml file grants cross-domain permissions for reading data. It is used to provide the access to cross domain to obtain any document on the server. Misconfiguration in file may lead to access on protected areas and can be used to trigger other attacks.	Low
137	Weak TLS CBC cipher Detected	TLS connection to the site is made using CBC ciphers and this are considered weak in the current time. Server uses TLS 1.2 or TLS 1.1 or TLS 1.0 with CBC cipher, it could lead Zombie POODLE, GOLDENDOODLE, 0-Length OpenSSL and Sleeping POODLE vulnerabilities.	Medium
138	Possible Archive File or Compressio n File-log	Web applications that do not properly sanitize user input before passing it to a database system are vulnerable to SQL injection. This type of attack potentially allows a malicious user to recover and/or modify any data that the application has access to.	Medium
139	LFI in Apache mod-cgi	Local File Inclusion (LFI) vulnerability in the event mod-cgi enabled on Apache 2.4.49 allows remote attackers to access arbitrary/sensitive files which are normally inaccessible. By sending a crafted request, attacker can obtain the file requested which contains sensitive information which is further used to perform other attacks.	High
140	Code Injection	Code injection vulnerability occurs when a developer uses invalidated user controlled parameters to interpreted/executed by the application. Impact could cover loss of confidentiality, loss of integrity, loss of availability, and/or loss of accountability	High
141	Cross-Site Flashing (XSF)	Cross-site flashing occurs when user-controlled data is not validated and is used in the following functions or variables: loadVariables, loadMovie, getURL, loadMovieNum, FScrollPane.loadScrollContent, Sound.loadSound, NetStream.play, flash.external.ExternalInterface.call and	High

		htmlText. In other words, the Flash application must reference external URLs, and the locations of those URLs are set by userdefined parameters (usually Flash Vars).	

142	Edge Side Include Injection	Edge Side Include injection vulnerability arises when an application uses user-controllable data and reflect the ESI tags. It can lead to Server Side Request Forgery (SSRF) in the context of the surrogate server, various Cross-Site Scripting (XSS) vectors that bypass the HTTPOnly cookie mitigation flag, and serverside denial of service.	High
143	Apache Log4j RCE Vulnerabilit y	CVE-2021-44228 is a remote code execution (RCE) vulnerability in Apache Log4j 2.0 through 2.14.1. An attacker can exploit this vulnerability by sending a crafted request to the vulnerable server. This can be done by submitting an exploit string on the text field found in the website running on the vulnerable server or by including the exploit string as part of the header destined to the vulnerable server.	Critical
144	Apache Server ETag Header Information Disclosure	Apache server ETag (entity tag) response header field provides sensitive information of the inode number of requested files. It cause information disclosure and cache poisoning attack vulnerability.	Medium
145	Improper Session Manageme nt	Proper authentication and session management is critical to web application security. Flaws in this area frequently involve the failure to protect credentials and session tokens through their lifecycle. These flaws can lead to the hijacking of user or administrative accounts, undermine authorization and accountability controls, and cause privacy violations.	Medium
146	Insecure Direct Object References	Applications frequently use the actual name or key of an object when generating web pages. Applications donâ€™t always verify the user is authorized for the target object. This results in an insecure direct object reference flaw. Testers can easily manipulate parameter values to detect such flaws and code analysis quickly shows whether authorization is properly verified.	High
147	OS Command Injection - OOB	An OS command injection vulnerability occurs when a developer uses invalidated user controlled parameters to execute operating system commands. OS command injection vulnerabilities allow attackers to run arbitrary commands on the remote server. This is one of the flaws under the category of Code Injection, in the OWASP Top Ten.	Critical

148	XML External Entity (XXE) Injection Vulnerabilit y - OOB	An XML External Entity (XXE) is a parameter parsed entity that can access local or remote content via a declared system identifier which is assumed to be a URI that can be accessed by the XML processor when processing the entity. An XML input containing a reference to an external entity processed by a weakly configured XML parser can lead to disclosure of confidential data, denial of service, server-side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.	High

149	SQL Injection OOB	Web applications that do not properly sanitize user input before passing it to a database system are vulnerable to SQL injection. This type of attack potentially allows a malicious user to recover and/or modify any data that the application has access to.	Critical
150	Server-Side Request Forgery (SSRF) - OOB	Server-side request forgery (also known as SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make HTTP requests to an arbitrary domain of the attacker's choice. An attacker can trick the web server that could probably be running behind a firewall to send requests to itself to identify services running on it, or can even send outbound traffic to other servers.	Medium

151	Cross Site Scripting (XSS) OOB	The Web application is vulnerable to cross-site scripting (XSS), which allows attackers to take advantage of Web server scripts to inject JavaScript or HTML code that is executed on the clientside browser. This vulnerability is often caused by serverside scripts written in languages such as PHP, ASP, .NET, Perl or Java, which do not adequately filter data sent along with page requests or by vulnerable HTTP servers. This malicious code appears to come from your Web application when it runs in the browser of an unsuspecting user. An attacker can do the following damage with an exploit script: access other sites inside another client's private intranet steal another client's cookie(s) modify another client's cookie(s) steal another client's submitted form data modify another client's submitted form data before it reaches the server submit a form to your Web application on the user's behalf that modifies passwords or other application data The two most common methods of attack are: Having a user click a URL link sent in an e-mail Having a user click a URL link while visiting a Web site	High
		In both scenarios, the URL will generally link to the trusted site, but will contain additional data that is used to trigger the XSS attack. Note that SSL connectivity does not protect against this issue.	

152	HTTP Host Header Injection OOB	Host header is used by a web server to decide which website should process the received HTTP request. So whenever multiple websites are hosted on the same IP address, web server uses the value of this header to forward the HTTP request to the correct website for processing. If the application relies on the value of the Host header for writing links without HTML-encoding, importing scripts, deciding the location to redirect to or even generate password resets links with its value without proper filtering, validation and sanitization then it can lead to several vulnerabilities like Cache Poisoning, Cross Site Scripting, Routing-based-ssrf etc.	Medium
153	Server-side template injection OOB	Server-side Template injection vulnerability arises when an application uses user-controllable data added to server side template which is then processed by template engine. An attacker can abuse such functionality to inject template directives/code and execute arbitrary code in system in-turn compromising it.	Critical
154	Spring Expression Resource Access Vulnerabilit y (RCE)	In Spring Cloud Function, when using routing functionality it is possible for a user to provide a specially crafted Spring Expression Language (SpEL) as a routing-expression that may result in access to local resources and/or execute	High
155	Code Injection - OOB	Code injection vulnerability occurs when a developer uses invalidated user controlled parameters to interpreted/executed by the application. Impact could cover loss of confidentiality, loss of integrity, loss of availability, and/or loss of accountability	Critical

156	VMware Server-side Template Injection (RCE) Vulnerabilit y (CVE202222954)	VMware Workspace ONE Access and Identity Manager contain a remote code execution vulnerability due to serverside template injection. A malicious actor with network access can trigger a server-side template injection that may result in remote code execution.	Critical
157	Password found in server response	Password found in clear form in response. An attacker can be able to capture the password using weaknesses in session handling, broken access controls, and cross-site scripting vulnerability. It could lead an attacker to quickly compromise the entire application.	Medium
158	Credential found in token	The HTTP Authorization token has confidential data that assert some number of claims. For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client. The client could then use that	Medium
		token to prove that they are logged in as admin. An attacker can take over the	

		account of the victim if authorization tokens are not configured properly.	
159	Link Injection OOB	Link injection is a type of HTML/XSS Injection attack. The attack happens by inserting an link tags in to page/site content which may be used further for phishing, redirection to malicious sites, credential stuffing, etc. kind of attacks.	Medium
160	Path- Relative StyleSheet Import Vulnerabilit y	The Path-Relative StyleSheet Import vulnerability occurs when HTML uses path-relative CSS links and may not determine the correct directory. It could lead to cross-site scripting (XSS) and exfiltration of CSRF tokens.	Low

161	Iframe Injection OOB	Iframe injection is a type of HTML/XSS Injection attack. The attack happens by inserting iframe tags with ability to load third party links in to page/site content which may be used further for phishing, credential stuffing, backdoor download etc. kind of attacks.	Medium
162	Improper Token Handling	Token based transactions are used to store the user state on the client and the user data is encrypted into a token with a secret and then sent back to the client. Improper handling of tokens like not generating unique per user session or improper validation before accepting and executing it may allow remote attackers to bypass authentication mechanisms, CSRF etc.	Medium
163	Sensitive Information Disclosure Through URL	The GET of every web page you visit is recorded in your browser history file. An attacker can steal sensitive information from the history of the browser.	Low
164	Running Service(Ope n Port)	An open port is possible leading to data loss, DOS attack and other vulnerabilities.	Info
165	Unset/Insec ure X-Permitted-CrossDomain-Policies Header	A cross-domain policy file is an XML document that grants a web client to handle data across domains. When clients request content hosted on a particular source domain and that content makes requests directed towards a domain other than its own, the remote domain needs to host a cross-domain policy file that grants access to the source domain, allowing the client to continue the transaction.	Low
166	Session Resumption Enabled	Previous TLS sessions can be resumed, allowing for a connection to be established using an abbreviated handshake. All versions of TLS offer session resumption, although the mechanism for performing resumption differs. It could leads steal existing TLS sessions and replay attacks.	Info

167	DNSSEC unsigned	DNS Security Extensions (DNSSEC) provide source authentication for the DNS. DNS to verify the authenticity of its data. It is not valid or unsigned, it could lead to DNS spoofing / malicious activity.	Info

168	Content Injection	Content spoofing, also referred to as content injection or context injection, is an attack targeting a user made possible by an injection vulnerability in a web application. When an application does not properly handle user supplied data, an attacker can supply content to a web application, typically via a parameter value, that is reflected back to the user. This presents the user with a modified page under the context of the trusted domain. This attack is typically used as, or in conjunction with, social engineering because the attack is exploiting a code-based vulnerability and a user's trust.	Medium
169	JWT none algorithm	An attacker alters the token and changes the hashing algorithm to indicate, through, the none keyword, that the integrity of the token has already been verified. Some libraries treated tokens signed with the none algorithm as a valid token with a verified signature, so an attacker can alter the token claims and token will be trusted by the application.	Medium
170	Weak Encoding	An attacker can steal sensitive information from weak encoding. Encoding is the process of putting a sequence of characters into a special format for transmission or storage purposes	Medium
171	Reveals Sensitive Information	Sensitive information in Request and Response should be encoded with proper technique with salting. eg: Password, Account Details, Personal Identity information, etc. This may lead to exposure of other vulnerabilities.	Low
172	Reveals Sensitive Information	Sensitive information in Request and Response should be encoded with proper technique with salting. eg: Password, Account Details, Personal Identity information, etc. This may lead to exposure of other vulnerabilities.	Info
173	Accessible By IP Address	A server is serving the page if accessed by the IP address. The server should not allow this as worms who scan for IP address randomly can spot the site.	Low

174	No CAPTCHA on login page	The absence of a CAPTCHA on a login page is a security vulnerability that allows for automated attacks like brute force attacks, credential stuffing attacks, and account enumeration. CAPTCHAs are a security measure that distinguishes between humans and bots. By implementing a CAPTCHA, websites add an extra layer of security and ensure that only legitimate human users can access the login page, reducing the risk of automated attacks.	Info
175	EPMM Authenticati on Bypass	Web applications with weak authentication controls & access control polices may allow remote attackers to bypass authentication. Successful attacks result in unauthenticated, remote attackers to gain complete control of the account/admin privileges and conducts attacks further.	Critical
176	WebSocket URL poisoning	Successful submission using untrusted, user-controlled, or attacker-controlled data in a WebSocket URL can cause XSS, information leakage, denial of service and unauthorized access to sensitive data.	Medium
177	BruteForce Directory/Fil e	These directory/files may expose sensitive information that could help a malicious user to prepare more advanced attacks	Medium
178	Client-side Template Injection	Client-side Template injection vulnerability arises when an application uses user-controllable data added to client side template which is then processed by template engine. An attacker can abuse such functionality to inject template directives/code and execute arbitrary JavaScript code in the victim.	Critical
179	Insecure transition from HTTP to HTTPS in form post	Insecure HTTP pages serving HTTPS forms. The problem is that an insecure HTTP page can easily be hijacked by MITM and a secure HTTPS form replaced or spoofed.	Medium
180	Insecure transition from HTTP to HTTPS in form post	Secure HTTPS pages serving insecure HTTP forms. The problem is that when data is uploaded through a form the secure page becomes an insecure page.	Low

181	Insecure Transport	HTTPS is used to secure the communication between the server and the browser. However, the problem occurs when a web application allows users to access a website via "HTTP" instead of "HTTPS" and does not automatically redirect users to HTTPS. That can lead to steal login credentials, session IDs or other sensitive information.	Medium
182	Cross-Site Tracing (XST)	A Cross-Site Tracing (XST) attack involves the use of Cross-site Scripting (XSS) and the TRACK HTTP methods. TRACE allows the client to see what is being received at the other end of the request chain and use that data for testing or diagnostic information. The TRACE method, while apparently harmless, can be successfully leveraged in some scenarios to steal legitimate users' credentials.	Medium
183	ExtJs Arbitrary File Read	Ext JS, a JavaScript framework for dynamic web apps, This flaw allows for the reading of arbitrary files and the initiation of internal HTTP service requests	High
184	Body Parameters Accepted in Query	With GET requests, there are numerous ways for sensitive information to be exposed in clear text like browser history files, Referrer header fields, server, proxy and log files.	Medium

185	PHP CGI Argument Injection Vulnerabilty	In PHP versions 8.1.* before 8.1.29, 8.2.* before 8.2.20, 8.3.* before 8.3.8, when using Apache and PHP-CGI on Windows, if the system is set up to use certain code pages, Windows may use "Best-Fit" behavior to replace characters in command line given to Win32 API functions. PHP CGI module may misinterpret those characters as PHP options, which may allow a malicious user to pass options to PHP binary being run, and thus reveal the source code of scripts, run arbitrary PHP code on the server, etc	Critical

186	Cross-Site Request Forgery (CSRF)	CSRF is an attack which forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated. With a little help of social engineering (like sending a link via email/chat), an attacker may force the users of a web application to execute actions of the attacker's choice. A successful CSRF exploit can compromise end user data and operation in case of normal user. If the targeted end user is the administrator account, this can compromise the entire web application.	Medium
187	Unsafe thirdparty link	When a page uses target="_blank" in a link, it will open the link in a new tab or window. This can be a security risk because the new page can access the original page's Window object via window.opener , which could lead to phishing attacks or other malicious activity.	Medium
188	Path Traversal vulnerability in Apache OFBiz	The vulnerability allows an attacker to access files and directories that are stored outside the web serverâ€™s root directory. If exploited, it could allow the attacker to read sensitive files from the server, modify server configurations, or even execute arbitrary code, leading to full server compromise. which could lead to Remote Code Execution (RCE). This vulnerability affects versions of Apache OFBiz before 18.12.14.	Critical
189			

190	Path Traversal vulnerability in Apache OFBiz	The vulnerability allows an attacker to access files and directories that are stored outside the web serverâ€™s root directory. If exploited, it could allow the attacker to read sensitive files from the server, modify server configurations, or even execute arbitrary code, leading to full server compromise. which could lead to Remote Code Execution (RCE). This vulnerability affects versions of Apache OFBiz before 18.12.14.	Critical
191	Unsafe thirdparty link	An unsafe third-party link vulnerability, especially when using target="_blank" without rel="noopener noreferrer", can pose significant security risks. This vulnerability allows the linked page to access the window object of the linking page, potentially leading to security issues such as cross-site scripting (XSS) attacks. This can be a security risk because the new page can access the original page's Window object via window.opener , which could lead to phishing attacks or other malicious activity.	Low
192	Database Connection String Detected	A database connection string is a critical component used by applications to connect to databases. It often contains sensitive information, such as the database type, server address, database name, and authentication credentials (username and password). If such a connection string is exposed, it could lead to unauthorized access to the database, potential data breaches, and various other security issues.	High
193	Web Server Content Sniffing Enabled	Web server content sniffing vulnerabilities occur when a web server incorrectly identifies the MIME type of a file. This can lead to security issues such as cross-site scripting (XSS) attacks, where malicious scripts are executed in the context of another userâ€™s session.Content sniffing vulnerabilities can allow attackers to execute scripts in the context of another userâ€™s session, potentially leading to data theft or unauthorized actions.	Low

194	Unset/Insecure X-XSSProtection Header Vulnerability	The X-XSS-Protection header is a security feature designed to prevent cross-site scripting (XSS) attacks by enabling the XSS filter built into modern web browsers. When this header is disabled or improperly configured, it can leave your web application vulnerable to XSS attacks, which can lead to malicious scripts being executed within your application.Potential Impacts: Increased Risk of XSS Attacks, Data Theft, Session Hijacking, Defacement, Phishing Attacks and Malware Distribution.	Medium
195	Possible BREACH (Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext) Vulnerability	The BREACH (Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext) attack is a vulnerability that exploits the way compressed data is encrypted in HTTPS responses. By injecting known plaintext into an HTTPS request and analyzing the size of the corresponding compressed response, an attacker can infer the content of sensitive data, such as session tokens or other confidential information.	Low
196	Python Code Injection	Python code injection vulnerabilities happen when a web application allows user input to be included in a code snippet that is run by the server. If this input is not properly checked or sanitized, attackers can manipulate it to run their own harmful code on the server. This can lead to the server executing unwanted commands or giving unauthorized access to sensitive data.	High
197	Client-side desync (CSD) attack	Client-side desync (CSD) is a type of attack that causes a victimâ€™s web browser to desynchronize its connection with a vulnerable website. This is different from traditional request smuggling attacks, which typically involve desynchronization between a front-end and back-end server. In a CSD attack, the attacker manipulates the browser to send a request that the server misinterprets, leading to potential security issues like cross-site scripting (XSS) or other malicious actions	High

198	Serialized Object in HTTP Message	A security vulnerability that arises when an application transmits data using serialized objects, which can be manipulated by injecting malicious payloads.	Medium
199	LDAP Injection	Injects LDAP payloads into request parameters and analyzes responses for vulnerabilities. Uses response comparison and error detection to confirm LDAP Injection.	Medium