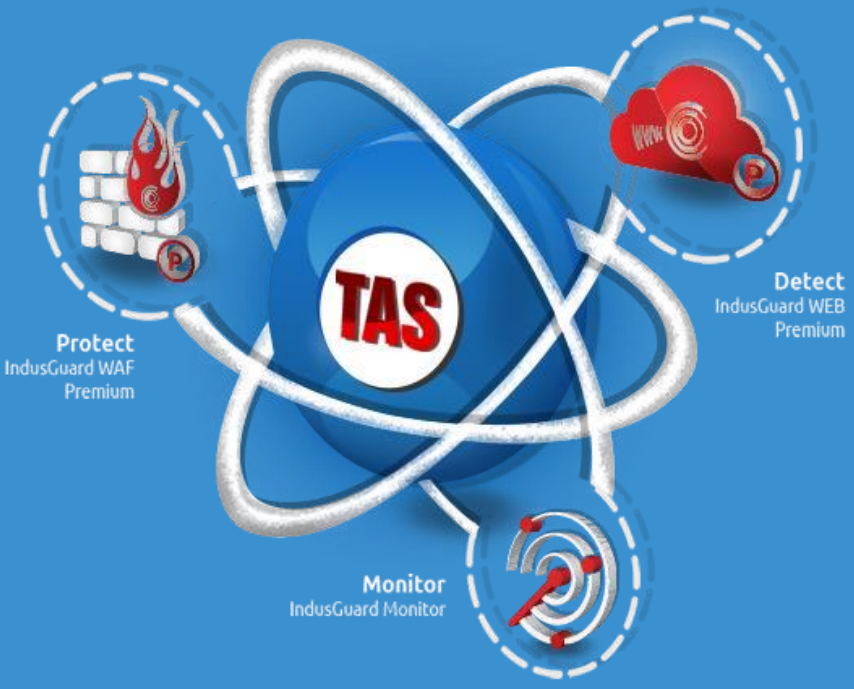
INDUSFACE™



Indusface Scan

Vulnerabilities that Indusface WAS Scans

DOCUMENT LAST REVISED ON **15, May 2020**

Confidentiality

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	Vulnerability Name	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.	Medium
2.	HTTP Response Splitting	HTTP response splitting is a form of web application attack where unsafe characters are inserted into user controllable field which are later inserted into HTTP header being used for 302 redirect. 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 Cross site scripting, web-cache poisoning, and cross-user-defacement attacks.	High
3.	Microsoft IIS Internal IP Address Disclosure (CVE-2002-0422)	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.	Possible Blind SQL Injection	Web applications usually store information in a SQL server in order to, for example, show them to other users. When the application developer uses unvalidated user controlled variables as part of a SQL query; a SQL injection or Blind SQL injection vulnerability is being introduced into the application. When an attacker executes SQL Injection attacks, sometimes the server responds with error messages from the database server complaining that the SQL Query's syntax is incorrect. Blind SQL injection is identical to normal SQL Injections except that when an attacker attempts to exploit an application, rather than getting a potentially useful error message, they get a generic page specified by the developer instead. This makes exploiting a potential Blind SQL Injection attack more difficult but not impossible. An attacker can still retrieve valuable information and potentially execute operating system commands by asking a series of True and False questions through SQL statements.	High

6.	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. 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: 1. Access other sites inside another client's private intranet. 2. Steal another client's cookie(s) 3. Modify another client's cookie(s) 4. Steal another client's submitted form data 5. Modify another client's submitted form data before it reaches the server 6. 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: 1. Having a user click a URL link sent in an e-mail 2. 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.	High
7.	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), orby 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
8.	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
9.	TLS/SSL Server Certificate Expired	The server's HTTPS X.509 certificate is expired.	Critical
10.	WebDAV Extensions Are Enabled	WebDAV is a set of extensions to the HTTP protocol that allows users to collaboratively edit and manage files on remote web servers. Many web servers enable WebDAV extensions by default, even when they are not needed. Because of its added complexity, it is considered good practice to disable WebDAV if it is not currently in use.	Info

11.	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
12.	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
13.	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
14.	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
15.	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's Existing files may be overwritten, resulting in a loss of integrity.	High
16.	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
17.	Missing Secure Flag from SSL Cookie	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
18.	Sensitive HTML Form Fields With auto-complete Enabled	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 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.	Low
19.	HTTP Basic Authentication 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 username and password are passed over the network as cleartext.	Medium

20.	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, processed or invoked by the Web server. This is one of the flaws under the category of Injection, in the OWASP Top Ten.	Critical
21.	ASP.NET Unencrypted "__VIEWSTATE" Parameter	The application uses the ASP.NET 2.0 view state (__VIEWSTATE) feature without encryption to maintain application state. The view state can be protected from tampering by using either encryption or signing. If only signing is used (without encryption), then the internal of the view state parameter can be exposed simply by Base64-decoding it. In a well-designed application, the view state should never contain any sensitive information. However, application designers have been known to put passwords and other sensitive data inside the view state. Therefore, it is a good idea to always use view state encryption in ASP.NET applications.	Medium
22.	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 back-end, 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
23.	Missing HttpOnly Flag from Cookie	HttpOnly is an additional flag included in a Set-Cookie HTTP response header. If supported by the browser, using the HttpOnly flag when generating a cookie helps mitigate the risk of client side script accessing the protected cookie. If a browser that supports HttpOnly detects a cookie containing the HttpOnly 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.	Low
24.	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
25.	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

26.	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
27.	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
28.	Email Address Disclosure	There are number of crawlers running across the Internet to search email addresses from all the publicly available websites. 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 lists, your inbox will receive dozens of spams daily. This may lead to missing out an important email.	Info
29.	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
30.	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_new like SQL Injection, the information can be very helpful to the hacker to gain access to your database.	Medium
31.	Internal IP Address Disclosure	Subtle data may be used by an attacker to exploit the target hosting network, web application, or its users.	Low
32.	Possible Backup File(s) Detected	A possible backup file has been found on your webserver. These files are usually created by developers to back up their work or by administrators when making backups of the web server.	Medium
33.	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
34.	Local File Inclusion (LFI)	This script is possibly vulnerable to file inclusion attacks. xIt 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

35.	Permissive Cross-domain 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 href="http://www.adobe.com/devnet/flashplayer/articles/cross_domain_policy.html Cross-domain policy file usage recommendations for Flash Player.	Low
36.	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
37.	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
38.	Web Server Version 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
39.	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
40.	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
41.	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

42.	HTML Injection	HTML injection attack is like Cross-site Scripting (XSS). In XSS vulnerability attacker can execute the injected 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 into the application which results in this vulnerability. Once this is exploited it can further be used by attacker to perform other attacks.	High
43.	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
44.	HTTPS And Mixed Content Vulnerability	HTTPS is used to make communication between the server and the browser secure. However, a problem occurs when an HTTPS page loads HTTP content: this is called mixed content vulnerability.	Medium
45.	Insecure Content Security Policy (CSP)/X-Frame-Options	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.	Low
46.	Session ID In URL	Storing the HTTP session information only URL is a highly insecure practice and leaves the HTTP session information open to theft through packet sniffing or observation of proxy logs.	Medium
47.	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 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 etc.	Medium
48.	Unencoded 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

49.	Cross-Origin Resource Sharing (CORS)	The HTML5 cross-origin resource sharing policy controls whether and how content running on other domains can perform two-way 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, security vulnerabilities within that domain could potentially be leveraged by a third-party attacker to exploit the trust relationship and attack the application which allows access.	Low
50.	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 intended 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, via a self-service unlock mechanism, or intervention by an administrator.	High
51.	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
52.	CVS Web Repository Disclosure	CVS Web Repository was found on this webpage. The CVS directory is a special directory. CVS/Entries list 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
53.	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
54.	Configuration File Disclosure	Configuration file or backup of configuration file has been found. This file may contain sensitive data which should not be available publicly.	High
55.	X-XSS-Protection 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

56.	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
57.	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
58.	Insecure Flash Parameter "AllowScriptAccess" 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 cross-domain 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 a 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's in transit between client and server.	High
64.	Apache Struts2 Development 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 server-status 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. It's 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 "price" has been detected within the html source code of an application and altering the value of such input filed will cause serious damage.	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.	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.	Medium
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.	Application Development Configuration File Detected	Development configuration files are used to configure the parameters and initial settings of development framework of user applications. Successful disclosure of development configuration files allows attacks to gain sensitive information about the applications and assist in conducting further attacks.	Low
76.	Missing 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 attacks 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 an 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 also refers generically to the use of XML for a remote procedure call, independently of the specific protocol. A public facing WordPress XML-RPC interface has been detected. An attacker may exploit this issue to execute arbitrary commands or code in the context of the web server. This may facilitate various attacks, including unauthorized remote access.	Medium
79.	Apache Tomcat Remote Code Execution Vulnerability (CVE-2019-0232)	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
80.	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
81.	Oracle WebLogic Server Deserialization Remote Command Execution Vulnerability (CVE-2019-2725)	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
82.	Dot Net Insecure Deserialization Remote Command Execution Vulnerability	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
83.	Perl Deserialization Remote Command Execution Vulnerability	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
84.	Passive Mixed Content Vulnerability	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

85.	Active Mixed Content Vulnerability	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 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.	Medium
86.	PHP Deserialization Remote Command Execution Vulnerability (CVE-2017-17672)	"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
87.	Ruby on Rails XML/JSON Processor YAML Deserialization Code Execution Vulnerability (CVE-2013-0156)	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 Denial-Of-Service attacks. This has been tested against 3.x & 2.x versions of RoR which are vulnerable.	High
88.	Oracle WebLogic Server Deserialization Remote Command Execution Bypass Vulnerability (CVE-2019-2729)	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 XML Decoder 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
89.	XML External Entity (XXE) Injection Vulnerability	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
90.	Possible Archive File or Compression 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	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 Cross-Site 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 CSRF Protection	Cross-Site Request Forgery (CSRF/XSRF) is a vulnerability where attacker tricks victim into making a request victim did not make. So, 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.	Medium
97.	Web Administration Login Page Detected	An application can be configured & controlled by an administrator 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 brute-force attacks, SQL injection, etc.	Low
98.	Web Server Content Sniffing Enabled	Content Sniffing is a technique used by the browsers to determine an asset's correct file format. Web Servers examine the content of HTTP responses in addition to the Content-Type header field in order to determine the effective MIME (Multipurpose Internet Mail Extensions) type of the response. If any response does not contain the content type, then browser will try to sniff the MIME means try to determine type of content based on looking at the response body. Attackers can use this to perform XSS attacks by manipulating content sent to server or injecting code in the file.	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 Vulnerability	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 JavaScript 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 application 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.	Common Credentials Found	Common credentials found vulnerability means that username and/or password provided/found for the account are commonly used credentials.	Info

107.	SQL Injection Authentication 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
108.	Login Username Enumeration	Web applications which fail to respond with consistent error messages when a user attempts to login with existing and non-existing 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
109.	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
110.	JSF Client-Side ViewState Detected	Java Server Faces (JSF) is a Java-based Web application framework that implements the Model-View-Controller 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.	Medium
111.	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
112.	Insecure Cache-Control 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 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.	Low
113.	Old SSL/TLS Version Detected	If the connection to site is made using old ssl/tls versions like SSLv3, TLSv1 & TLSv1.1 which are SSLv3 is 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
114.	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
115.	Apache Axis2 LFI	Local File Inclusion (LFI) vulnerability in the 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

116.	Insecure/Deprecated Cryptography	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
117.	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 quering in the url. Once exploited attacker would have access sensitive information like secret key, tokens etc.	High
118.	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
119.	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
120.	HTTP Verb Tampering	HTTP Verb Tampering vulnerability allows an attacker to bypass authorization by manipulating or using unsupported, specially crafted HTTP methods. Successful exploitation leads to obtain sensitive information and perform unauthorized actions; this may aid in launching further attacks.	Medium