Vulnerability Assessment Report: Reliance Power

Prepared by:

Kushal M V (kushal.kmv@gmail.com)

Contents

Executive Summary:

As part of the state-level competition - March for Secure Code, a few outstanding participants were considered eligible for the SkillsDA Internship. March for Secure Code was a month-long programme followed by a state-level competition on secure application coding and cybersecurity by SkillsDA, in partnership with NASSCOM.

The primary goal of this internship is to make the websites of the most valuable organizations more secure for both the clients visiting the sites and the organizations hosting the sites. The interns perform Vulnerability Assessments on websites from Banking, Aviation and Power Sector of the participants' choice. Interns can choose websites in groups of two and conduct Vulnerability Assessments on them. Report of which has to be submitted to the internship guides at SkillsDA.

Assessment Methodology:

The above report is prepared by performing black box hacking which is a type of ethical hacking performed when we have no information at all about the network, system or application we are going to hack or find vulnerabilities of.

We have performed external penetration testing which is a testing done typically from outside of a network and tests the ability for hackers to break in from outside. We have performed all these tests with the public IP addresses available for the network.

We have performed a website penetration testing it is important to perform this because if the company is offering an application online that collects private information of people or contains very important private data, then they need to know if their application is safe.

We have used automation testing strategies like-

- Defining goals
- Planning the test approach
- Selecting automation framework
- Selecting testing tools
- Test case design and execution

About the Website:

Website Title: Reliance Power

IP Address: 220.226.182.125

Registrant Org: Anil Dhirubhai Ambani Ventures Private Limited

Registrant Country: India

Server-Side Technology:Java Servlet

Client-Side Technology: JavaScript

Vulnerability Analysis:

Summary of the Assessment:

Vulnerability	Status	Risk
Vulnerable libraries	Found	High
Directory Traversal	Found	High
Missing Content-Type Header Vulnerabilities	Found	Medium
Open Ports	Found	Medium
SQLi	Not Found	-
Source code disclosure vulnerability	Not Found	-
Cross-Site Scripting	Not Found	-
Host header injection	Not Found	-
Directory Listing Vulnerability	Not Found	-
XML External Entities Injections	Not Found	-

Risk Level Description:

The bellow vulnerability ranging risk pattern indicates the ratings of the vulnerability according to their respective CVSS3.1 Score.

Risk Level	Risk Description and Necessary Action
High	**The high risk level** indicates maximum risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to successfully exploit the underlying application and its data and partially or completely to compromise the application and its data to modify application behaviour to become other than its original intended purpose. The vulnerability marked as "High Risk" is recommended tobe handled with utmost priority.
Medium	**The medium risk level** indicates considerable risk associated with a specific vulnerability instance. Such vulnerability may enable an attacker to exploit the underlying application and its data to a particular level so that the attacker can gain low level information about the application. Such information can be used by an attacker to craft more specific attacks based on the information collected.
Low	**The low risk level** indicates lowest risk associated with a specific vulnerability instance. Such vulnerability may allow an attacker to gain some information about the application which was not intended to be known otherwise. The attacker may not have exploiting techniques available at that instance based on the information revealed by the system.

High Risk Findings:

1. Vulnerable libraries:

Common security vulnerabilities within JavaScript include cross- site scripting, cross-site request forgery, and buffer overflows. Cross-site scripting allows attackers to inject malicious code into trusted web pages, which then serve that malicious code to users that view the site.

a. Home page: https://www.reliancepower.co.in/web/reliance-Power/index

jquery 2.1.4	Found in https://www.reliancepower.co.in/o/frontend-js-web/everything.jsp?browserId=firefox&themeId=ReliancePowerThemeLive_WAR_ReliancePowerThemeLive&colorSchemeId=01&minifierType=js&minifierBundleId=javascript.everything.files&languageId=en_US&b=7010&t=1623660042253 - Vulnerability info: medium 2432 3rd party CORS request may execute CVE-2015-9251 medium CVE-2015-9251 11974 parseHTML() executes scripts in event handlers medium CVE-2019-11358 jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution medium CVE-2020-11022 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS medium CVE-2020-11023 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS

b. Tenders page: https://www.reliancepower.co.in/web/reliance-Power/

jquery 2.1.4	Found in https://www.reliancepower.co.in/o/frontend-js-web/everything.jsp?browserId=firefox&themeId=ReliancePowerThemeLive_WAR_ReliancePowerThemeLive&colorSchemeId=01&minifierType=js&minifierBundleId=javascript.everything.files&languageId=en_US&b=7010&t=1623660042253 - Vulnerability info: medium 2432 3rd party CORS request may execute CVE-2015-9251 medium CVE-2015-9251 11974 parseHTML() executes scripts in event handlers medium CVE-2019-11358 jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution medium CVE-2020-11022 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS medium CVE-2020-11023 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS

c. Shareholder registration page: https://www.reliancepower.co.in/web/reliance-Power/ shareholder-registration

jquery-mobile	1.4.5	Found in https://tatapower-ddl.com/Script/jquery.mobile.custom.min.js - Vulnerability info: medium open redirect leads to cross site scripting 1
jquery	3.3.1	Found in https://tatapower-ddl.com/Script/jquery.min.js - Vulnerability info: medium CVE-2019-11358 jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution medium CVE-2020-11022 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS medium CVE-2020-11023 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS
jquery	3.4.1	Found in https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js - Vulnerability info: medium CVE-2020-11022 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS 1 medium CVE-2020-11023 Regex in its jQuery.htmlPrefilter sometimes may introduce XSS 1
jquery	3.6.0	Found in https://cdn.yellowmessenger.com/plugin/latest/dist/main.min.js
jquery	3.6.0	Found in https://cdn.yellowmessenger.com/plugin/latest/dist/widget.min.js

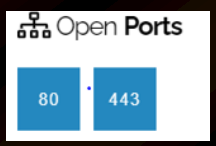
2. Directory Traversal:

Directory traversal (also known as file path traversal) is a web security vulnerability that allows an attacker to read arbitrary files on the server that is running an application. This might include application code and data, credentials for back-end systems, and sensitive operating system files.

```
http://220.226.182.128:80/../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../etc/issue <- VULNERABLE!
http://220.226.182.128:80/../../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../../etc/issue <- VULNERABLE!
http://220.226.182.128:80/../../../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../../../etc/issue ← VULNERABLE!
http://220.226.182.128:80/../../../../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../../../../etc/issue ← VULNERABLE!
http://220.226.182.128:80/../../../../../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../../../../../etc/issue ← VULNERABLE!
http://220.226.182.128:80/../../../../../../etc/passwd ← VULNERABLE!
http://220.226.182.128:80/../../../../../../etc/issue ← VULNERABLE!
http://220.226.182.128:80/..%5Cetc%5Cpasswd ← VULNERABLE!
http://220.226.182.128:80/..%5Cetc%5Cissue ← VULNERABLE!
```

Medium Risk Findings

3. Open Ports:




```
┌──(root💀kali)-[/home/dart]
└─# nmap -p 443 220.226.182.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-18 08:23 EDT
Nmap scan report for www.rcem.co.in (220.226.182.128)
Host is up (0.0011s latency).

PORT    STATE    SERVICE
443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 0.65 seconds

┌──(root💀kali)-[/home/dart]
└─# nmap 220.226.182.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-18 08:23 EDT
Nmap scan report for www.rcem.co.in (220.226.182.128)
Host is up (0.039s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 49.84 seconds
```

Open ports can be dangerous when the service listening on the port is misconfigured, unpatched, vulnerable to exploits, or has poor network security rules.

Other Vulnerabilities Scanned for:

4. SQLi:

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve.

Using uniscan we found that website is successfully able to escape the scan of its important data because it does not return 404, also I was unsuccessful in injecting the script into the website, hence sqli vulnerability did not exist.

5. Missing Content-Type Header Vulnerabilities

MIME type sniffing is a standard functionality in browsers to find an appropriate way to render data where the HTTP headers sent by the server are either inconclusive or missing. This vulnerability allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the intended content type.

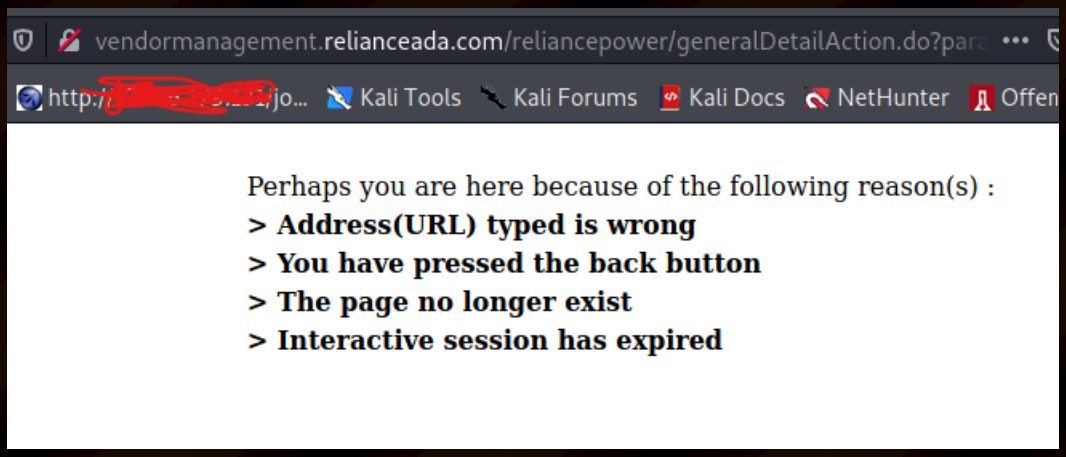
```
└─# nikto -h 220.226.182.128 -p 80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          220.226.182.128
+ Target Hostname:    220.226.182.128
+ Target Port:        80
+ Start Time:         2021-08-18 08:20:02 (GMT-4)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some f
orms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the s
ite in a different fashion to the MIME type
```

X-Frame-Options header was not found, which means that this website cannot be clickjacked.

6. Cross-Site Scripting:

Cross-site scripting attacks, also called XSS attacks, are a type of injection attack that injects malicious code into otherwise safe websites. An attacker will use a flaw in a target web application to send some kind of malicious code, most commonly client-side JavaScript, to an end user.

Xspear script was used to test for XSS, no vulnerability was found

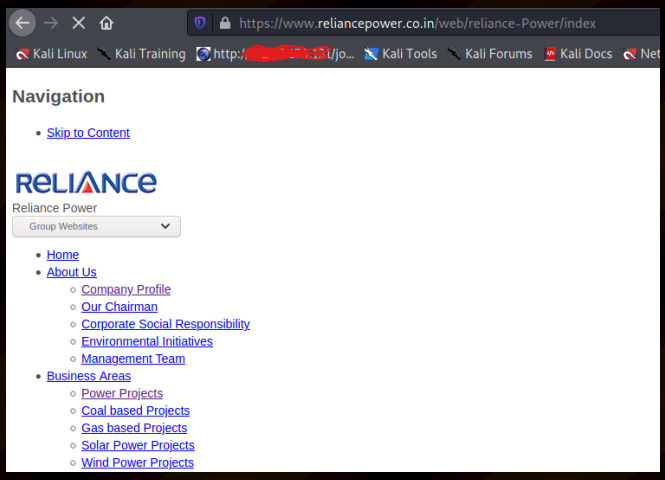


7. Host header injection:

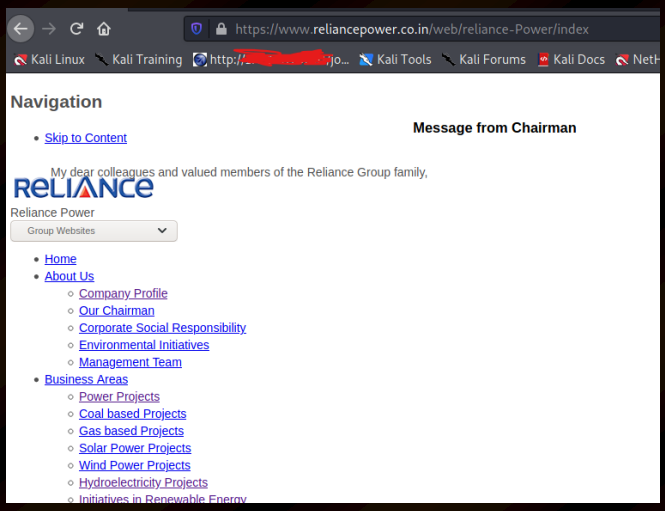
A Host header attack, also known as Host header injection, is a web attack where the attacker provides a false Host header to the web application. Find more information about other types of injection attacks.

a. Changing host:

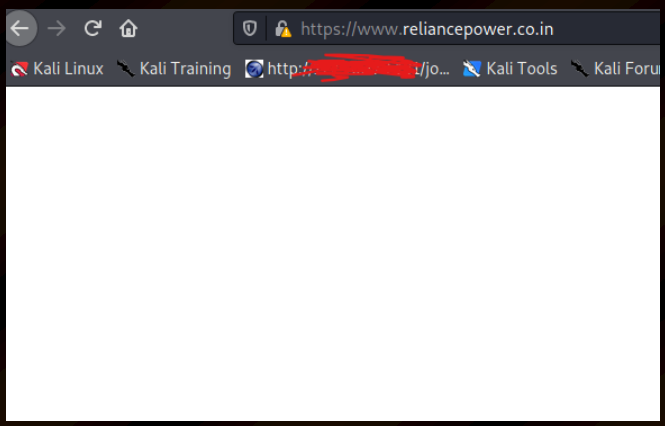
Tried injecting the header with www.bing.com, but the website resisted it and gave the below output.



b. X-Forwarded-Host:



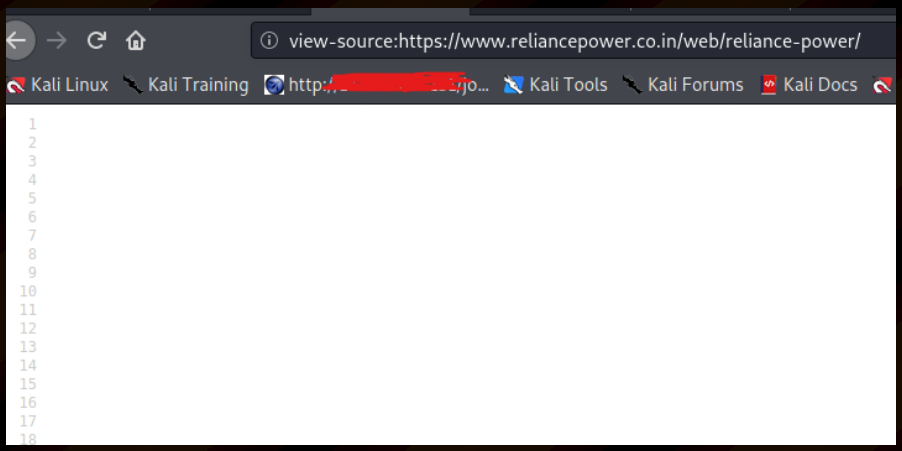
c. Swapping X-Forwarded-Host:



8. Directory Listing Vulnerability:

A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers. It provides an attacker with the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible.

It is a web server function. The web server processes this request and searches the document root directory for the default file name and sends this page to the client. If this page is not present, the web server will issue a directory listing and send the output to the client. It is dangerous to leave this function turned on for the web server because it leads to information disclosure.



9. XML External Entities Injections

An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the 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.

Since no comment or other similar text input labels were found, it makes the website resistant to XXE.

10. Source code Disclosure Vulnerability

Source code disclosure attacks allow a malicious user to obtain the source code of a server-side application. This vulnerability grants the attacker deeper knowledge of the Web application logic. When the browser requests a dynamic file, the Web server first executes the file and then returns the result to the browser. Source code intended to be kept server-side can sometimes end up being disclosed to users. Such code may contain sensitive information such as database passwords and secret keys, which may help malicious users formulate attacks against the application.

Reliance Power website did not disclose any noticeable source code; hence it is safe from this perspective.

Conclusion:

After analyzing the results, we can say that www.reliancepower.co.in is a well secured website which abides by the optimal security measures. Although it is fairly secure website, it is still having some vulnerable libraries and it still has directory traversal vulnerability.

Thank You

Submitted by:

Kushal M V (kushal.kmv@gmail.com)