200.200.200.137/8005

Annexure-II

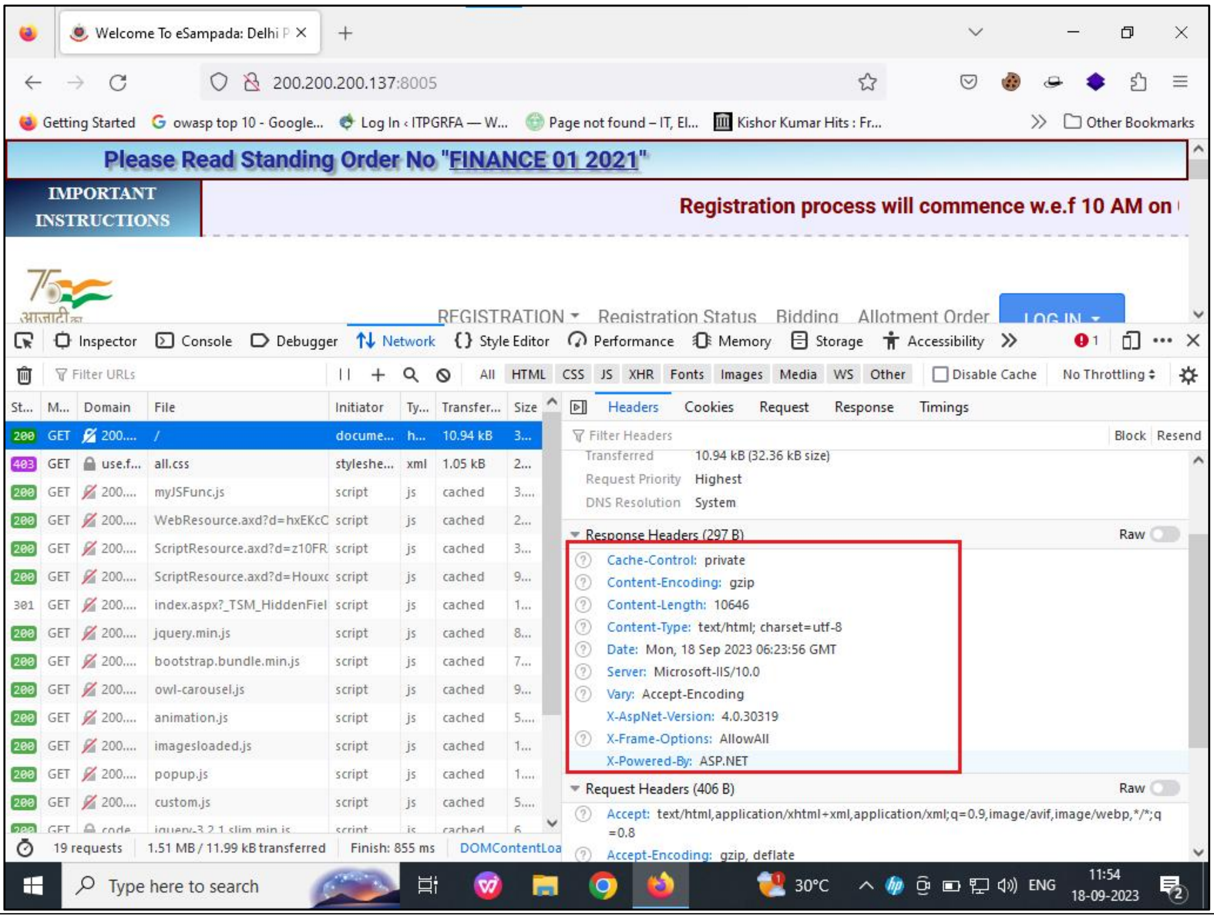
1. Name of Vulnerability/Error:	1. Content Security Policy Header Missing (Medium)
2. Location of Vulnerability/Error:	1.1 200.200.200.137/8005

3. Vulnerability Description:

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

4. Proof of concept and steps of verification of vulnerability with screenshots:

POC:



St...	M...	Domain	File	Initiator	Ty...	Transfer...	Size
200	GET	200....	/	docume...	h...	10.94 kB	3...
403	GET	use.f...	all.css	styleshe...	xml	1.05 kB	2...
200	GET	200....	myJSFunc.js	script	js	cached	3....
200	GET	200....	WebResource.axd?d=hxEKcC	script	js	cached	2...
200	GET	200....	ScriptResource.axd?d=z10FR	script	js	cached	3...
200	GET	200....	ScriptResource.axd?d=Houxc	script	js	cached	9...
301	GET	200....	index.aspx?_TSM_HiddenFiel	script	js	cached	1...
200	GET	200....	jquery.min.js	script	js	cached	8...
200	GET	200....	bootstrap.bundle.min.js	script	js	cached	7...
200	GET	200....	owl-carousel.js	script	js	cached	9...
200	GET	200....	animation.js	script	js	cached	5....
200	GET	200....	imagesloaded.js	script	js	cached	1...
200	GET	200....	popup.js	script	js	cached	1....
200	GET	200....	custom.js	script	js	cached	5....
200	GET	code...	jquery-3.2.1.slim.min.js	script	js	cached	6...

```
19 requests    1.51 MB / 11.99 kB transferred    Finish: 855 ms    DOMContentLoa
```

```
Transferred        10.94 kB (32.36 kB size)
Request Priority   Highest
DNS Resolution     System

Response Headers (297 B)
Cache-Control: private
Content-Encoding: gzip
Content-Length: 10646
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Sep 2023 06:23:56 GMT
Server: Microsoft-IIS/10.0
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-Frame-Options: AllowAll
X-Powered-By: ASP.NET

Request Headers (406 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

6. References:

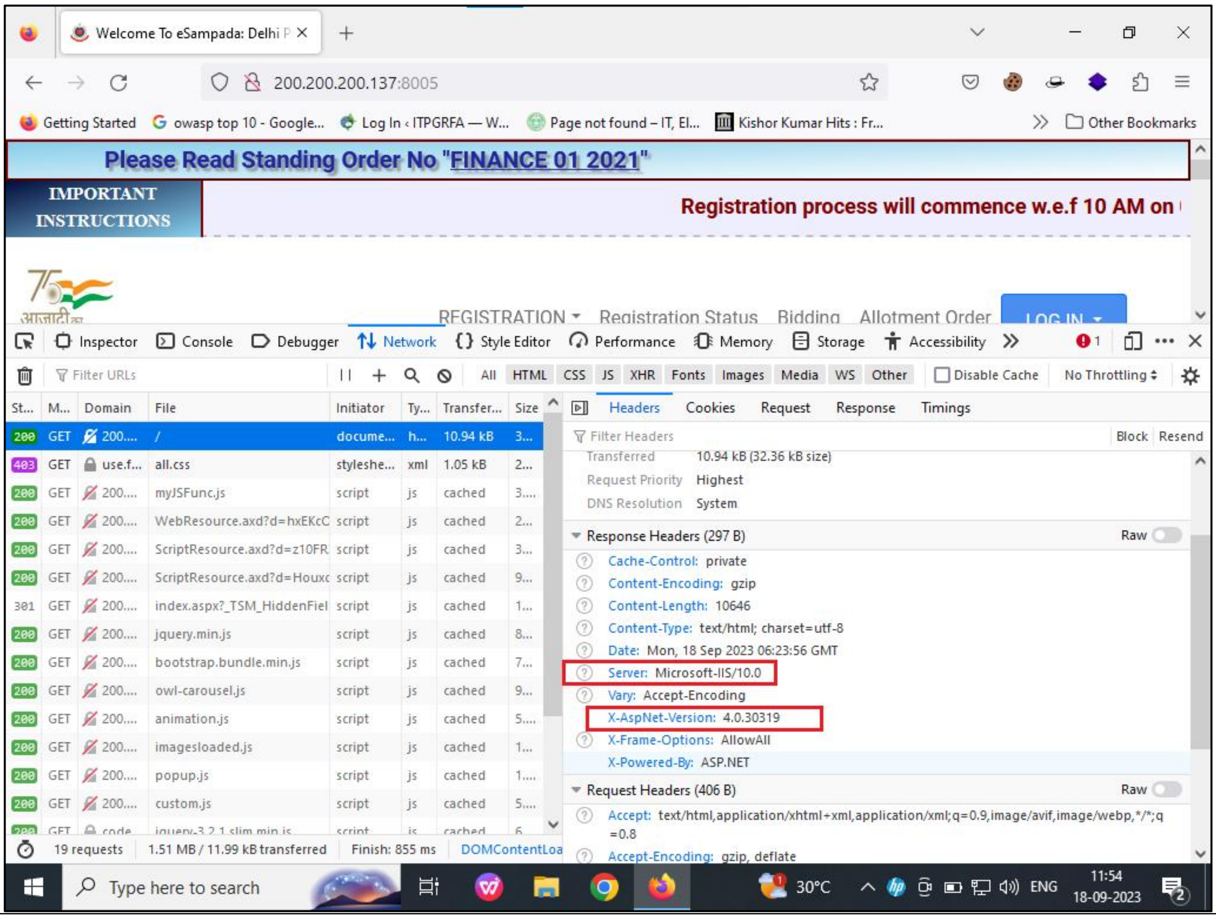
- **CWE-693: Protection Mechanism Failure**
  https://cwe.mitre.org/data/definitions/693.html

1. Name of Vulnerability/Error:	**2. Version Disclosure (Medium)**
2. Location of Vulnerability/Error:	2.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:
The product exposes sensitive information to an actor that is not explicitly authorized to have access to that information.

4. Proof of concept and steps of verification of vulnerability with screenshots

POC:



Welcome To eSampada: Delhi P

200.200.200.137:8005

Please Read Standing Order No "FINANCE 01 2021"

IMPORTANT INSTRUCTIONS

Registration process will commence w.e.f 10 AM on

```
St...  M...  Domain   File                         Initiator    Ty...  Transfer...  Size
200    GET   200....  /                            docume...    h...   10.94 kB     3...
403    GET   use.f... all.css                      styleshe...  xml    1.05 kB      2...
200    GET   200....  myJSFunc.js                  script       js     cached       3....
200    GET   200....  WebResource.axd?d=hxEKcC     script       js     cached       2...
200    GET   200....  ScriptResource.axd?d=z10FR   script       js     cached       3...
200    GET   200....  ScriptResource.axd?d=Houxc   script       js     cached       9...
301    GET   200....  index.aspx?_TSM_HiddenFiel   script       js     cached       1...
200    GET   200....  jquery.min.js                script       js     cached       8...
200    GET   200....  bootstrap.bundle.min.js      script       js     cached       7...
200    GET   200....  owl-carousel.js              script       js     cached       9...
200    GET   200....  animation.js                 script       js     cached       5....
200    GET   200....  imagesloaded.js              script       js     cached       1...
200    GET   200....  popup.js                     script       js     cached       1....
200    GET   200....  custom.js                    script       js     cached       5....
19 requests   1.51 MB / 11.99 kB transferred   Finish: 855 ms   DOMContentLoa
```

```
Transferred         10.94 kB (32.36 kB size)
Request Priority    Highest
DNS Resolution      System

Response Headers (297 B)
Cache-Control: private
Content-Encoding: gzip
Content-Length: 10646
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Sep 2023 06:23:56 GMT
Server: Microsoft-IIS/10.0
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-Frame-Options: AllowAll
X-Powered-By: ASP.NET

Request Headers (406 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- Compartmentalize the system to have "safe" areas where trust boundaries can be unambiguously drawn. Do not allow sensitive data to go outside of the trust boundary and always be careful when interfacing with a compartment outside of the safe area.

6. References:

- **CWE-200: Exposure of Sensitive Information to an Unauthorized Actor,** https://cwe.mitre.org/data/definitions/200.html

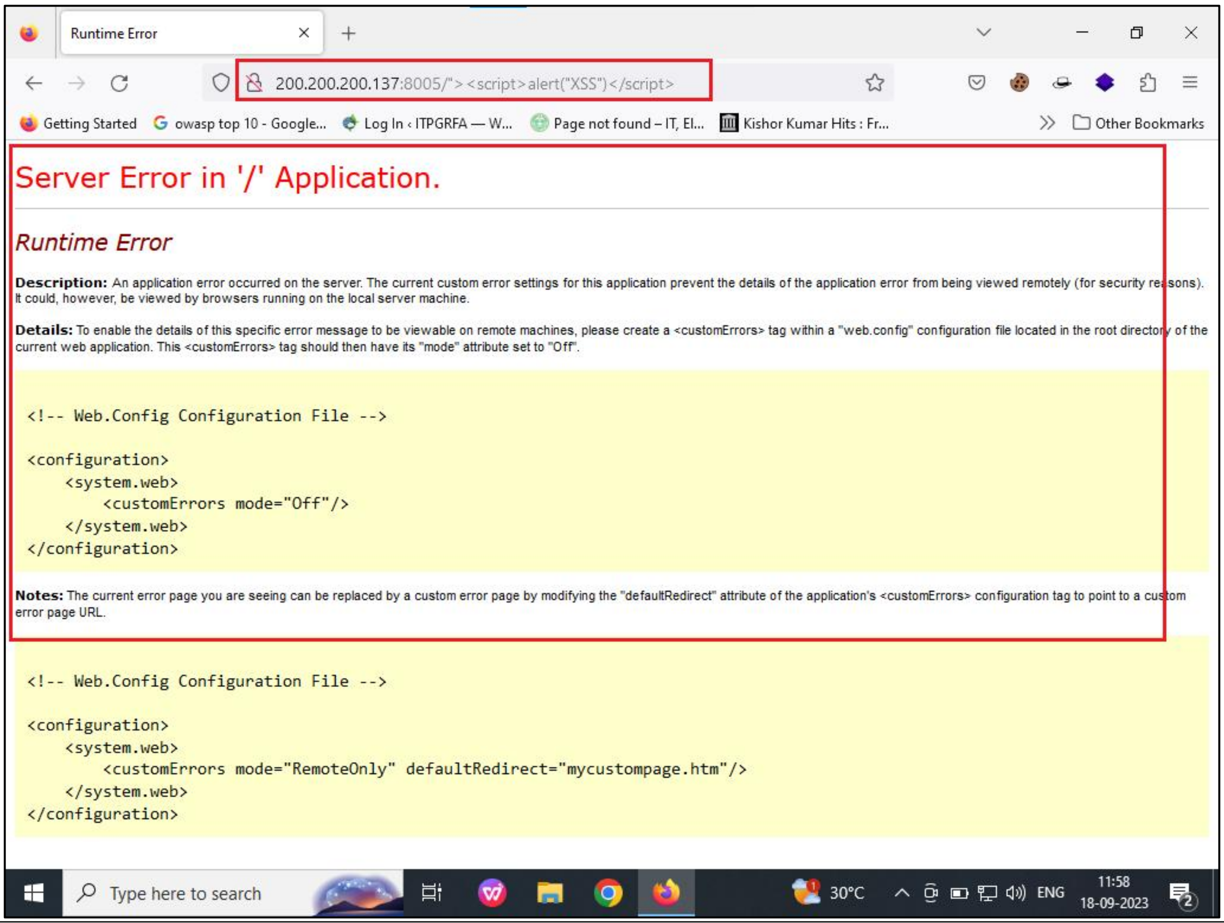
1. Name of Vulnerability/Error:	**3. Application Error Disclosure (Medium)**
2. Location of Vulnerability/Error:	3.1 200.200.200.137/8005/"><script>alert("XSS")</script>

3. Vulnerability Description:

This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page..

4. Proof of concept and steps of verification of vulnerability with screenshots

POC



Server Error in '/' Application.

Runtime Error

Description: An application error occurred on the server. The current custom error settings for this application prevent the details of the application error from being viewed remotely (for security reasons). It could, however, be viewed by browsers running on the local server machine.

Details: To enable the details of this specific error message to be viewable on remote machines, please create a <customErrors> tag within a "web.config" configuration file located in the root directory of the current web application. This <customErrors> tag should then have its "mode" attribute set to "Off".

```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="Off"/>
    </system.web>
</configuration>
```

Notes: The current error page you are seeing can be replaced by a custom error page by modifying the "defaultRedirect" attribute of the application's <customErrors> configuration tag to point to a custom error page URL.

```
<!-- Web.Config Configuration File -->

<configuration>
    <system.web>
        <customErrors mode="RemoteOnly" defaultRedirect="mycustompage.htm"/>
    </system.web>
</configuration>
```

5. Solution & Work around:

- Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.

6. References:

- **CWE-200: Exposure of Sensitive Information to an Unauthorized Actor.** https://cwe.mitre.org/data/definitions/200.html

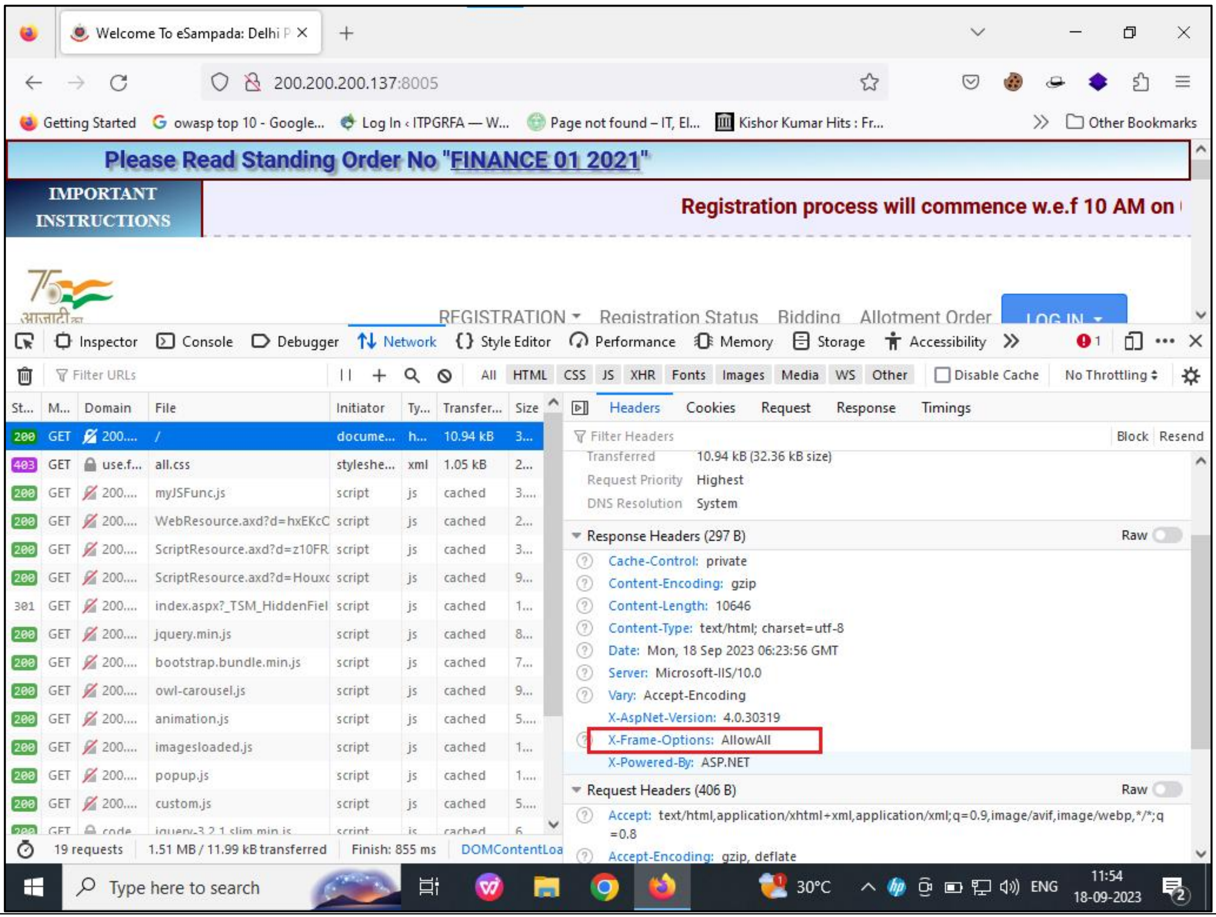
1. Name of Vulnerability/Error:	**4. X-Frame-Options Setting Malformed (Medium)**
2. Location of Vulnerability/Error	4.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

The X-Frame-Options header is a security feature that helps protect web applications against Click jacking attacks. Click jacking is a type of attack where an attacker uses a malicious website or application to trick a user into clicking on something that they did not intend to, such as a button or link.

4. Proof of concept and steps of verification of vulnerability with screenshots

POC:



St...	M...	Domain	File	Initiator	Ty...	Transfer...	Size
200	GET	200....	/	docume...	h...	10.94 kB	3...
403	GET	use.f...	all.css	styleshe...	xml	1.05 kB	2...
200	GET	200....	myJSFunc.js	script	js	cached	3....
200	GET	200....	WebResource.axd?d=hxEKcC	script	js	cached	2...
200	GET	200....	ScriptResource.axd?d=z10FR	script	js	cached	3...
200	GET	200....	ScriptResource.axd?d=Houxc	script	js	cached	9...
301	GET	200....	index.aspx?_TSM_HiddenFiel	script	js	cached	1...
200	GET	200....	jquery.min.js	script	js	cached	8...
200	GET	200....	bootstrap.bundle.min.js	script	js	cached	7...
200	GET	200....	owl-carousel.js	script	js	cached	9...
200	GET	200....	animation.js	script	js	cached	5....
200	GET	200....	imagesloaded.js	script	js	cached	1...
200	GET	200....	popup.js	script	js	cached	1....
200	GET	200....	custom.js	script	js	cached	5....
200	GET	code	jquery-3.2.1.slim.min.js	script	js	cached	6

19 requests | 1.51 MB / 11.99 kB transferred | Finish: 855 ms | DOMContentLoa

```
Transferred         10.94 kB (32.36 kB size)
Request Priority    Highest
DNS Resolution      System

Response Headers (297 B)
Cache-Control: private
Content-Encoding: gzip
Content-Length: 10646
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Sep 2023 06:23:56 GMT
Server: Microsoft-IIS/10.0
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-Frame-Options: AllowAll
X-Powered-By: ASP.NET

Request Headers (406 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- Ensure a valid setting is used on all web pages returned by your site (if you expect the page to be framed only by pages on your server (e.g. it’s part of a FRAMESET) then you’ll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider

6. References:

- **CWE-1021: Improper Restriction of Rendered UI Layers or Frames,** https://cwe.mitre.org/data/definitions/1021.html

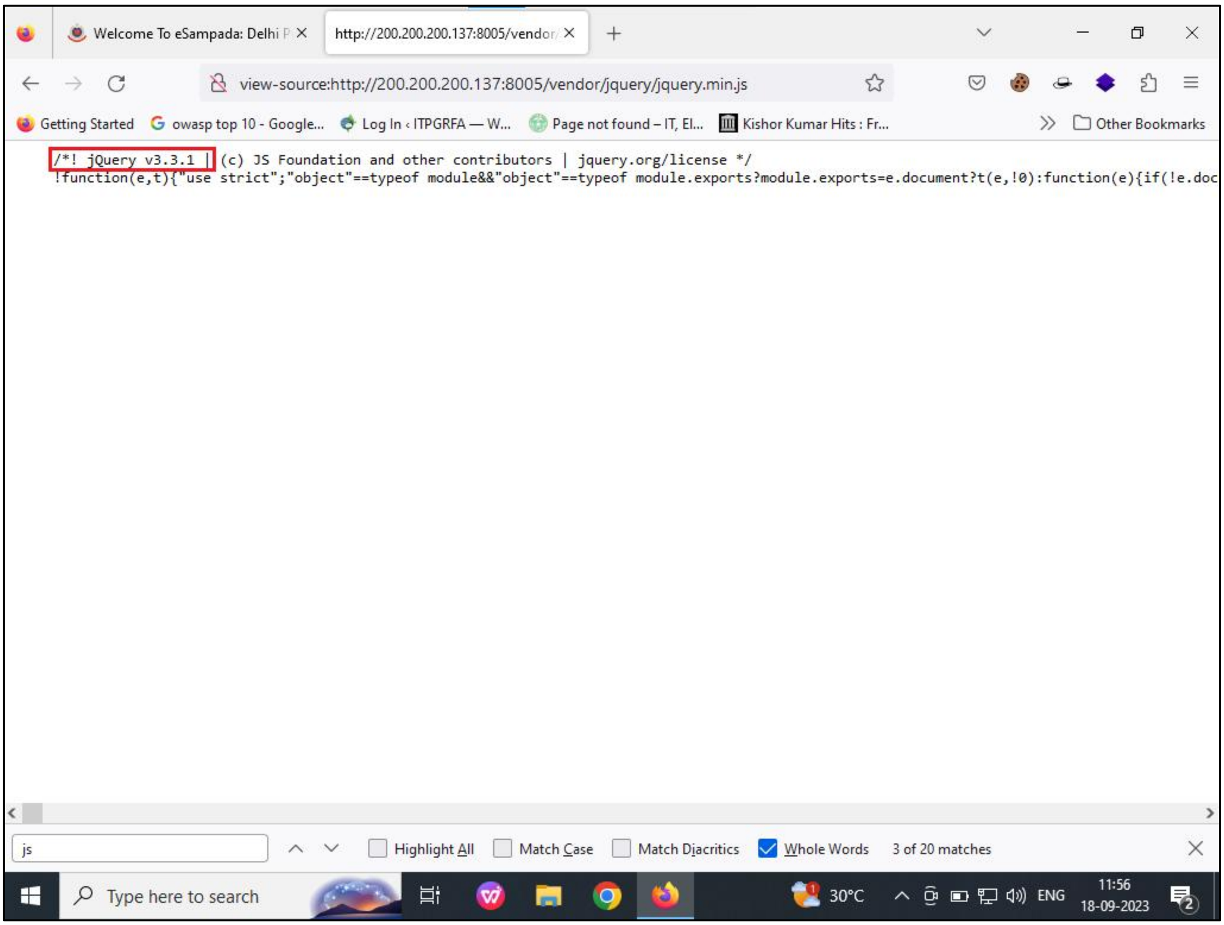
1. Name of Vulnerability/Error:	**5. Vulnerable JS Library (Medium)**
2. Location of Vulnerability/Error	5.1 view-source:http://200.200.200.137/8005/vendor/jquery/jquery.min.js

3. Vulnerability Description:

During audit we found Jquery version 3.3.1 is vulnerable. The use of third-party outdated components can introduce a range of DOM-based vulnerabilities, including some that can be used to hijack user accounts like DOM-XSS.

4. Proof of concept and steps of verification of vulnerability with screenshots:

POC:



view-source:http://200.200.200.137:8005/vendor/jquery/jquery.min.js

```
/*! jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.org/license */
!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports?module.exports=e.document?t(e,!0):function(e){if(!e.doc
```

5. Solution & Work around:

- Please upgrade to the latest version of Jquery.

6. References:

- **CWE-829: Inclusion of Functionality from Untrusted Control Sphere**, https://cwe.mitre.org/data/definitions/829.html

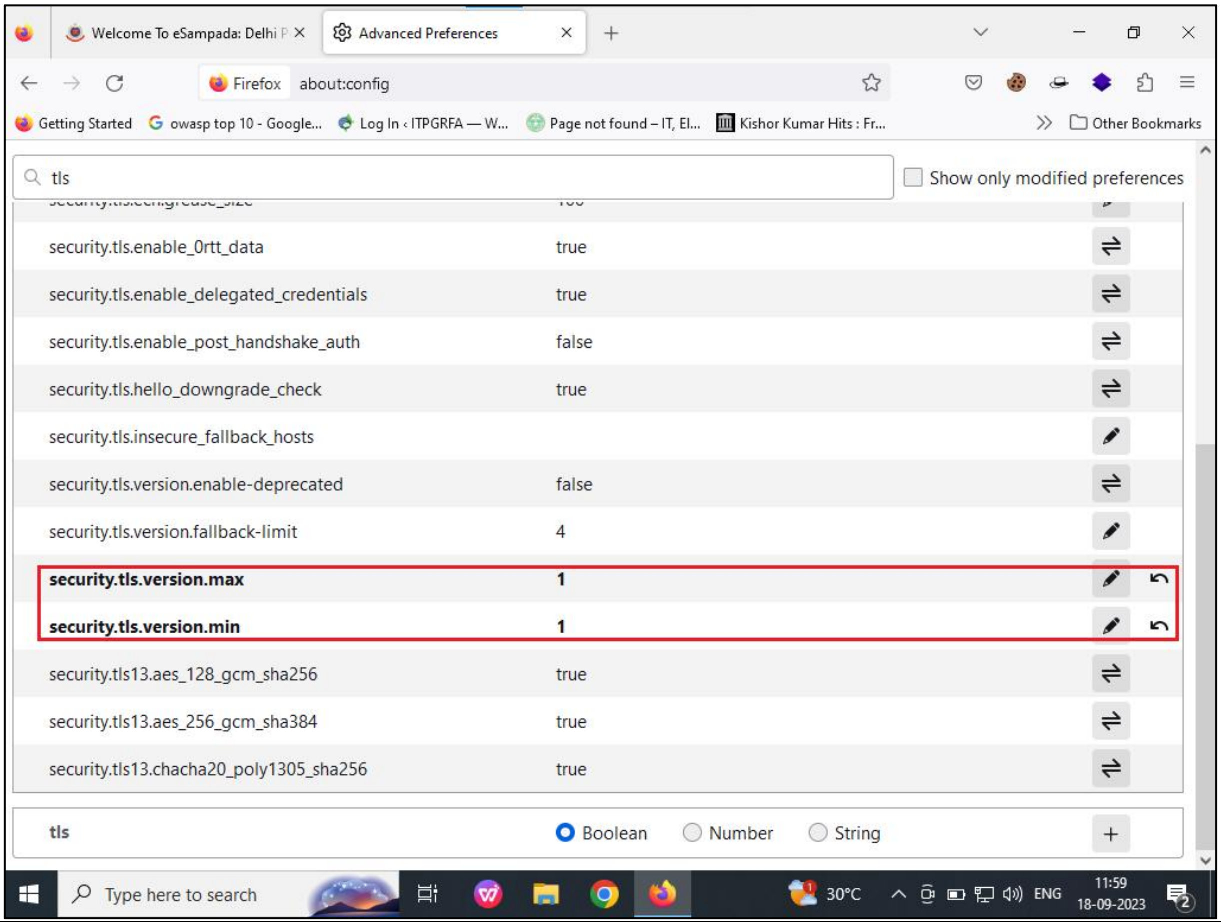
1. Name of Vulnerability/Error:	**6. Insecure Transportation Security Protocol Supported (TLS1.0) (Medium)**
2. Location of Vulnerability/Error:	6.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

The web server supports encryption through TLS 1.0. TLS 1.0 is not considered to be "strong cryptography" as defined and required by the PCI Data Security Standard 3.2. When used to protect sensitive information transferred to or from web sites. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

4. Proof of concept and steps of verification of vulnerability with screenshots:

POC:



tls

security.tls.enable_0rtt_data	true
security.tls.enable_delegated_credentials	true
security.tls.enable_post_handshake_auth	false
security.tls.hello_downgrade_check	true
security.tls.insecure_fallback_hosts	
security.tls.version.enable-deprecated	false
security.tls.version.fallback-limit	4
security.tls.version.max	**1**
security.tls.version.min	**1**
security.tls13.aes_128_gcm_sha256	true
security.tls13.aes_256_gcm_sha384	true
security.tls13.chacha20_poly1305_sha256	true

tls ● Boolean ○ Number ○ String

5. Solution & Work around:

- It is recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher.

6. References:

- **CWE-326: Inadequate Encryption Strength**
  https://cwe.mitre.org/data/definitions/326.html

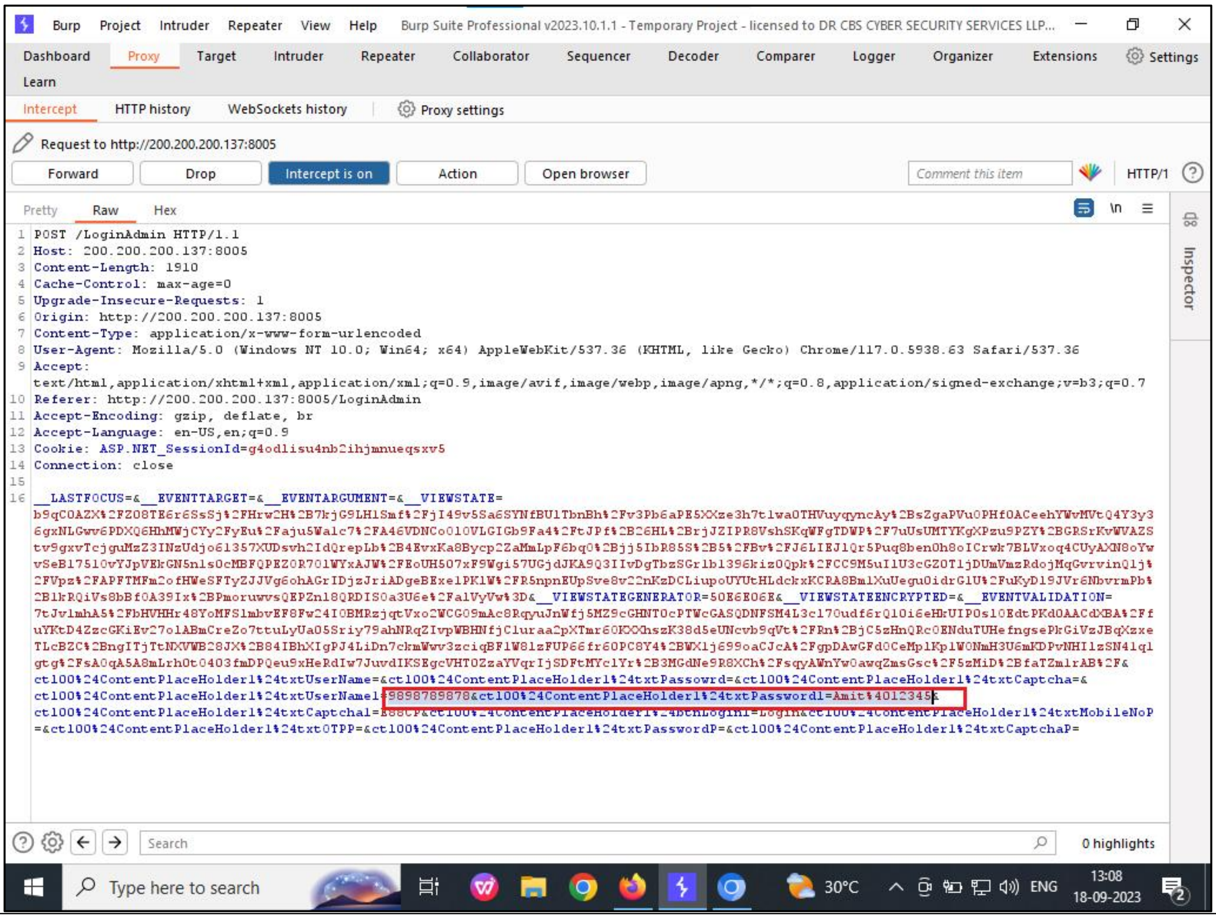
1. Name of Vulnerability/Error:	7. User credentials are sent in clear text (Medium)
2. Location of Vulnerability/Error:	7.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

User credentials are transmitted over an unencrypted channel. This information should always be transferred via an encrypted channel (HTTPS) to avoid being intercepted by malicious users.

4. Proof of concept and steps of verification of vulnerability with screenshots

POC:



Request to http://200.200.200.137:8005

```
POST /LoginAdmin HTTP/1.1
Host: 200.200.200.137:8005
Content-Length: 1910
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://200.200.200.137:8005
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.63 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://200.200.200.137:8005/LoginAdmin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: ASP.NET_SessionId=g4odlisu4nb2ihjmnueqsxv5
Connection: close

__LASTFOCUS=&__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=
b9qC0AZX%2FZ08TE6r6SsSj%2FHrw2H%2B7kjG9LH1Smf%2FjI49v5Sa6SYNfBU1TbnBh%2Fv3Pb6aPE5XXze3h7t1wa0THVuyqyncAy%2BsZgaPVu0PHf0ACeehYWvMVtQ4Y3y3
6gxNLGwv6PDXQ6HhMWjCYy2FyEu%2Faju5Walc7%2FA46VDNCo010VLGIGb9Fa4%2FtJPf%2B26HL%2BrjJZIPR8VshSKqWFgTDWP%2F7uUsUMTYKgXPzu9PZY%2BGRSrKvWVAZS
tv9gxvTcjguMzZ3INzUdjo61357XUDsvh2IdQrepLb%2B4EvxKa8Bycp2ZaMmLpF6bq0%2Bjj5IbR85S%2B5%2FBv%2FJ6LIEJ1Qr5Puq8ben0h8oICrwk7BLVxoq4CUyAXN8oYw
vSeB1751OvYJpVEkGN5nls0cMBFQPEZ0R701WYxAJW%2FEoUH507xF9Wgi57UGjdJKA9Q3IIvDgTbzSGr1b1396kiz0Qpk%2FCC9M5uI1U3cGZ0T1jDUmVmzRdojMqGvrvinQ1j%
2FVpz%2FAPFTMFm2ofHWeSFTyZJJVg6ohAGrIDjzJriADgeBExelPK1W%2FR5npnEUpSve8v22nKzDCLiupoUYUtHLdckxKCRA8BmlXuUegu0idrG1U%2FuKyD19JVr6NbvrmPb%
2B1kRQiVs8bBf0A39Ix%2BPmoruwvsQEPZn18QRDIS0a3U6e%2FalVyVw%3D&__VIEWSTATEGENERATOR=50E6E06E&__VIEWSTATEENCRYPTED=&__EVENTVALIDATION=
7tJvlmhA5%2FbHVHHr48YoMFSlmbvEF8Fw24I0BMRzjqtVxo2WCG09mAc8RqyuJnWfj5MZ9cGHNT0cPTWcGASQDNFSM4L3c170udf6rQ10i6eHkUIP0s10EdtPKd0AACdXBA%2Ff
uYKtD4ZzcGKiEv27olABmCreZo7ttuLyUa05Sriy79ahNRqZIvpWBHNfjCluraa2pXTmr60KXXhszK38d5eUNcvb9qVt%2FRn%2BjC5zHnQRc0ENduTUHefngsePkGiVzJBqXzxe
TLcBZC%2BngITjTtNXVWB28JX%2B84IBhXIgPJ4LiDn7ckmWwv3zciqBF1W81zFUP66fr60PC8Y4%2BWX1j699oaCJcA%2FgpDAwGFd0CeMp1Kp1W0NmH3U6mKDPvNHI1zSN41ql
gtg%2FsA0qA5A8mLrh0t0403fmDPQeu9xHeRdIw7JuvdIKSEgcVHT0ZzaYVqrIjSDFtMYclYr%2B3MGdNe9R8XCh%2FsqyAWnYw0awqZmsGsc%2F5zMiD%2BfaTZmlrAB%2F&
ctl00%24ContentPlaceHolder1%24txtUserName=&ctl00%24ContentPlaceHolder1%24txtPassowrd=&ctl00%24ContentPlaceHolder1%24txtCaptcha=&
ctl00%24ContentPlaceHolder1%24txtUserName1=9898789878&ctl00%24ContentPlaceHolder1%24txtPassword1=Amit%4012345&
ctl00%24ContentPlaceHolder1%24txtCaptcha1=E88CP&ctl00%24ContentPlaceHolder1%24btnLogin1=Login&ctl00%24ContentPlaceHolder1%24txtMobileNoP
=&ctl00%24ContentPlaceHolder1%24txtOTPP=&ctl00%24ContentPlaceHolder1%24txtPasswordP=&ctl00%24ContentPlaceHolder1%24txtCaptchaP=
```

5. Solution & Work around:

- Because user credentials are considered sensitive information, should always be transferred to the server over an encrypted connection (HTTPS).

6. References:

- **CWE-319: Clear text Transmission of Sensitive Information**, https://cwe.mitre.org/data/definitions/319.html

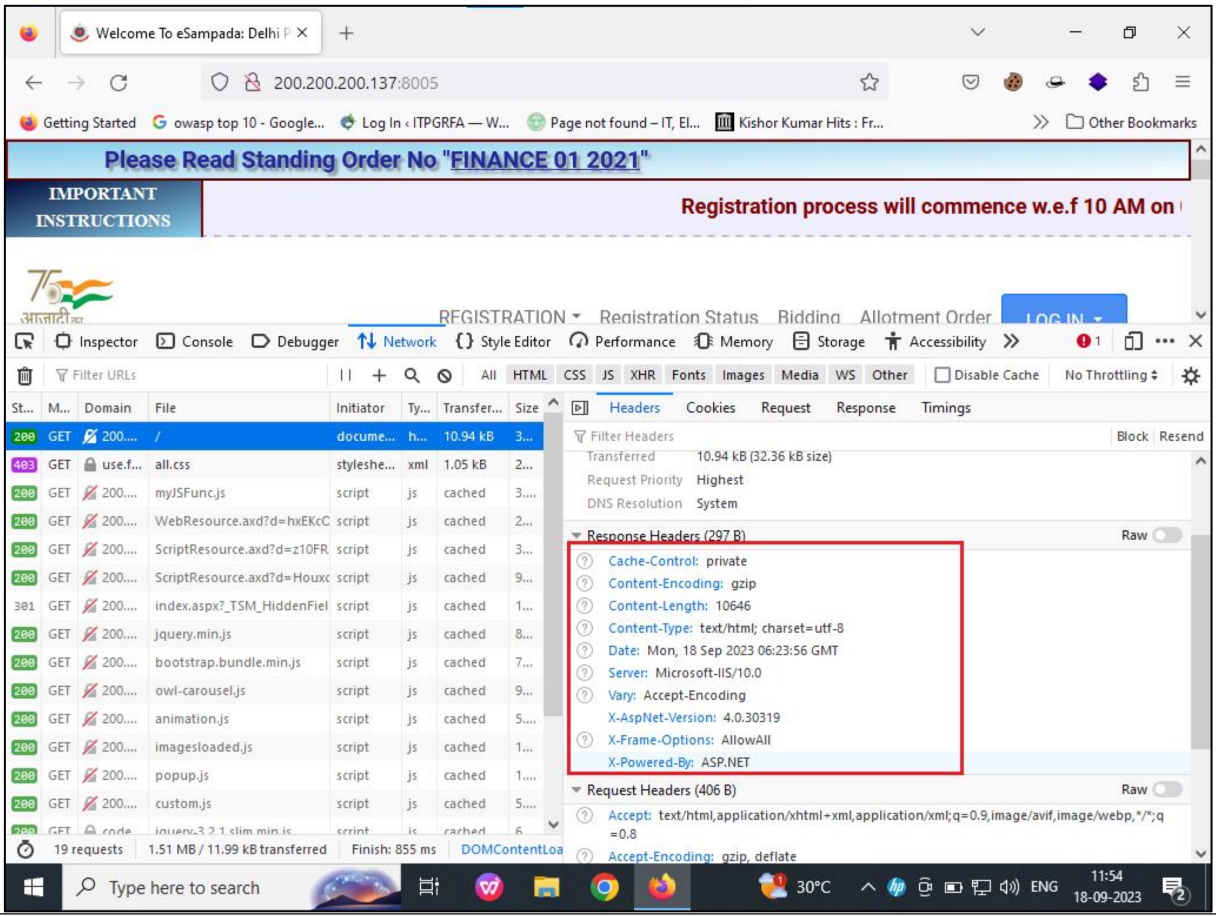
1. Name of Vulnerability/Error:	**8. X-Content-Type-Options Header Missing (Low)**
2. Location of Vulnerability/Error:	8.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This 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 declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing

4. Proof of concept and steps of verification of vulnerability with screenshots

POC:



St...	M...	Domain	File	Initiator	Ty...	Transfer...	Size
200	GET	200....	/	docume...	h...	10.94 kB	3...
403	GET	use.f...	all.css	styleshe...	xml	1.05 kB	2...
200	GET	200....	myJSFunc.js	script	js	cached	3....
200	GET	200....	WebResource.axd?d=hxEKcC	script	js	cached	2...
200	GET	200....	ScriptResource.axd?d=z10FR	script	js	cached	3...
200	GET	200....	ScriptResource.axd?d=Houxc	script	js	cached	9...
301	GET	200....	index.aspx?_TSM_HiddenFiel	script	js	cached	1...
200	GET	200....	jquery.min.js	script	js	cached	8...
200	GET	200....	bootstrap.bundle.min.js	script	js	cached	7...
200	GET	200....	owl-carousel.js	script	js	cached	9...
200	GET	200....	animation.js	script	js	cached	5....
200	GET	200....	imagesloaded.js	script	js	cached	1...
200	GET	200....	popup.js	script	js	cached	1....
200	GET	200....	custom.js	script	js	cached	5....
200	GET	code...	jquery-3.2.1.slim.min.js	script	js	cached	6...

19 requests | 1.51 MB / 11.99 kB transferred | Finish: 855 ms | DOMContentLoa

Headers | Cookies | Request | Response | Timings

```
Transferred        10.94 kB (32.36 kB size)
Request Priority   Highest
DNS Resolution     System

Response Headers (297 B)
Cache-Control: private
Content-Encoding: gzip
Content-Length: 10646
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Sep 2023 06:23:56 GMT
Server: Microsoft-IIS/10.0
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-Frame-Options: AllowAll
X-Powered-By: ASP.NET

Request Headers (406 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

6. References:

- **CWE-693: Protection Mechanism Failure**
  https://cwe.mitre.org/data/definitions/693.html

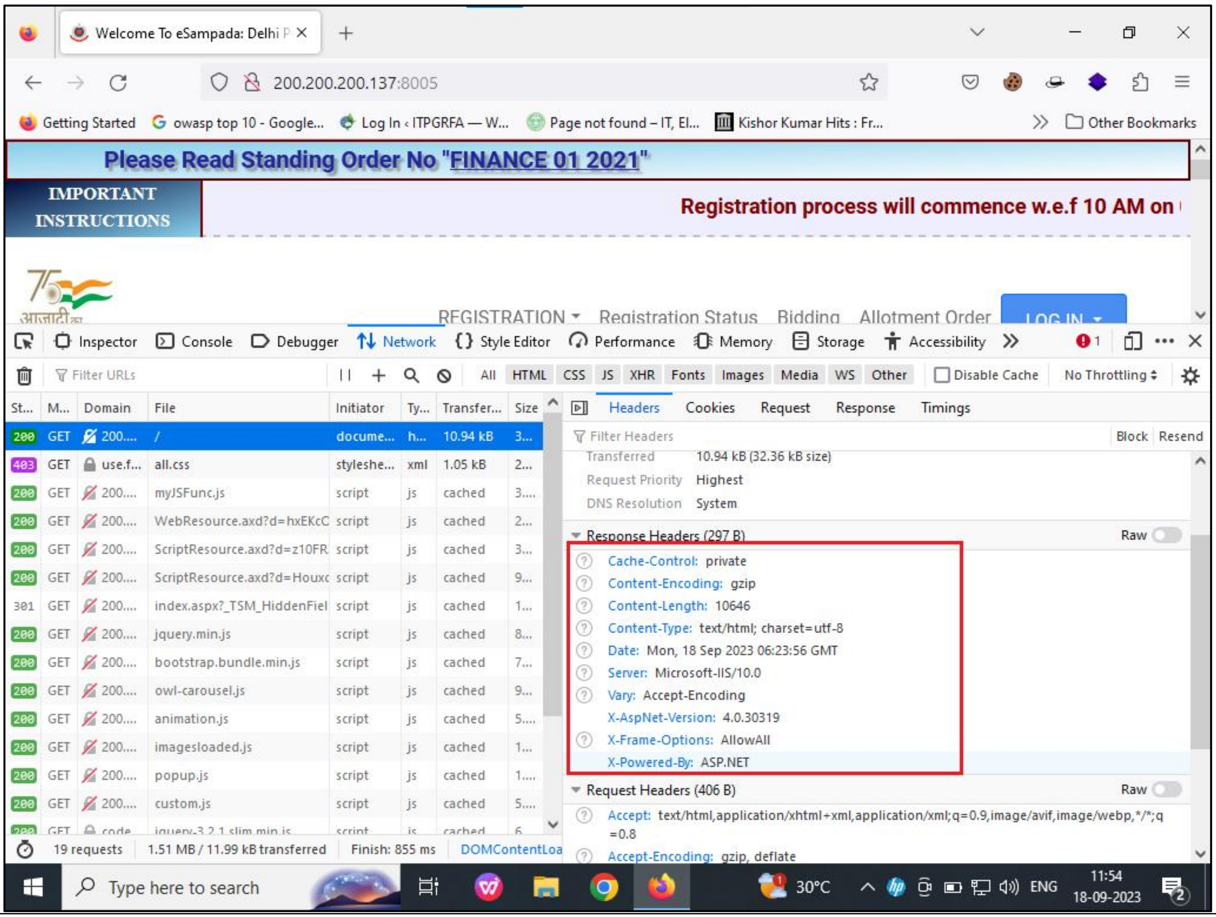
1. Name of Vulnerability/Error:	**9. HTTP Strict Transport Security (HSTS) Header Not Set (Low)**
2. Location of Vulnerability/Error	9.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

HTTP Strict Transport Security (HSTS) tells a browser that a web site is only accessable using HTTPS. It was detected that your web application doesn't implement HTTP Strict Transport Security (HSTS) as the Strict Transport Security header is missing from the response.

4. Proof of concept and steps of verification of vulnerability with screenshots:

POC:



St...	M...	Domain	File	Initiator	Ty...	Transfer...	Size
200	GET	200....	/	docume...	h...	10.94 kB	3...
403	GET	use.f...	all.css	styleshe...	xml	1.05 kB	2...
200	GET	200....	myJSFunc.js	script	js	cached	3....
200	GET	200....	WebResource.axd?d=hxEKcC	script	js	cached	2...
200	GET	200....	ScriptResource.axd?d=z10FR	script	js	cached	3...
200	GET	200....	ScriptResource.axd?d=Houxc	script	js	cached	9...
301	GET	200....	index.aspx?_TSM_HiddenFiel	script	js	cached	1...
200	GET	200....	jquery.min.js	script	js	cached	8...
200	GET	200....	bootstrap.bundle.min.js	script	js	cached	7...
200	GET	200....	owl-carousel.js	script	js	cached	9...
200	GET	200....	animation.js	script	js	cached	5....
200	GET	200....	imagesloaded.js	script	js	cached	1...
200	GET	200....	popup.js	script	js	cached	1....
200	GET	200....	custom.js	script	js	cached	5....
200	GET	code...	jquery-3.2.1.slim.min.js	script	js	cached	6...

```
19 requests   1.51 MB / 11.99 kB transferred   Finish: 855 ms   DOMContentLoa
```

```
Transferred          10.94 kB (32.36 kB size)
Request Priority     Highest
DNS Resolution       System

Response Headers (297 B)
  Cache-Control: private
  Content-Encoding: gzip
  Content-Length: 10646
  Content-Type: text/html; charset=utf-8
  Date: Mon, 18 Sep 2023 06:23:56 GMT
  Server: Microsoft-IIS/10.0
  Vary: Accept-Encoding
  X-AspNet-Version: 4.0.30319
  X-Frame-Options: AllowAll
  X-Powered-By: ASP.NET

Request Headers (406 B)
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
  =0.8
  Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- It's recommended to implement HTTP Strict Transport Security (HSTS) into your web application. Consult web references for more information

6. References:

- **CWE-523: Unprotected Transport of Credentials,** https://cwe.mitre.org/data/definitions/523.html

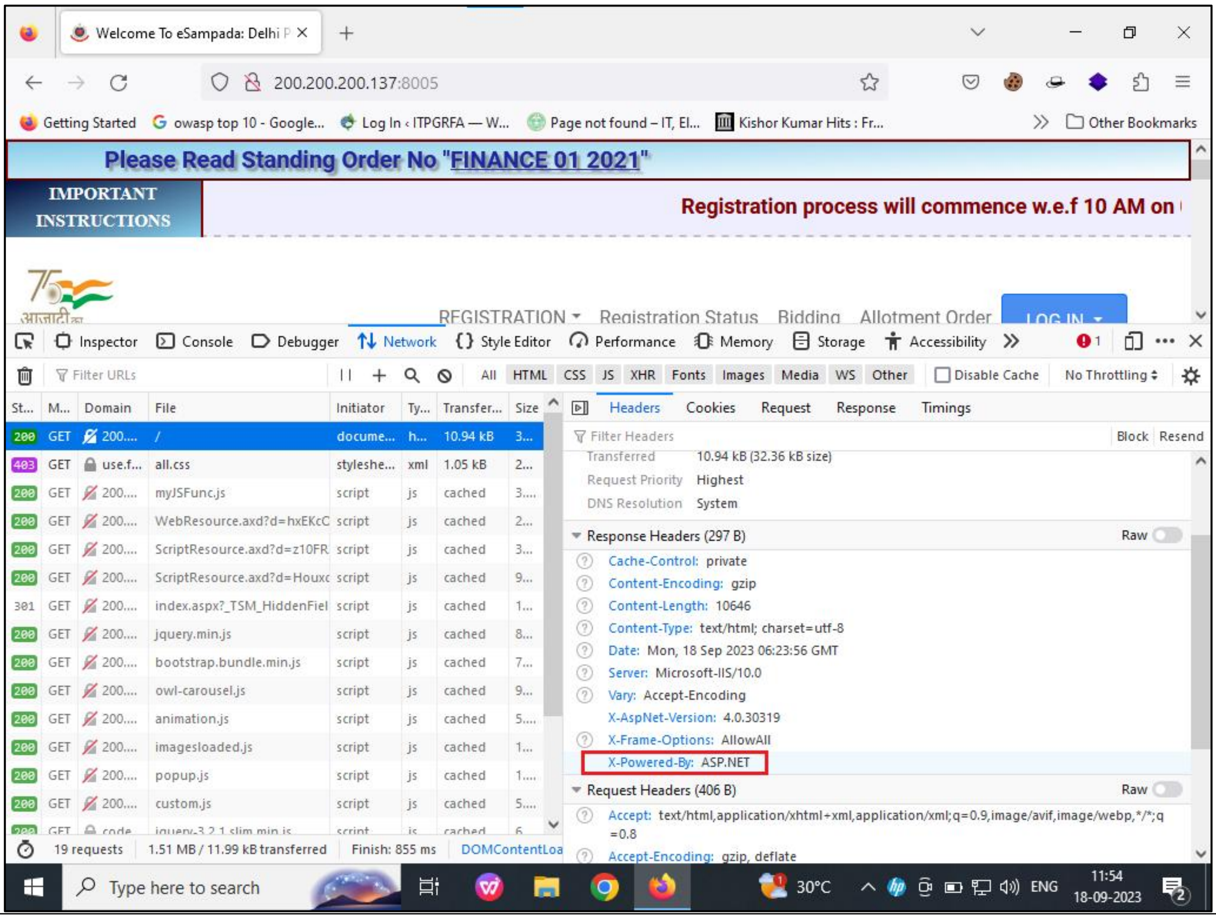
1. Name of Vulnerability/Error:	**10. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (Low)**
2. Location of Vulnerability/Error	10.1 [200.200.200.137/8005](200.200.200.137/8005)

3. Vulnerability Description:

The web/application server is leaking information via one or more "X-Powered-By" HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.

4. Proof of concept and steps of verification of vulnerability with screenshots

POC:



St...	M...	Domain	File	Initiator	Ty...	Transfer...	Size
200	GET	200....	/	docume...	h...	10.94 kB	3...
403	GET	use.f...	all.css	styleshe...	xml	1.05 kB	2...
200	GET	200....	myJSFunc.js	script	js	cached	3....
200	GET	200....	WebResource.axd?d=hxEKcC	script	js	cached	2...
200	GET	200....	ScriptResource.axd?d=z10FR	script	js	cached	3...
200	GET	200....	ScriptResource.axd?d=Houxc	script	js	cached	9...
301	GET	200....	index.aspx?_TSM_HiddenFiel	script	js	cached	1...
200	GET	200....	jquery.min.js	script	js	cached	8...
200	GET	200....	bootstrap.bundle.min.js	script	js	cached	7...
200	GET	200....	owl-carousel.js	script	js	cached	9...
200	GET	200....	animation.js	script	js	cached	5....
200	GET	200....	imagesloaded.js	script	js	cached	1...
200	GET	200....	popup.js	script	js	cached	1....
200	GET	200....	custom.js	script	js	cached	5....
200	GET	code...	jquery-3.2.1.slim.min.js	script	js	cached	6...

19 requests | 1.51 MB / 11.99 kB transferred | Finish: 855 ms | DOMContentLoa

```
Transferred          10.94 kB (32.36 kB size)
Request Priority     Highest
DNS Resolution       System

Response Headers (297 B)
Cache-Control: private
Content-Encoding: gzip
Content-Length: 10646
Content-Type: text/html; charset=utf-8
Date: Mon, 18 Sep 2023 06:23:56 GMT
Server: Microsoft-IIS/10.0
Vary: Accept-Encoding
X-AspNet-Version: 4.0.30319
X-Frame-Options: AllowAll
X-Powered-By: ASP.NET

Request Headers (406 B)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q
=0.8
Accept-Encoding: gzip, deflate
```

5. Solution & Work around:

- Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

6. References:

- **CWE-200: Exposure of Sensitive Information to an Unauthorized Actor,** https://cwe.mitre.org/data/definitions/200.html

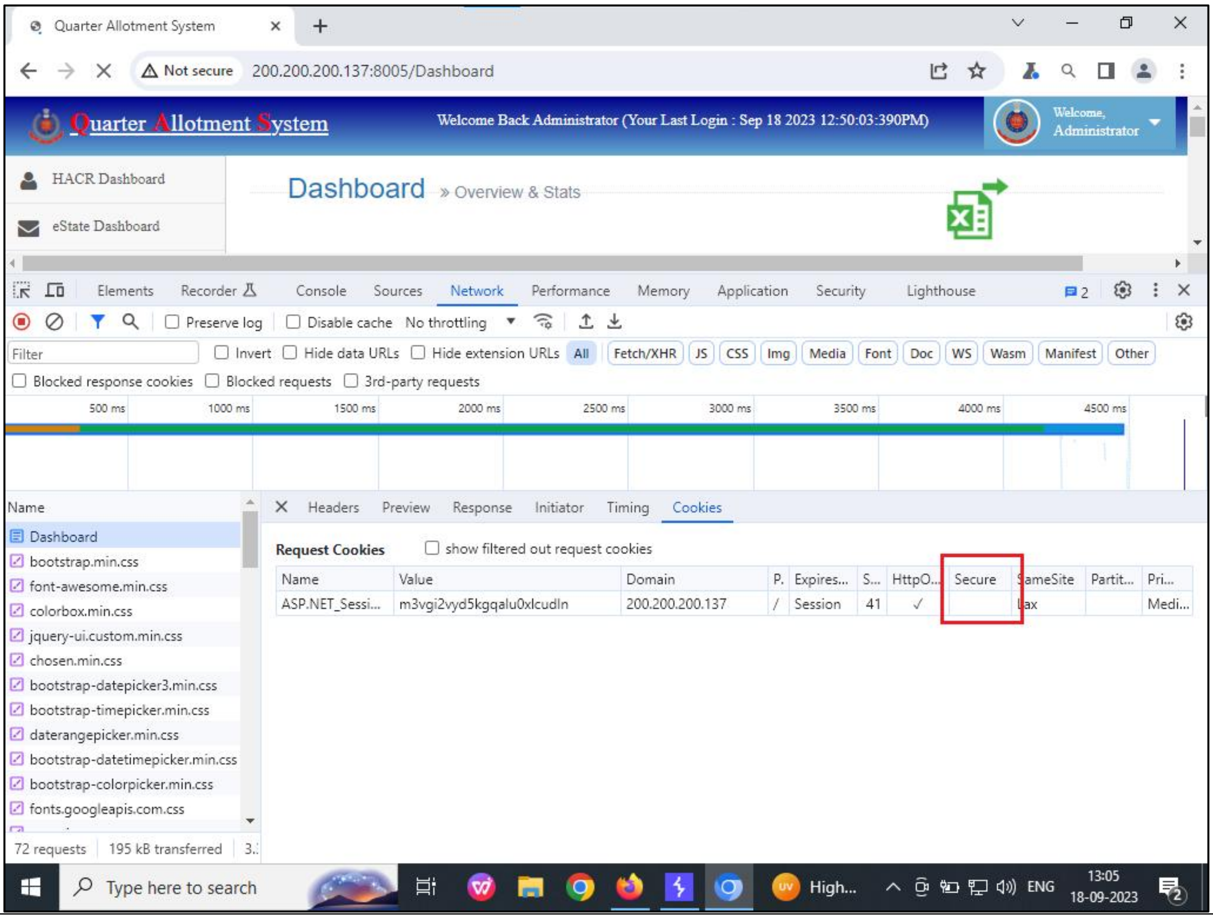
1. Name of Vulnerability/Error:	**11. Cookie Without Secure Flag (Low)**
2. Location of Vulnerability/Error	11.1 200.200.200.137/8005/Dashboard

3. Vulnerability Description:

During audit it was observed that a cookie has been set without the secure flag, which means that the cookie can be accessed via unencrypted connections.

4. Proof of concept and steps of verification of vulnerability with screenshots:

POC:



Name
- Dashboard
- bootstrap.min.css
- font-awesome.min.css
- colorbox.min.css
- jquery-ui.custom.min.css
- chosen.min.css
- bootstrap-datepicker3.min.css
- bootstrap-timepicker.min.css
- daterangepicker.min.css
- bootstrap-datetimepicker.min.css
- bootstrap-colorpicker.min.css
- fonts.googleapis.com.css

72 requests | 195 kB transferred

Request Cookies ☐ show filtered out request cookies

Name	Value	Domain	P.	Expires...	S...	HttpO...	Secure	SameSite	Partit...	Pri...
ASP.NET_Sessi...	m3vgi2vyd5kgqalu0xlcudln	200.200.200.137	/	Session	41	✓		Lax		Medi...

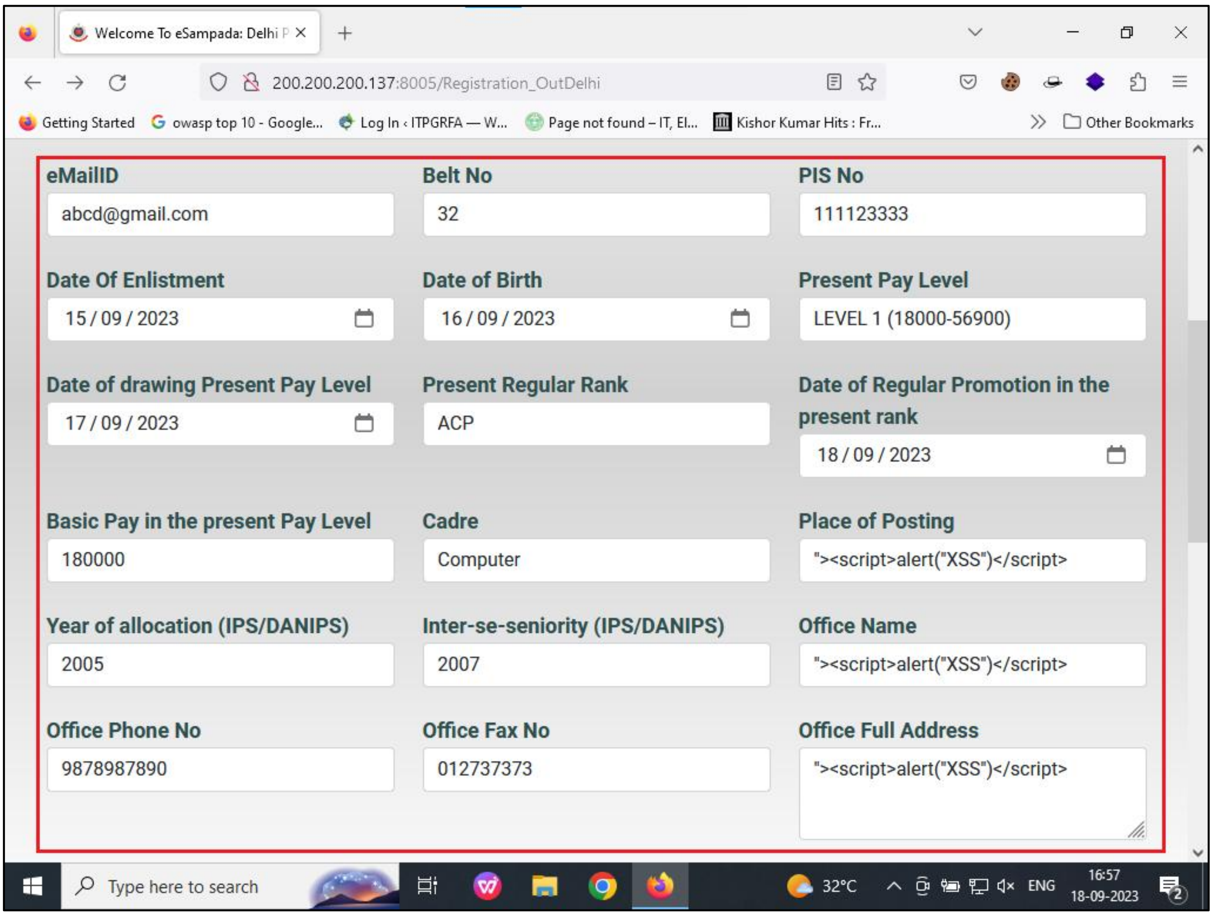
5. Solution & Work around:

- Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.

6. References:

- **CWE-614: Sensitive Cookie in HTTPS Session Without 'Secure' Attribute,** https://cwe.mitre.org/data/definitions/614.html

Functional Issue: Form was not being submitted



Welcome To eSampada: Delhi P
200.200.200.137:8005/Registration_OutDelhi
Getting Started
owasp top 10 - Google...
Log In ‹ ITPGRFA — W...
Page not found – IT, El...
Kishor Kumar Hits : Fr...
Other Bookmarks

eMailID	Belt No	PIS No
abcd@gmail.com	32	111123333
Date Of Enlistment	Date of Birth	Present Pay Level
15 / 09 / 2023	16 / 09 / 2023	LEVEL 1 (18000-56900)
Date of drawing Present Pay Level	Present Regular Rank	Date of Regular Promotion in the present rank
17 / 09 / 2023	ACP	18 / 09 / 2023
Basic Pay in the present Pay Level	Cadre	Place of Posting
180000	Computer	"><script>alert("XSS")</script>
Year of allocation (IPS/DANIPS)	Inter-se-seniority (IPS/DANIPS)	Office Name
2005	2007	"><script>alert("XSS")</script>
Office Phone No	Office Fax No	Office Full Address
9878987890	012737373	"><script>alert("XSS")</script>