Risk Category

The category describes the risk level of the vulnerability which is derived as follows. These risk categories have been defined by Skillmine' s vast experience in system security testing, basedon available Common Vulnerabilities and Exposures (CVEs):

Risk Level	Description
High	This risk level indicates that successful exploitation of the vulnerability may result in a significant impact on the information accessible through the server or even the backend resources like databases, operating systems, etc. It may also lead to damage to reputation.
Medium	This risk level indicates that successful exploitation of the vulnerability may reveal information about the server and its underlying infrastructure that may be used by an attacker in conjunction with another vulnerability to gain further access.
Low	This risk level indicates that successful exploitation of the vulnerability may result in little or no loss of sensitive information but may enable an attacker to gain enough information regarding the server and its underlying infrastructure, which he/she may use to narrow down the attack approach.

Quick View

The table below is designed to provide a quick view of all the identified findings and their respective risk rating. Please see the following section for a detailed listing of the identified findings:

S no.	Issue found	Impact	Risk
1.	Missing Web Application Firewall	The impact of the missing vulnerability in the WAF could be severe, leading to unauthorized access, data breaches, injection attacks, and other web application vulnerabilities. Attackers may exploit this weakness to bypass security controls, execute malicious code, or gain unauthorized access to sensitive information. Additionally, the website may be vulnerable to common web attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), posing a significant risk to both the website's users and the organization's data.	HIGH
2.	Vulnerable Version of the Library 'jQuery' Found	Affected by multiple cross site scripting vulnerabilities.	MEDIUM
3.	HSTS not enabled	The HTTP Strict Transport Security policy defines a timeframe where a browser must connect to the web server via HTTPS. Without a Strict Transport Security policy, the web application is vulnerable against several attacks: If the web application mixes usage of HTTP and HTTPS, an attacker can manipulate pages in the unsecured area of the application or change redirection targets in a manner that the switch to the secured page is not performed or done in a manner, that the attacker remains between client and server. If there is no HTTP server, an attacker in the same network could simulate a HTTP server and motivate the user to click on a prepared URL by a social engineering attack. The protection is effective only for the given amount of time. Multiple occurrences of this header could cause undefined behaviour in browsers and should be avoided.	LOW

4.	Missing Content Security Policy	The vulnerability arising from the absence of a Content Security Policy exposes our web application to heightened risks of XSS attacks, data manipulation, and unauthorized script executions. Without a CSP in place, malicious actors can exploit these vulnerabilities, potentially compromising user data, session integrity, and the overall confidentiality of our system. Furthermore, the lack of content restrictions may lead to the injection of harmful scripts, undermining the trust and reliability of our web services.	LOW
5.	Cookie without HttpOnly flag set	The cookie appears to contain a session token, which may increase the risk associated with this issue.	LOW
6.	Missing X-Frame-Options Header	Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top-level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both. Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account but are instead typing into an invisible frame controlled by the attacker.	LOW
7.	Referrer-Policy Not Implemented	Referrer header is a request header that indicates the site which the traffic originated from. If there is no adequate prevention in place, the URL itself, and even sensitive information contained in the URL will be leaked to the cross-site. The lack of Referrer-Policy header might affect privacy of the users and sites itself.	LOW
8.	Missing 'X-Content-Type-Options' Header	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 allows older versions of Internet Explorer and Chrome to perform MIME-	LOW

		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. The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.	
9.	Web browser XSS protection not enabled	Some browsers, including Internet Explorer, contain built-in filters designed to protect against cross-site scripting (XSS) attacks. Applications can instruct browsers to disable this filter by setting the following response header: • X-XSS-Protection: 0 This behavior does not in itself constitute a vulnerability; in some cases, XSS filters may themselves be leveraged to perform attacks against application users. However, in typical situations XSS filters do provide basic protection for application users against some XSS vulnerabilities in applications. The presence of this header should be reviewed to establish whether it affects the application's security posture.	LOW
10.	Content type incorrectly stated	If a response specifies an incorrect content type, then browsers may process the response in unexpected ways. If the content type is specified to be a renderable text-based format, then the browser will usually attempt to interpret the response as being in that format, regardless of the actual contents of the response. Additionally, some other specified content types might sometimes be interpreted as HTML due to quirks in particular browsers. This behavior might lead to otherwise "safe" content such as images being rendered as HTML, enabling cross-site scripting attacks in certain conditions. The presence of an incorrect content type statement typically only constitutes a security flaw when the affected resource is dynamically generated, uploaded by a user, or otherwise contains user input. You should	LOW

		review the contents of affected responses, and the context in which they appear, to determine whether any vulnerability exists.	

Detailed Vulnerability Information and Recommendations

The detailed explanation of the vulnerabilities as well as the recommendations for the same are explained below:

Issue #01 - Missing Web Application Firewall | HIGH

Abstract

We observed no Web Application Firewall on the website.

Path

/

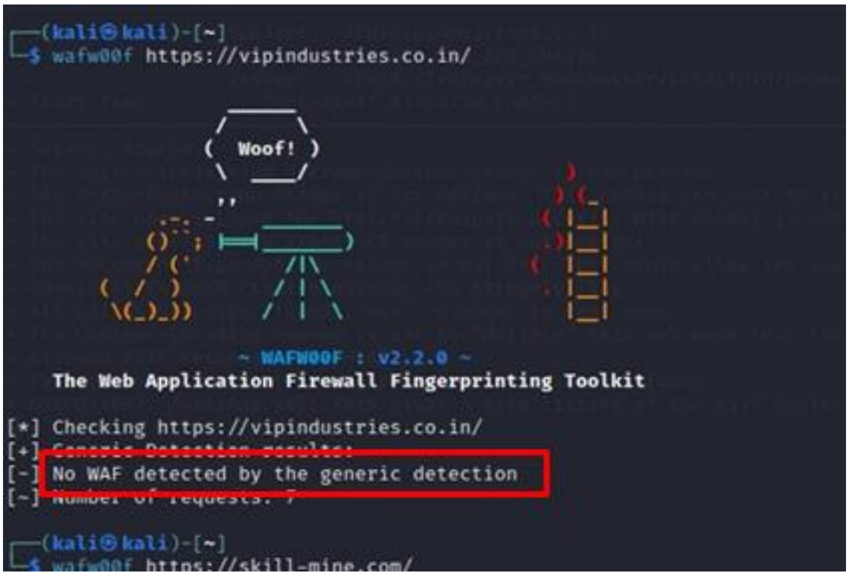
Impact

The impact of the missing vulnerability in the WAF could be severe, leading to unauthorized access, data breaches, injection attacks, and other web application vulnerabilities. Attackers may exploit this weakness to bypass security controls, execute malicious code, or gain unauthorized access to sensitive information. Additionally, the website may be vulnerable to common web attacks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), posing a significant risk to both the website's users and the organization's data.

Recommendation

Implement and configure Web Application Firewall.

Proof of Concept:



```
┌──(kali㉿kali)-[~]
└─$ wafw00f https://vipindustries.co.in/

                ~ WAFW00F : v2.2.0 ~
  The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://vipindustries.co.in/
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[-] Number of requests: 7

┌──(kali㉿kali)-[~]
└─$ wafw00f https://skill-mine.com/
```

Issue #02 - Vulnerable Version of the Library ‘jQuery’ Found | MEDIUM

Abstract

Cross Site Scripting vulnerability in jQuery before 3.5.0 allows a remote attacker to execute arbitrary code via the <options> element.

Path

/assets/js/all.min.js

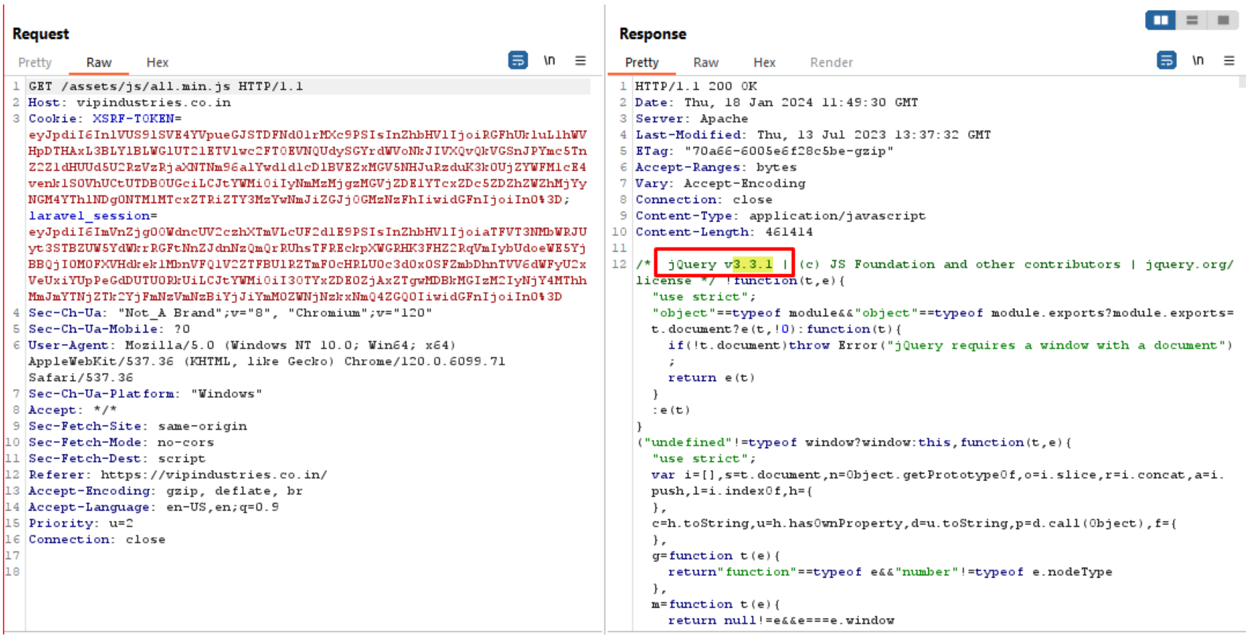
Impact

Affected by multiple cross site scripting vulnerabilities.

Recommendation

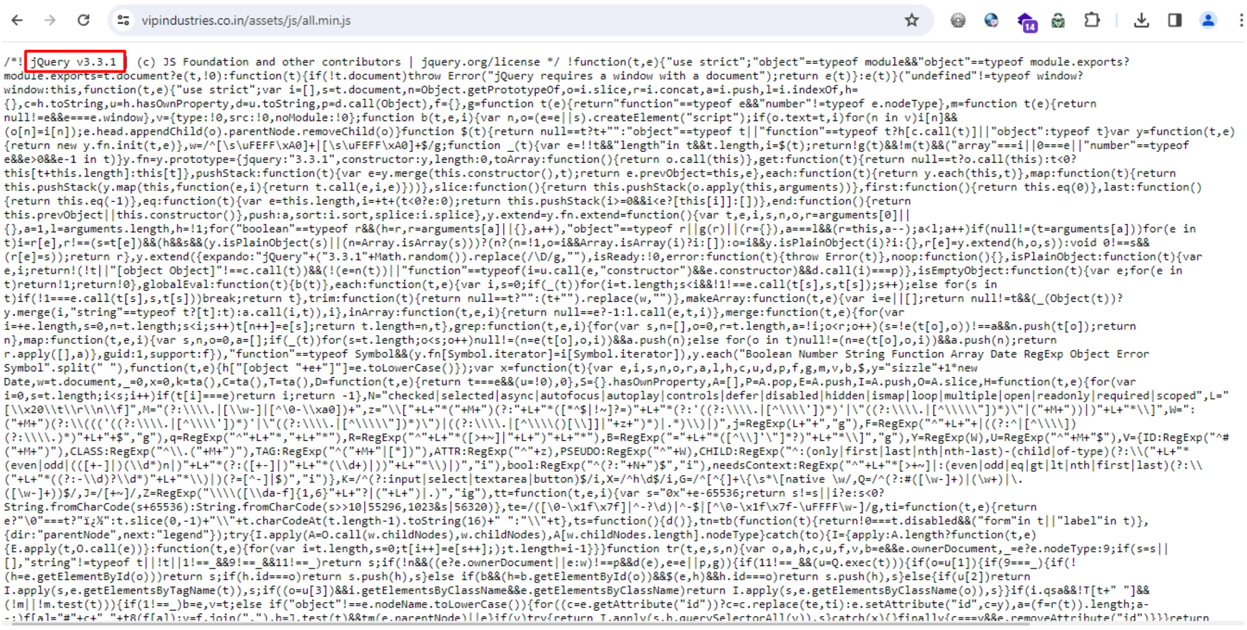
Upgrade to JQuery version 3.5.0 or later.

Proof of Concept:



```
GET /assets/js/all.min.js HTTP/1.1
Host: vipindustries.co.in
Cookie: XSRF-TOKEN=
eyJpdiI6InlVUS91SVE4YVpueGJSTDFNd01rMXc9PSIsInZhbHVlIjoiRGFhUkluL1hWV
HpDTHAxL3BLY1BLWGlUT21ETVlwc2FT0EVNQUdySGYrdWVoNkJIVXQvQkVGSnJPYmc5Tn
Z2Z1dHUUd5U2RzVzRjaXNTNm96alYwd1dlcDlBVEZxMGV5NHJuRzduK3k0UjZYWFMlcE4
venk1S0VhUCtUTDB0UGciLCJtYWMi0iIyNmMzMjgzMGVjZDE1YTcxZDc5ZDZhZWZhMjYy
NGM4YTh1NDg0NTM1MTcxZTRiZTY3MzYwNmJiZGJj0GMzNzFhIiwidGFnIjoiIn0%3D;
laravel_session=
eyJpdiI6ImVnZjg0OWdncUV2czhXTmVLcUF2d1E9PSIsInZhbHVlIjoiaTFVT3NMbWRJU
yt3STBZUW5YdWkrRGFtNnZJdnNzQmQrRUhsTFREckpXWGRHK3FHZ2RqVmIybUdoeWE5Yj
BBQjI0M0FXVHdkek1MbnVFQ1V2ZTFBU1RZTmF0cHRLU0c3d0x0SFZmbDhnTVV6dWFyU2x
VeUxiYUpPeGdDUTU0RkUiLCJtYWMi0iI30TYxZDE0ZjAxZTgwMDBkMGIzM2IyNjY4MThh
MmJmYTNjZTk2YjFmNzVmNzBiYjJiYmM0ZWNjNzkxNmQ4ZGQ0IiwidGFnIjoiIn0%3D
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer: https://vipindustries.co.in/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=2
Connection: close


```

```
HTTP/1.1 200 OK
Date: Thu, 18 Jan 2024 11:49:30 GMT
Server: Apache
Last-Modified: Thu, 13 Jul 2023 13:37:32 GMT
ETag: "70a66-6005e6f28c5be-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Connection: close
Content-Type: application/javascript
Content-Length: 461414

/*  jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.org/
license */ !function(t,e){
  "use strict";
  "object"==typeof module&&"object"==typeof module.exports?module.exports=
  t.document?e(t,!0):function(t){
    if(!t.document)throw Error("jQuery requires a window with a document")
    ;
    return e(t)
  }
  :e(t)
}
("undefined"!=typeof window?window:this,function(t,e){
  "use strict";
  var i=[],s=t.document,n=Object.getPrototypeOf,o=i.slice,r=i.concat,a=i.
  push,l=i.indexOf,h={
  },
  c=h.toString,u=h.hasOwnProperty,d=u.toString,p=d.call(Object),f={
  },
  g=function t(e){
    return"function"==typeof e&&"number"!=typeof e.nodeType
  },
  m=function t(e){
    return null!=e&&e===e.window
```



```
/*! jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.org/license */ !function(t,e){"use strict";"object"==typeof module&&"object"==typeof module.exports?
module.exports=t.document?e(t,!0):function(t){if(!t.document)throw Error("jQuery requires a window with a document");return e(t)}:e(t)}("undefined"!=typeof window?
window:this,function(t,e){"use strict";var i=[],s=t.document,n=Object.getPrototypeOf,o=i.slice,r=i.concat,a=i.push,l=i.indexOf,h=
{},c=h.toString,u=h.hasOwnProperty,d=u.toString,p=d.call(Object),f={},g=function t(e){return"function"==typeof e&&"number"!=typeof e.nodeType},m=function t(e){return
null!=e&&e===e.window},v={type:!0,src:!0,noModule:!0};function b(t,e,i){var n,o=(e=e||s).createElement("script");if(o.text=t,i)for(n in v)i[n]&&
(o[n]=i[n]);e.head.appendChild(o).parentNode.removeChild(o)}function $(t){return null==t?t+"":"object"==typeof t||"function"==typeof t?h[c.call(t)]||"object":typeof t}var y=function(t,e)
{return new y.fn.init(t,e)},w=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g;function _(t){var e=!!t&&"length"in t&&t.length,i=$(t);return!g(t)&&!m(t)&&("array"===i||0===e||"number"==typeof
e&&e>0&&e-1 in t)}y.fn=y.prototype={jquery:"3.3.1",constructor:y,length:0,toArray:function(){return o.call(this)},get:function(t){return null==t?o.call(this):t<0?
this[t+this.length]:this[t]},pushStack:function(t){var e=y.merge(this.constructor(),t);return e.prevObject=this,e},each:function(t){return y.each(this,t)},map:function(t){return
this.pushStack(y.map(this,function(e,i){return t.call(e,i,e)}))},slice:function(){return this.pushStack(o.apply(this,arguments))},first:function(){return this.eq(0)},last:function()
{return this.eq(-1)},eq:function(t){var e=this.length,i=+t+(t<0?e:0);return this.pushStack(i>=0&&i<e?[this[i]]:[])},end:function(){return
this.prevObject||this.constructor()},push:a,sort:i.sort,splice:i.splice},y.extend=y.fn.extend=function(){var t,e,i,s,n,o,r=arguments[0]||
{},a=1,l=arguments.length,h=!1;for("boolean"==typeof r&&(h=r,r=arguments[a]||{},a++),"object"==typeof r||g(r)||(r={}),a===l&&(r=this,a--);a<l;a++)if(null!=(t=arguments[a]))for(e in
t)i=r[e],r!==(s=t[e])&&(h&&s&&(y.isPlainObject(s)||(n=Array.isArray(s)))?(n?(n=!1,o=i&&Array.isArray(i)?i:[]):o=i&&y.isPlainObject(i)?i:{},r[e]=y.extend(h,o,s)):void 0!==s&&
(r[e]=s));return r},y.extend({expando:"jQuery"+("3.3.1"+Math.random()).replace(/\D/g,""),isReady:!0,error:function(t){throw Error(t)},noop:function(){},isPlainObject:function(t){var
e,i;return!(!t||"[object Object]"!==c.call(t))&&(!(e=n(t))||"function"==typeof(i=u.call(e,"constructor")&&e.constructor)&&d.call(i)===p)},isEmptyObject:function(t){var e;for(e in
t)return!1;return!0},globalEval:function(t){b(t)},each:function(t,e){var i,s=0;if(_(t))for(i=t.length;s<i&&!1!==e.call(t[s],s,t[s]);s++);else for(s in
t)if(!1===e.call(t[s],s,t[s]))break;return t},trim:function(t){return null==t?"":(t+"").replace(w,"")},makeArray:function(t,e){var i=e||[];return null!=t&&(_(Object(t))?
y.merge(i,"string"==typeof t?[t]:t):a.call(i,t)),i},inArray:function(t,e,i){return null==e?-1:l.call(e,t,i)},merge:function(t,e){for(var
i=+e.length,s=0,n=t.length;s<i;s++)t[n++]=e[s];return t.length=n,t},grep:function(t,e,i){for(var s,n=[],o=0,r=t.length,a=!i;o<r;o++)(s=!e(t[o],o))!==a&&n.push(t[o]);return
n},map:function(t,e,i){var s,n,o=0,a=[];if(_(t))for(s=t.length;o<s;o++)null!=(n=e(t[o],o,i))&&a.push(n);else for(o in t)null!=(n=e(t[o],o,i))&&a.push(n);return
r.apply([],a)},guid:1,support:f}),"function"==typeof Symbol&&(y.fn[Symbol.iterator]=i[Symbol.iterator]),y.each("Boolean Number String Function Array Date RegExp Object Error
Symbol".split(" "),function(t,e){h["[object "+e+"]"]=e.toLowerCase()});var x=function(t){var e,i,s,n,o,r,a,l,h,c,u,d,p,f,g,m,v,b,$,y="sizzle"+1*new
Date,w=t.document,_=0,x=0,k=ta(),C=ta(),T=ta(),D=function(t,e){return t===e&&(u=!0),0},S={}.hasOwnProperty,A=[],P=A.pop,E=A.push,I=A.push,O=A.slice,H=function(t,e){for(var
i=0,s=t.length;i<s;i++)if(t[i]===e)return i;return -1},N="checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|ismap|loop|multiple|open|readonly|required|scoped",L="
[\\x20\\t\\r\\n\\f]",M="(?:\\\\.|[\\w-]|[^\0-\\xa0])+",z="\\["+L+"*("+M+")(?:"+L+"*([*^$|!~]?=)"+L+"*(?:'((?:\\\\.|[^\\\\'])*)'|\"((?:\\\\.|[^\\\\\"])*)\"|("+M+"))|)"+L+"*\\]",W=":
("+M+")(?:\\((('((?:\\\\.|[^\\\\'])*)'|\"((?:\\\\.|[^\\\\\"])*)\")|((?:\\\\.|[^\\\\()[\\]]|"+z+")*)|.*)\\)|)",j=RegExp(L+"+","g"),F=RegExp("^"+L+"+|((?:^|[^\\\\])
(?:\\\\.)*)"+L+"+$","g"),q=RegExp("^"+L+"*,"+L+"*"),R=RegExp("^"+L+"*([>+~]|"+L+")"+L+"*"),B=RegExp("="+L+"*([^\\]'\"]*?)"+L+"*\\]","g"),Y=RegExp(W),U=RegExp("^"+M+"$"),V={ID:RegExp("^#
("+M+")"),CLASS:RegExp("^\\.("+M+")"),TAG:RegExp("^("+M+"|[*])"),ATTR:RegExp("^"+z),PSEUDO:RegExp("^"+W),CHILD:RegExp("^:(only|first|last|nth|nth-last)-(child|of-type)(?:\\("+L+"*
(even|odd|(([+-]|)(\\d*)n|)"+L+"*(?:([+-]|)"+L+"*(\\d+)|))"+L+"*\\)|)","i"),bool:RegExp("^(?:"+N+")$","i"),needsContext:RegExp("^"+L+"*[>+~]|:(even|odd|eq|gt|lt|nth|first|last)(?:\\
("+L+"*((?:-\\d)?\\d*)"+L+"*\\)|)(?=[^-]|$)","i")},K=/^(?:input|select|textarea|button)$/i,X=/^h\d$/i,G=/^[^{]+\{\s*\[native \w/,Q=/^(?:#([\w-]+)|(\w+)|\.
([\w-]+))$/,J=/[+~]/,Z=RegExp("\\\\([\\da-f]{1,6}"+L+"?|("+L+")|.)","ig"),tt=function(t,e,i){var s="0x"+e-65536;return s!=s||i?e:s<0?
String.fromCharCode(s+65536):String.fromCharCode(s>>10|55296,1023&s|56320)},te=/([\0-\x1f\x7f]|^-?\d)|^-$|[^\0-\x1f\x7f-\uFFFF\w-]/g,ti=function(t,e){return
e?"\0"===t?"ï¿½":t.slice(0,-1)+"\\"+t.charCodeAt(t.length-1).toString(16)+" ":"\\"+t},ts=function(){d()},tn=tb(function(t){return!0===t.disabled&&("form"in t||"label"in t)},
{dir:"parentNode",next:"legend"});try{I.apply(A=O.call(w.childNodes),w.childNodes),A[w.childNodes.length].nodeType}catch(to){I={apply:A.length?function(t,e)
{E.apply(t,O.call(e))}:function(t,e){for(var i=t.length,s=0;t[i++]=e[s++];);t.length=i-1}}}function tr(t,e,s,n){var o,a,h,c,u,f,v,b=e&&e.ownerDocument,_=e?e.nodeType:9;if(s=s||
[],"string"!=typeof t||!t||1!==_&&9!==_&&11!==_)return s;if(!n&&((e?e.ownerDocument||e:w)!==p&&d(e),e=e||p,g)){if(11!==_&&(u=Q.exec(t))){if(o=u[1]){if(9===_){if(!
(h=e.getElementById(o)))return s;if(h.id===o)return s.push(h),s}else if(b&&(h=b.getElementById(o))&&$(e,h)&&h.id===o)return s.push(h),s}else{if(u[2])return
I.apply(s,e.getElementsByTagName(t)),s;if((o=u[3])&&i.getElementsByClassName&&e.getElementsByClassName)return I.apply(s,e.getElementsByClassName(o)),s}if(i.qsa&&!T[t+" "]&&
(!m||!m.test(t))){if(1!==_)b=e,v=t;else if("object"!==e.nodeName.toLowerCase()){for((c=e.getAttribute("id"))?c=c.replace(te,ti):e.setAttribute("id",c=y),a=(f=r(t)).length;a-
-;)f[a]="#"+c+" "+t8(f[a]);v=f.join(","),h=J.test(t)&&tm(e.parentNode)||e}if(v)try{return I.apply(s,h.querySelectorAll(v)),s}catch(x){}finally{c===y&&e.removeAttribute("id")}}}return
```

Issue #03 - HSTS not enabled | LOW

Abstract

There was no "Strict-Transport-Security" header in the server response at multiple instances.

Path

/
/assets/img/find-bg.webp
/assets/img/bt-arrow.png
/assets/img/02.webp
/assets/img/dlip.webp
/assets/img/News&Media_HomePage_04.webp
/assets/img/News&Media_HomePage_03.webp
/robots.txt

Impact

The HTTP Strict Transport Security policy defines a timeframe where a browser must connect to the web server via HTTPS. Without a Strict Transport Security policy, the web application is vulnerable against several attacks:

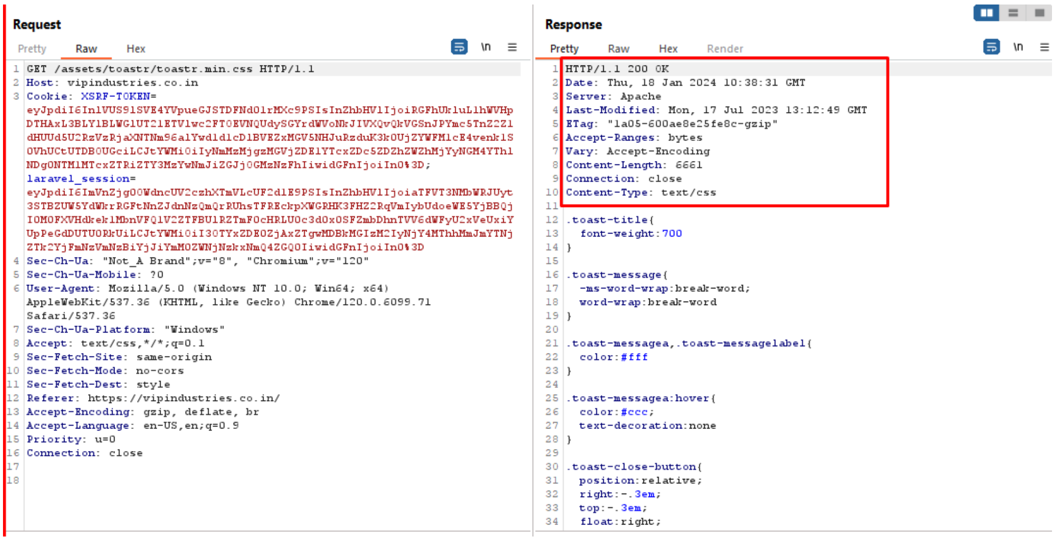
- If the web application mixes usage of HTTP and HTTPS, an attacker can manipulate pages in the unsecured area of the application or change redirection targets in a manner that the switch to the secured page is not performed or done in a manner, that the attacker remains between client and server.
- If there is no HTTP server, an attacker in the same network could simulate a HTTP server and motivate the user to click on a prepared URL by a social engineering attack. The protection is effective only for the given amount of time. Multiple occurrences of this header could cause undefined behaviour in browsers and should be avoided.

Recommendation

A Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

Strict-Transport-Security: max-age=<seconds> [; includeSubDomains]. The parameter max-age gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g., several months. A value below 7776000 is considered too low. The flag includeSubDomains defines that the policy applies also for sub domains of the sender of the response.

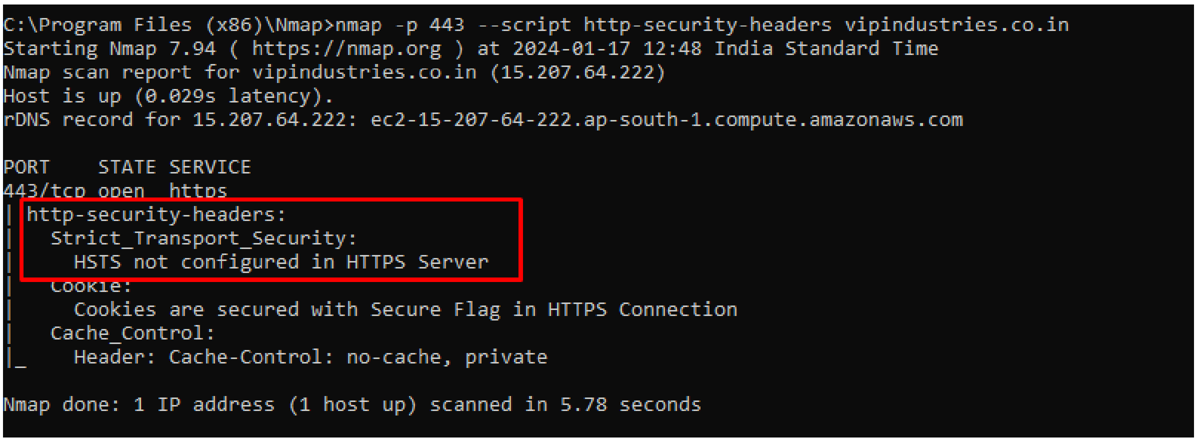
Proof of Concept:



```
GET /assets/toastr/toastr.min.css HTTP/1.1
Host: vipindustries.co.in
Cookie: XSRF-TOKEN=
eyJpdiI6InlVUS9lSVE4YVpueGJSTDFNd0lrMXc9PSIsInZhbHVlIjoiRGFhUkluLlhWVHp
DTHAxL3BLYlBLWGlUT2lETVlwc2FTOEVNQUdySGYrdWVoNkJIVXQvQkVGSnJPYmc5TnZ2Zl
dHUUd5U2RzVzRjaXNTNm96alYwdldlcDlBVEZxMGV5NHJuRzduK3k0UjZYWFMlcE4venklS
OVhUCtUTDB0UGciLCJtYWMi0iIyNmMzMjgzMGVjZDE1YTcxZDc5ZDZhZWZhMjYyNGM4YThl
NDg0NTM1MTcxZTRiZTY3MzYwNmJiZGJj0GMzNzFhIiwidGFnIjoiIn0%3D;
laravel_session=
eyJpdiI6ImVnZjg00WdncUV2czhXTmVLcUF2dlE9PSIsInZhbHVlIjoiaTFVT3NMbWRJUyt
3STBZUW5YdWkrRGFtNnZJdnNzQmQrRUhsTFREckpXWGRHK3FHZ2RqVmIybUdoeWE5YjBBQj
I0M0FXVHdkeklMbnVFQ1V2ZTFBUlRZTmF0cHRLU0c3d0x0SFZmbDhnTVV6dWFyU2xVeUxiY
UpPeGdDUTU0RkUiLCJtYWMi0iI30TYxZDE0ZjAxZTgwMDBkMGIzM2IyNjY4MThhMmJmYTNj
ZTk2YjFmNzVmNzBiYjJiYmM0ZWNjNzkxNmQ4ZGQ0IiwidGFnIjoiIn0%3D
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: https://vipindustries.co.in/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0
Connection: close
```

```
HTTP/1.1 200 OK
Date: Thu, 18 Jan 2024 10:38:31 GMT
Server: Apache
Last-Modified: Mon, 17 Jul 2023 13:12:49 GMT
ETag: "1a05-600ae8e25fe8c-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 6661
Connection: close
Content-Type: text/css

.toast-title{
  font-weight:700
}

.toast-message{
  -ms-word-wrap:break-word;
  word-wrap:break-word
}

.toast-messagea,.toast-messagelabel{
  color:#fff
}

.toast-messagea:hover{
  color:#ccc;
  text-decoration:none
}

.toast-close-button{
  position:relative;
  right:-.3em;
  top:-.3em;
  float:right;
```

HTSTS is missing in the header.



```
C:\Program Files (x86)\Nmap>nmap -p 443 --script http-security-headers vipindustries.co.in
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-17 12:48 India Standard Time
Nmap scan report for vipindustries.co.in (15.207.64.222)
Host is up (0.029s latency).
rDNS record for 15.207.64.222: ec2-15-207-64-222.ap-south-1.compute.amazonaws.com

PORT    STATE SERVICE
443/tcp open  https
| http-security-headers:
|   Strict_Transport_Security:
|     HSTS not configured in HTTPS Server
|   Cookie:
|     Cookies are secured with Secure Flag in HTTPS Connection
|   Cache_Control:
|_    Header: Cache-Control: no-cache, private

Nmap done: 1 IP address (1 host up) scanned in 5.78 seconds
```

Issue #04 - Missing Content Security Policy | LOW

Abstract

The absence of a Content Security Policy (CSP) poses a significant vulnerability to our web application's security. CSP serves as a critical defense mechanism against various web-based attacks, including cross-site scripting (XSS) and data injection. This report outlines the impact of the missing CSP and provides strategic recommendations to implement and enforce a robust Content Security Policy across our web environment.

Path

/

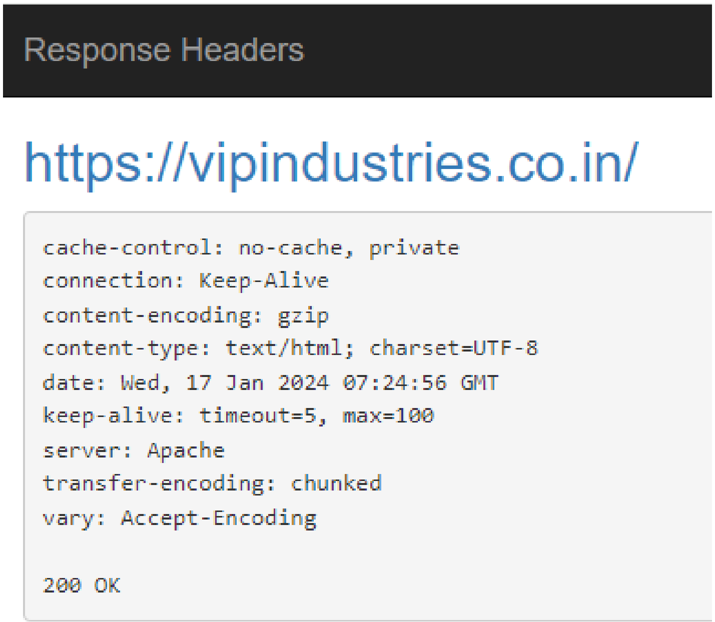
Impact

The vulnerability arising from the absence of a Content Security Policy exposes our web application to heightened risks of XSS attacks, data manipulation, and unauthorized script executions. Without a CSP in place, malicious actors can exploit these vulnerabilities, potentially compromising user data, session integrity, and the overall confidentiality of our system. Furthermore, the lack of content restrictions may lead to the injection of harmful scripts, undermining the trust and reliability of our web services.

Recommendation

To address the identified vulnerability, it is imperative to implement and enforce a comprehensive Content Security Policy (CSP) across our web application. A well-structured CSP can significantly mitigate the risks associated with XSS attacks, providing a robust defense layer against unauthorized script executions and data manipulation. The implementation should include a thorough review of trusted sources for scripts, styles, and other resources, defining and adhering to a policy that restricts the acceptance of content from untrusted origins. Regular audits and updates to the CSP will ensure ongoing protection against emerging threats, reinforcing the overall security posture of our web infrastructure.

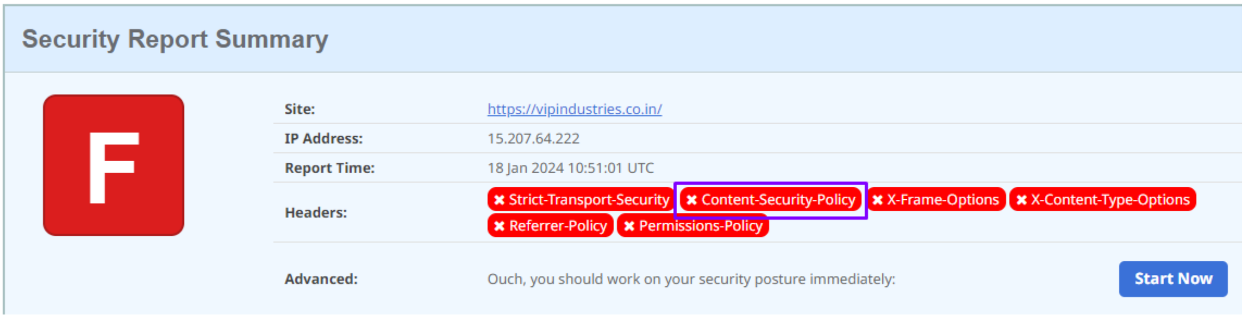
Proof of Concept:



Response Headers

https://vipindustries.co.in/

```
cache-control: no-cache, private
connection: Keep-Alive
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Wed, 17 Jan 2024 07:24:56 GMT
keep-alive: timeout=5, max=100
server: Apache
transfer-encoding: chunked
vary: Accept-Encoding

200 OK
```



Security Report Summary

Site:	https://vipindustries.co.in/
IP Address:	15.207.64.222
Report Time:	18 Jan 2024 10:51:01 UTC
Headers:	✖ Strict-Transport-Security ✖ Content-Security-Policy ✖ X-Frame-Options ✖ X-Content-Type-Options ✖ Referrer-Policy ✖ Permissions-Policy
Advanced:	Ouch, you should work on your security posture immediately:

Issue #05 - Cookie without HttpOnly flag set | LOW

Abstract

The following cookie was issued by the application and does not have the HttpOnly flag set: XSRF-TOKEN

Path

/

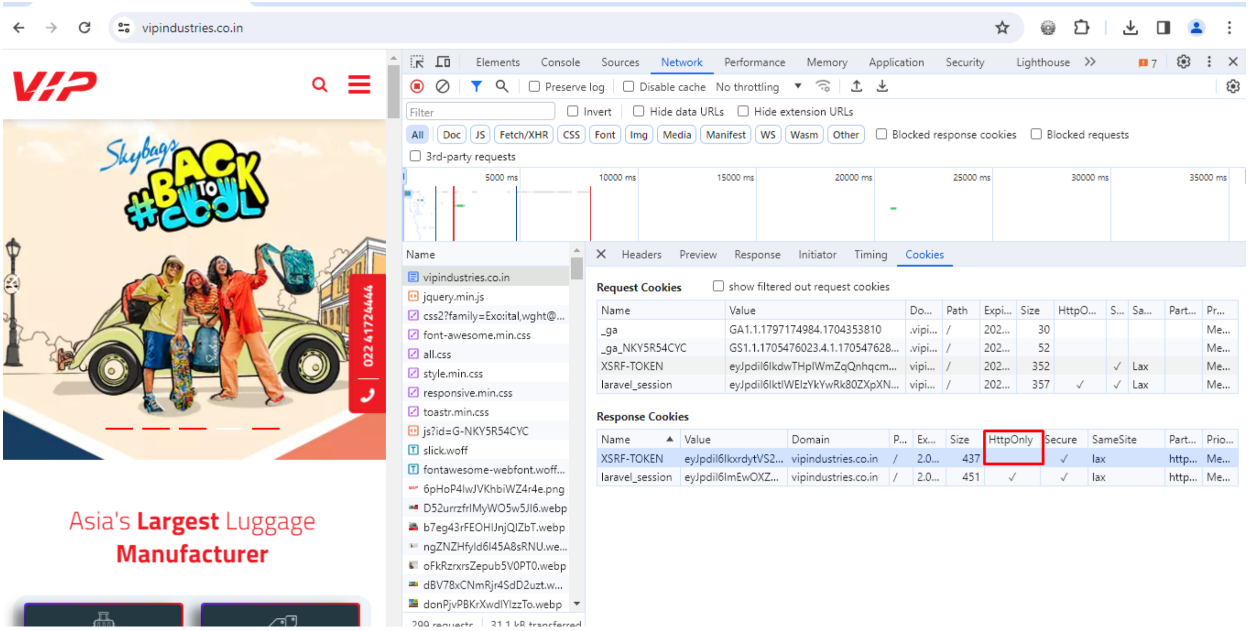
Impact

The cookie appears to contain a session token, which may increase the risk associated with this issue.

Recommendation

To enhance the security posture of our web application, it is strongly recommended to systematically implement the HttpOnly flag for cookies. This precautionary measure serves as a fundamental defense against cross-site scripting (XSS) attacks by restricting client-side script access to cookies. By enforcing the HttpOnly flag, we mitigate the risk of unauthorized access to sensitive user data stored in cookies, thereby fortifying the security of user sessions. It is imperative that we prioritize a comprehensive review and update of our cookie settings, ensuring the consistent application of the HttpOnly flag throughout our web application. This proactive approach is integral to maintaining the integrity and resilience of our security framework.

Proof of Concept:



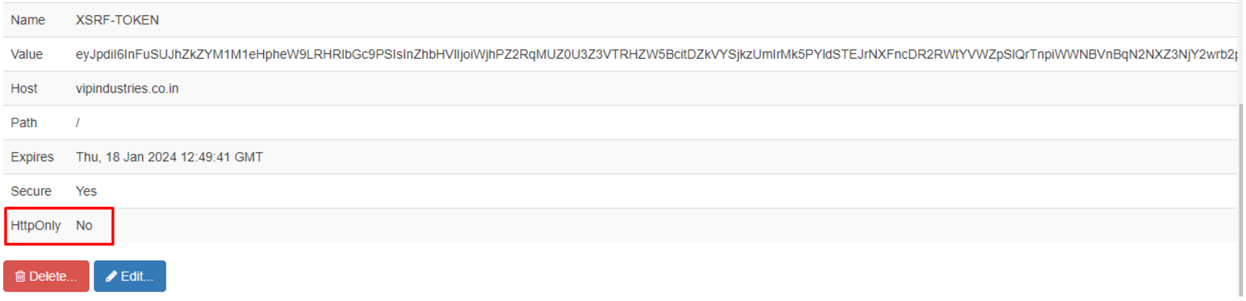
Name
vipindustries.co.in
jquery.min.js
css2?family=Exo:ital,wght@...
font-awesome.min.css
all.css
style.min.css
responsive.min.css
toastr.min.css
js?id=G-NKY5R54CYC
slick.woff
fontawesome-webfont.woff...
6pHoP4lwJVKhbiWZ4r4e.png
D52urrzfrlMyWO5w5JI6.webp
b7eg43rFEOHIJnjQIZbT.webp
ngZNZHfyld6I45A8sRNU.we...
oFkRzrxrsZepub5V0PT0.webp
dBV78xCNmRjr4SdD2uzt.w...
donPjvPBKrXwdIYIzzTo.webp

Request Cookies □ show filtered out request cookies

Name	Value	Do...	Path	Expi...	Size	HttpO...	S...	Sa...	Part...	Pr...
_ga	GA1.1.1797174984.1704353810	.vipi...	/	202...	30					Me...
_ga_NKY5R54CYC	GS1.1.1705476023.4.1.170547628...	.vipi...	/	202...	52					Me...
XSRF-TOKEN	eyJpdiI6IkdwTHpIWmZqQnhqcm...	vipi...	/	202...	352		✓	Lax		Me...
laravel_session	eyJpdiI6IktIWElzYkYwRk80ZXpXN...	vipi...	/	202...	357	✓	✓	Lax		Me...

Response Cookies

Name ▲	Value	Domain	P...	Ex...	Size	HttpOnly	Secure	SameSite	Part...	Prio...
XSRF-TOKEN	eyJpdiI6IkxrdytVS2...	vipindustries.co.in	/	2.0...	437		✓	lax	http...	Me...
laravel_session	eyJpdiI6ImEwOXZ...	vipindustries.co.in	/	2.0...	451	✓	✓	lax	http...	Me...



Name	XSRF-TOKEN
Value	eyJpdiI6InFuSUJhZkZYM1M1eHpheW9LRHRlbGc9PSIsInZhbHVlIjoiWjhPZ2RqMUZ0U3Z3VTRHZW5BcitDZkVYSjkzUmlrMk5PYldSTEJrNXFncDR2RWtYVWZpSlQrTnpiWWNBVnBqN2NXZ3NjY2wrb2
Host	vipindustries.co.in
Path	/
Expires	Thu, 18 Jan 2024 12:49:41 GMT
Secure	Yes
HttpOnly	No

Issue #06 - Missing X-Frame-Options Header | LOW

Abstract

The X-Frame-Options HTTP header field indicates a policy that specifies whether the browser should render the transmitted resource within a frame or an iframe. Servers can declare this policy in the header of their HTTP responses to prevent clickjacking attacks, which ensures that their content is not embedded into other pages or frames.

Path

/

Impact

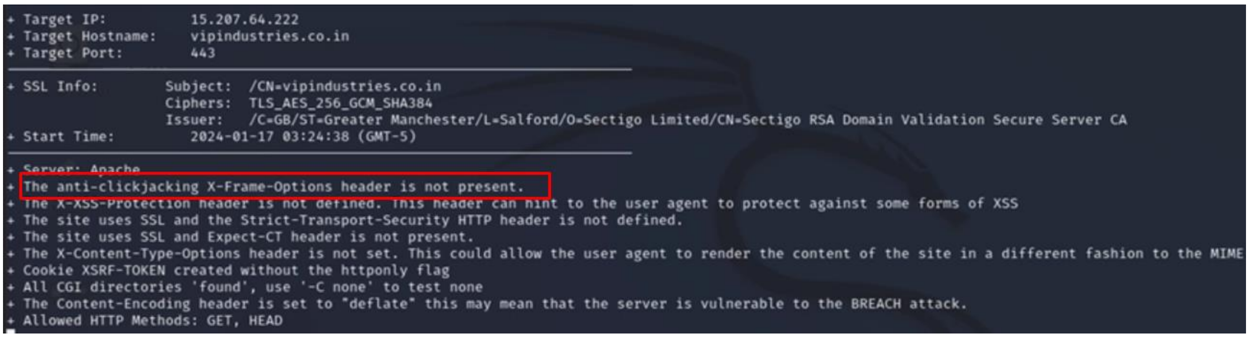
Clickjacking is when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on a framed page when they were intending to click on the top level page. Thus, the attacker is "hijacking" clicks meant for their page and routing them to other another page, most likely owned by another application, domain, or both.

Using a similar technique, keystrokes can also be hijacked. With a carefully crafted combination of stylesheets, iframes, and text boxes, a user can be led to believe they are typing in the password to their email or bank account but are instead typing into an invisible frame controlled by the attacker.

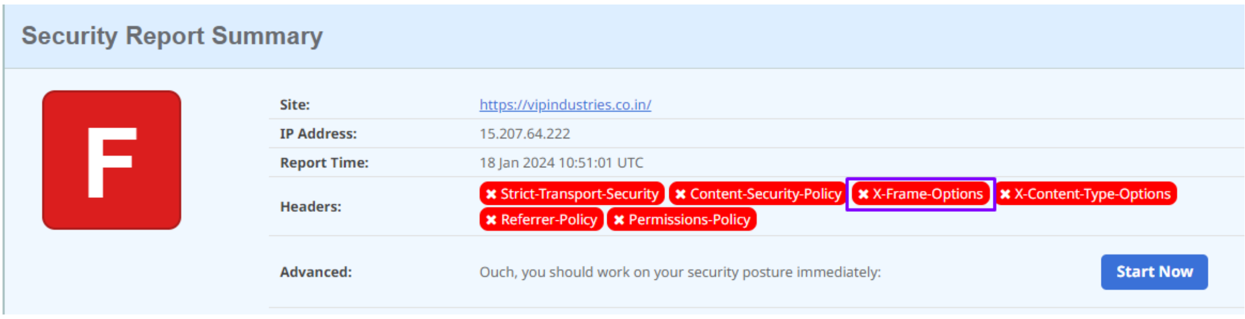
Recommendation

Sending the proper X-Frame-Options in HTTP response headers that instruct the browser to not allow framing from other domains.
X-Frame-Options: DENY It completely denies being loaded in frame/iframe.
X-Frame-Options: SAMEORIGIN It allows only if the site which wants to load has the same origin.
X-Frame-Options: ALLOW-FROM URL It grants a specific URL to load itself in a iframe. However please pay attention to that, not all browsers support this.
Employing defensive code in the UI to ensure that the current frame is the most top-level window.

Proof of Concept:



```
+ Target IP:          15.207.64.222
+ Target Hostname:    vipindustries.co.in
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /CN=vipindustries.co.in
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Domain Validation Secure Server CA
+ Start Time:         2024-01-17 03:24:38 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache
+ 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 forms of XSS
+ The site uses SSL and the Strict-Transport-Security HTTP header is not defined.
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME
+ Cookie XSRF-TOKEN created without the httponly flag
+ All CGI directories 'found', use '-C none' to test none
+ The Content-Encoding header is set to "deflate" this may mean that the server is vulnerable to the BREACH attack.
+ Allowed HTTP Methods: GET, HEAD
```



Security Report Summary

F

Site:	https://vipindustries.co.in/
IP Address:	15.207.64.222
Report Time:	18 Jan 2024 10:51:01 UTC
Headers:	✖ Strict-Transport-Security ✖ Content-Security-Policy ✖ X-Frame-Options ✖ X-Content-Type-Options ✖ Referrer-Policy ✖ Permissions-Policy
Advanced:	Ouch, you should work on your security posture immediately:

Issue #07 - Referrer-Policy Not Implemented | LOW

Abstract

We observed that no Referrer-Policy header was implemented.

Referrer-Policy is a security header designed to prevent cross-domain Referrer leakage.

Path

/

Impact

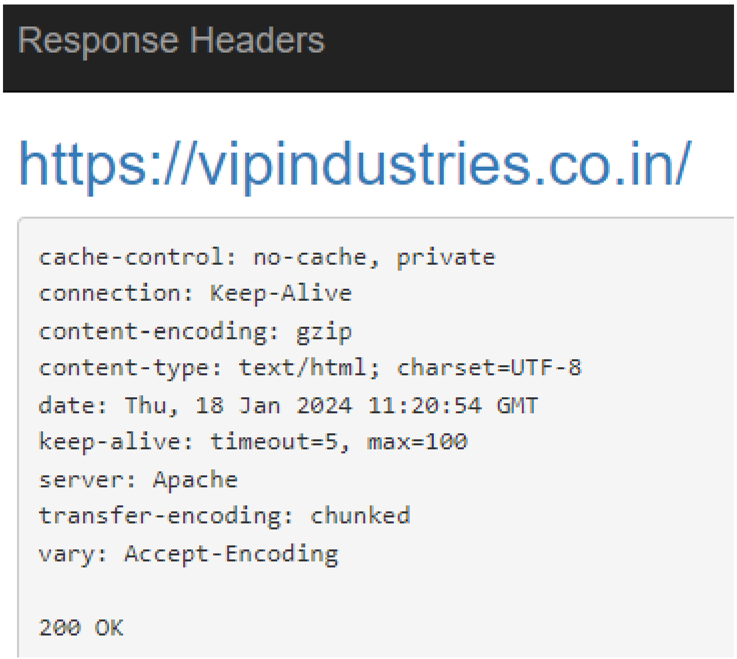
Referrer header is a request header that indicates the site which the traffic originated from. If there is no adequate prevention in place, the URL itself, and even sensitive information contained in the URL will be leaked to the cross-site.

The lack of Referrer-Policy header might affect privacy of the users and sites itself.

Recommendation

Implement a Referrer-Policy by using the Referrer-Policy response header or by declaring it in the meta tags. It's also possible to control referrer information over an HTML-element by using the rel attribute.

Proof of Concept:

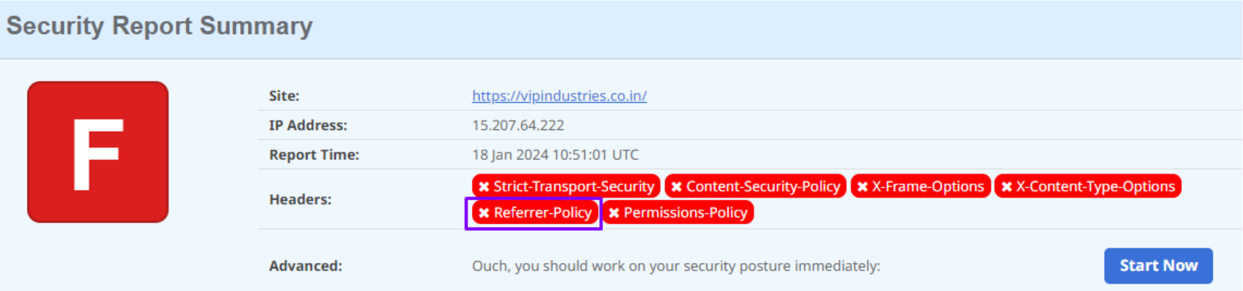


Response Headers

https://vipindustries.co.in/

```
cache-control: no-cache, private
connection: Keep-Alive
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Thu, 18 Jan 2024 11:20:54 GMT
keep-alive: timeout=5, max=100
server: Apache
transfer-encoding: chunked
vary: Accept-Encoding

200 OK
```

Referrer-policy is missing in response header



Security Report Summary

F

Site:	https://vipindustries.co.in/
IP Address:	15.207.64.222
Report Time:	18 Jan 2024 10:51:01 UTC
Headers:	✖ Strict-Transport-Security ✖ Content-Security-Policy ✖ X-Frame-Options ✖ X-Content-Type-Options ✖ Referrer-Policy ✖ Permissions-Policy
Advanced:	Ouch, you should work on your security posture immediately:

Issue #08 - Missing 'X-Content-Type-Options' Header | LOW

Abstract

The HTTP 'X-Content-Type-Options' response header prevents the browser from MIME-sniffing a response away from the declared content-type.

The server did not return a correct 'X-Content-Type-Options' header, which means that this website could be at risk of a Cross-Site Scripting (XSS) attack.

Path

/

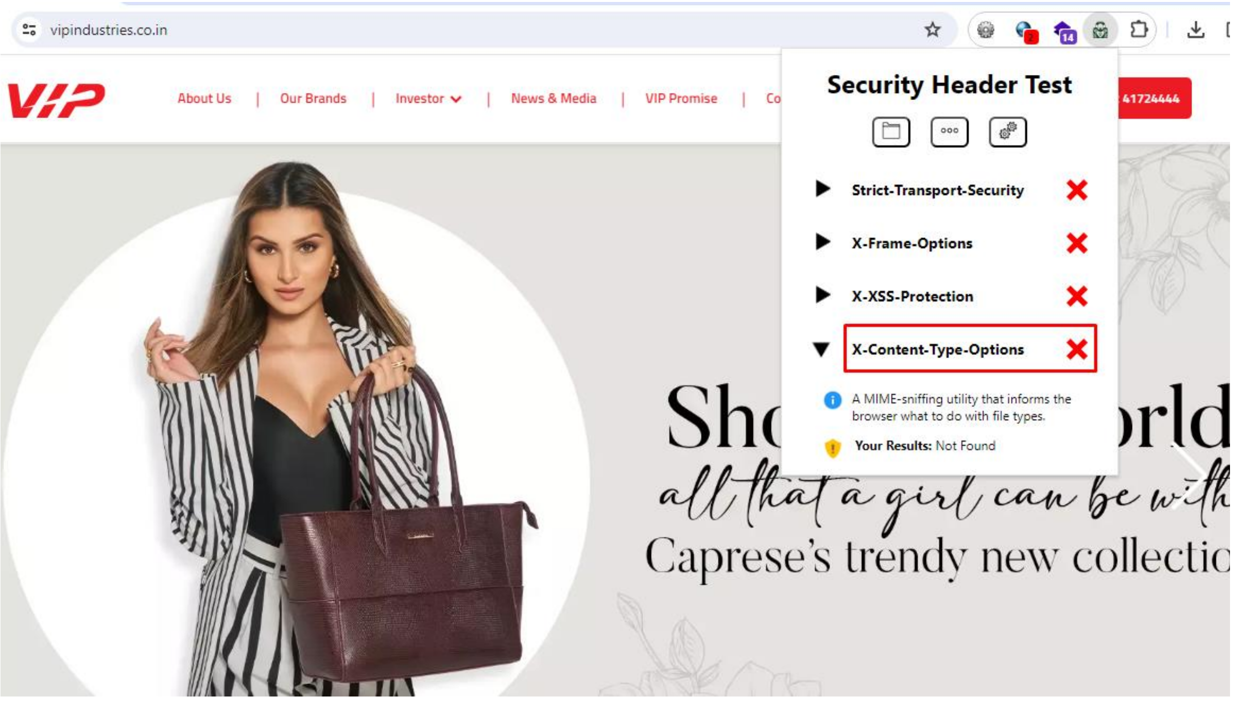
Impact

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 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.
The problem arises once a website allows users to upload content which is then published on the web server. If an attacker can carry out XSS (Cross-site Scripting) attack by manipulating the content in a way to be accepted by the web application and rendered as HTML by the browser, it is possible to inject code in e.g. an image file and make the victim execute it by viewing the image.

Recommendation

Configure your web server to include an 'X-Content-Type-Options' header with a value of 'nosniff'.

Proof of Concept:

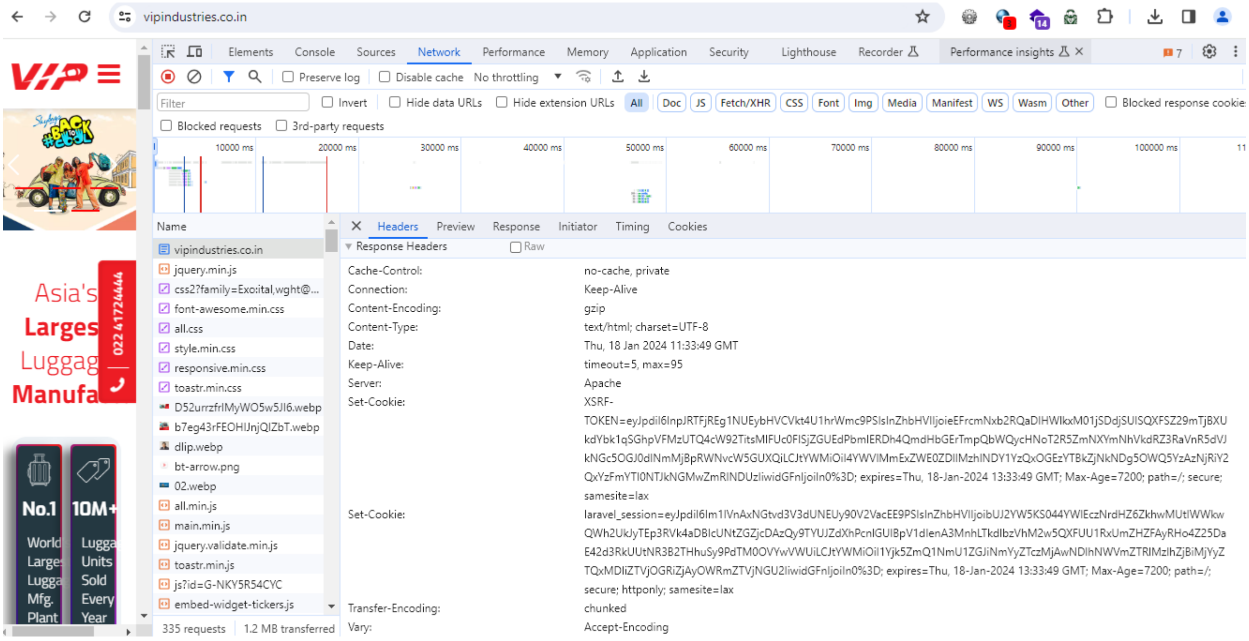

vipindustries.co.in
About Us | Our Brands | Investor | News & Media | VIP Promise | Co
41724444
Sho orld
all that a girl can be with
Caprese's trendy new collectio

Security Header Test

- Strict-Transport-Security ✗
- X-Frame-Options ✗
- X-XSS-Protection ✗
- X-Content-Type-Options ✗

A MIME-sniffing utility that informs the browser what to do with file types.

Your Results: Not Found



Name

- vipindustries.co.in
- jquery.min.js
- css2?family=Exo:ital,wght@...
- font-awesome.min.css
- all.css
- style.min.css
- responsive.min.css
- toastr.min.css
- D52urrzfrlMyWO5w5JI6.webp
- b7eg43rFEOHIJnjQIZbT.webp
- dlip.webp
- bt-arrow.png
- 02.webp
- all.min.js
- main.min.js
- jquery.validate.min.js
- toastr.min.js
- js?id=G-NKY5R54CYC
- embed-widget-tickers.js

335 requests | 1.2 MB transferred

```
Cache-Control:        no-cache, private
Connection:           Keep-Alive
Content-Encoding:     gzip
Content-Type:         text/html; charset=UTF-8
Date:                 Thu, 18 Jan 2024 11:33:49 GMT
Keep-Alive:           timeout=5, max=95
Server:               Apache
Set-Cookie:           XSRF-
                      TOKEN=eyJpdil6InpJRTFjREg1NUEybHVCVkt4U1hrWmc9PSIsInZhbHVlIjoieEFrcmNxb2RQaDlHWlkxM01jSDdjSUlSQXFSZ29mTjBXU
                      kdYbk1qSGhpVFMzUTQ4cW92TitsMlFUc0FISjZGUEdPbmlERDh4QmdHbGErTmpQbWQycHNoT2R5ZmNXYmNhVkdRZ3RaVnR5dVJ
                      kNGc5OGJ0dlNmMjBpRWNvcW5GUXQiLCJtYWMiOiI4YWVlMmExZWE0ZDllMzhlNDY1YzQxOGEzYTBkZjNkNDg5OWQ5YzAzNjRiY2
                      QxYzFmYTI0NTJkNGMwZmRlNDUzliwidGFnIjoiIn0%3D; expires=Thu, 18-Jan-2024 13:33:49 GMT; Max-Age=7200; path=/; secure;
                      samesite=lax
Set-Cookie:           laravel_session=eyJpdil6Im1lVnAxNGtvd3V3dUNEUy90V2VacEE9PSIsInZhbHVlIjoibUJ2YW5KS044YWlEczNrdHZ6ZkhwMUtlWWkw
                      QWh2UkJyTEp3RVk4aDBlcUNtZGZjcDAzQy9TYUJZdXhPcnlGUlBpV1dlenA3MnhLTkdlbzVhM2w5QXFUU1RxUmZHZFAyRHo4Z25Da
                      E42d3RkUUtNR3B2THhuSy9PdTM0OVYwVWUiLCJtYWMiOiI1Yjk5ZmQ1NmU1ZGJiNmYyZTczMjAwNDlhNWVmZTRlMzlhZjBiMjYyZ
                      TQxMDliZTVjOGRiZjAyOWRmZTVjNGU2liwidGFnIjoiIn0%3D; expires=Thu, 18-Jan-2024 13:33:49 GMT; Max-Age=7200; path=/;
                      secure; httponly; samesite=lax
Transfer-Encoding:    chunked
Vary:                 Accept-Encoding
```

Issue #09 - Web browser XSS protection not enabled | LOW

Abstract

Web Browser XSS Protection is not enabled or is disabled by the configuration of the 'X-XSS-Protection' HTTP response header on the web server. Hackers use XSS attacks to trick trusted websites into delivering malicious content.

Path

/

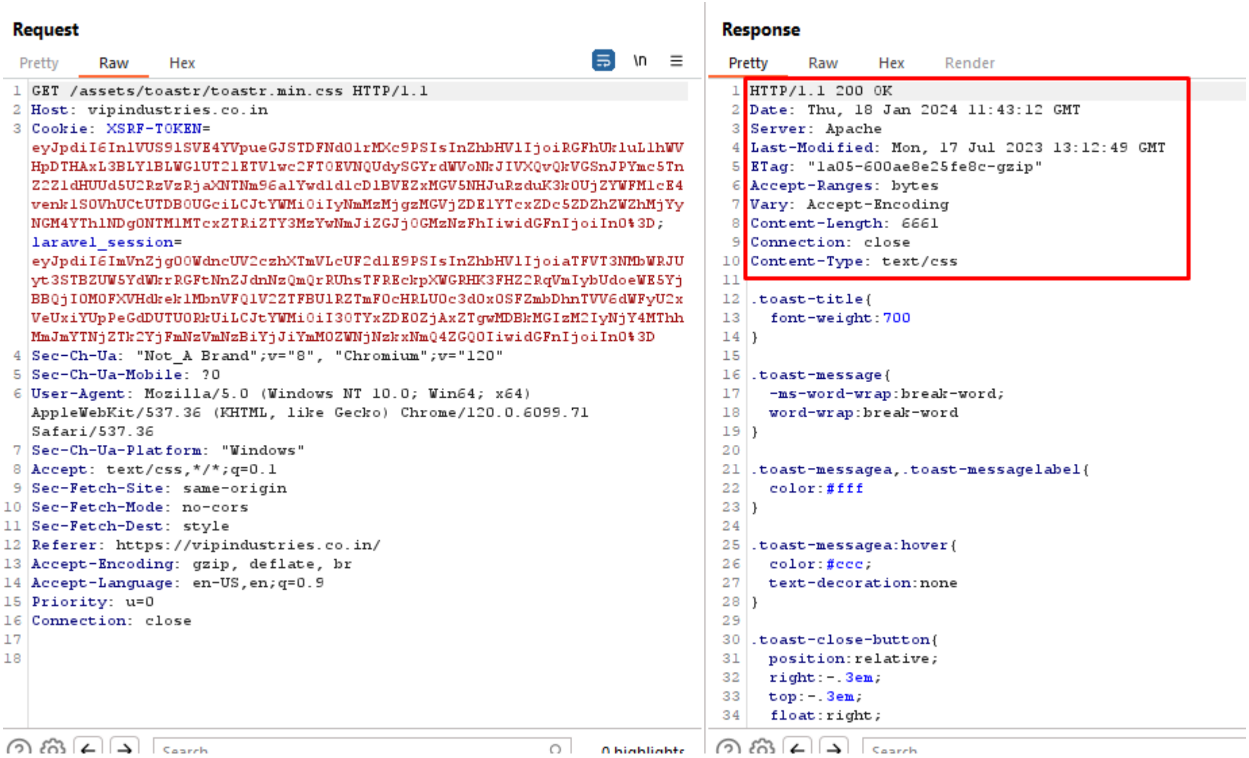
Impact

Some browsers, including Internet Explorer, contain built-in filters designed to protect against cross-site scripting (XSS) attacks. Applications can instruct browsers to disable this filter by setting the following response header: • X-XSS-Protection: 0 This behavior does not in itself constitute a vulnerability; in some cases, XSS filters may themselves be leveraged to perform attacks against application users. However, in typical situations XSS filters do provide basic protection for application users against some XSS vulnerabilities in applications. The presence of this header should be reviewed to establish whether it affects the application's security posture.

Recommendation

Review whether the application needs to disable XSS filters. In most cases you can gain the protection provided by XSS filters without the associated risks by using the following response header: • X-XSS-Protection: 1; mode=block When this header is set, browsers that detect an XSS attack will simply render a blank page instead of attempting to sanitize the injected script. This behaviour is considerably less likely to introduce new security issues.

Proof of Concept:



```
GET /assets/toastr/toastr.min.css HTTP/1.1
Host: vipindustries.co.in
Cookie: XSRF-TOKEN=
eyJpdiI6InlVUS91SVE4YVpueGJSTDFNd01rMXc9PSIsInZhbHVlIjoiRGFhUkluL1hWV
HpDTHAxL3BLY1BLWG1UT21ETV1wc2FTOEVNQUdySGYrdWVoNkJIVXQvQkVGSnJPYmc5Tn
Z2Z1dHUUd5U2RzVzRjaXNTNm96alYwdldlcD1BVEZxMGV5NHJuRzduK3k0UjZYWFM1cE4
venk1S0VhUCtUTDB0UGciLCJtYWMiOiIyNmMzMjgzMGVjZDE1YTcxZDc5ZDZhZWZhMjYy
NGM4YThlNDg0NTM1MTcxZTRiZTY3MzYwNmJiZGJjOGMzNzFhIiwidGFnIjoiIn0%3D;
laravel_session=
eyJpdiI6ImVnZjg0OWdncUV2czhXTmVLcUF2d1E9PSIsInZhbHVlIjoiaTFVT3NMbWRJU
yt3STBZUW5YdWkrRGFtNnZJdnNzQmQrRUhsTFREckpXWGRHK3FHZ2RqVmIybUdoeWE5Yj
BBQjIOMOFXVHdkek1MbnVFQ1V2ZTFBU1RZTmF0cHRLU0c3d0x0SFZmbDhnTVV6dWFyU2x
VeUxiYUpPeGdDUTU0RkUiLCJtYWMiOiI3OTYxZDE0ZjAxZTgwMDBkMGIzM2IyNjY4MThh
MmJmYTNjZTk2YjFmNzVmNzBiYjJiYmM0ZWNjNzkxNmQ4ZGQ0IiwidGFnIjoiIn0%3D
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: text/css,*/*;q=0.1
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: style
Referer: https://vipindustries.co.in/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0
Connection: close


```

```
HTTP/1.1 200 OK
Date: Thu, 18 Jan 2024 11:43:12 GMT
Server: Apache
Last-Modified: Mon, 17 Jul 2023 13:12:49 GMT
ETag: "1a05-600ae8e25fe8c-gzip"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 6661
Connection: close
Content-Type: text/css

.toast-title{
  font-weight:700
}

.toast-message{
  -ms-word-wrap:break-word;
  word-wrap:break-word
}

.toast-messagea,.toast-messagelabel{
  color:#fff
}

.toast-messagea:hover{
  color:#ccc;
  text-decoration:none
}

.toast-close-button{
  position:relative;
  right:-.3em;
  top:-.3em;
  float:right;
```

X-XSS protection header is missing

Issue #10 - Content type incorrectly stated | LOW

Abstract

The response states that the content type is font/woff. However, it actually appears to contain a WOFF font. If the URL path can be manipulated to end with ".html", the following browsers may interpret the response as HTML:
Internet Explorer 11
Internet Explorer 11 (Compatibility Mode)

Path

/assets/css/fonts/slick.woff

Impact

If a response specifies an incorrect content type then browsers may process the response in unexpected ways. If the content type is specified to be a renderable text-based format, then the browser will usually attempt to interpret the response as being in that format, regardless of the actual contents of the response. Additionally, some other specified content types might sometimes be interpreted as HTML due to quirks in particular browsers. This behavior might lead to otherwise "safe" content such as images being rendered as HTML, enabling cross-site scripting attacks in certain conditions.

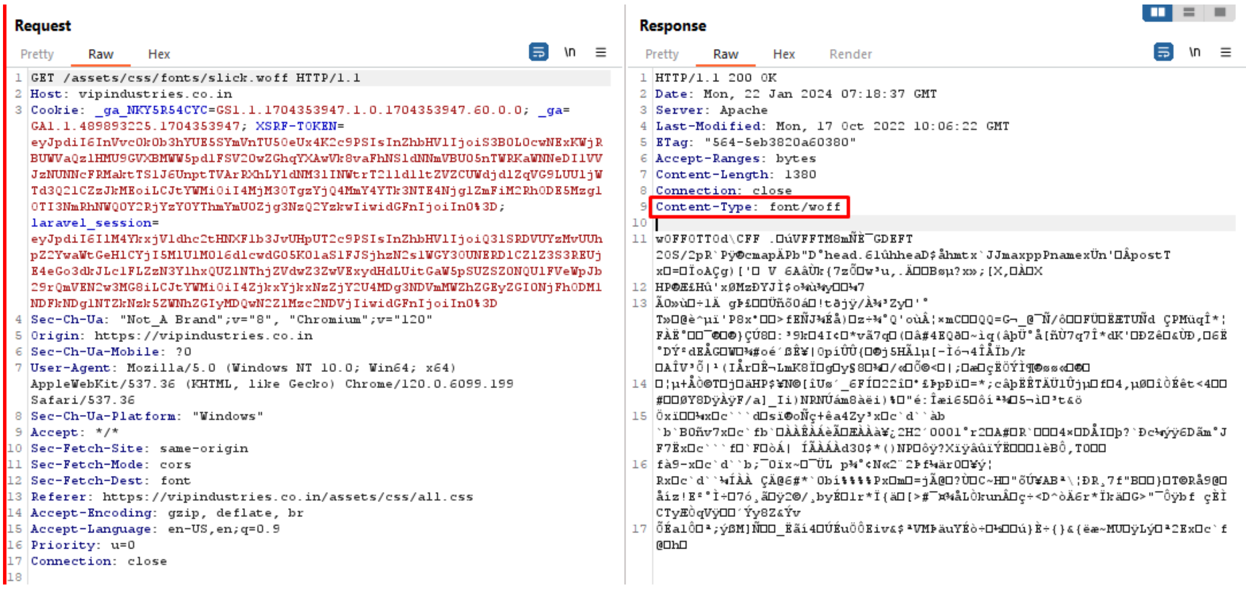
The presence of an incorrect content type statement typically only constitutes a security flaw when the affected resource is dynamically generated, uploaded by a user, or otherwise contains user input. You should review the contents of affected responses, and the context in which they appear, to determine whether any vulnerability exists.

Recommendation

For every response containing a message body, the application should include a single Content-type header that correctly and unambiguously states the MIME type of the content in the response body.

Additionally, the response header "X-content-type-options: nosniff" should be returned in all responses to reduce the likelihood that browsers will interpret content in a way that disregards the Content-type header.

Proof of Concept:



```
GET /assets/css/fonts/slick.woff HTTP/1.1
Host: vipindustries.co.in
Cookie: _ga_NKY5R54CYC=GS1.1.1704353947.1.0.1704353947.60.0.0; _ga=
GA1.1.489893225.1704353947; XSRF-TOKEN=
eyJpdiI6InVvc0k0b3hYUE5SYmVnTU50eUx4K2c9PSIsInZhbHVlIjoiS3B0L0cwNExKWjR
BUWVaQzlHMU9GVXBMWW5pdlFSV20wZGhqYXAwVk8vaFhNS1dNNmVBU05nTWRKaWNNeDI1VV
JzNUNNcFRMaktTS1J6UnptTVArRXhLY1dNM3lINWtrT211dl1tZVZCUWdjdlZqVG9LUUljW
Td3Q21CZzJkMEoiLCJtYWMiOiI4MjM3OTgzYjQ4MmY4YTk3NTE4NjglZmFiM2Rh0DE5Mzgl
OTI3NmRhNWQ0Y2RjYzY0YThmYmU0Zjg3NzQ2YzkwIiwidGFnIjoiIn0%3D;
laravel_session=
eyJpdiI6IlM4YkxjVldhc2tHNXFlb3JvUHpUT2c9PSIsInZhbHVlIjoiQ3lSRDVUYzMvUUh
pZ2YwaWtGeHlCYjI5MlU1MO16dlcwdG05K0laSlFJSjhzN2slWGY30UNERDlCZlZ3S3REUj
E4eGo3dkJLclFLZzN3YlhxQUZlNThjZVdwZ3ZwVExydHdLUitGaW5pSUZSZ0NQUlFVeWpJb
29rQmVEN2w3MG8iLCJtYWMiOiI4ZjkxYjkxNzZjY2U4MDg3NDVmMWZhZGEyZGI0NjFh0DM1
NDFkNDglNTZkNzk5ZWNhZGIyMDQwN2ZlMzc2NDVjIiwidGFnIjoiIn0%3D
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Origin: https://vipindustries.co.in
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199
Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: font
Referer: https://vipindustries.co.in/assets/css/all.css
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=0
Connection: close

```

```
HTTP/1.1 200 OK
Date: Mon, 22 Jan 2024 07:18:37 GMT
Server: Apache
Last-Modified: Mon, 17 Oct 2022 10:06:22 GMT
ETag: "564-5eb3820a60380"
Accept-Ranges: bytes
Content-Length: 1380
Connection: close
Content-Type: font/woff

wOFFOTTOd\CFF .□úVFFTM8mÑÈ¯GDEFT
2OS/2pR`Pÿ®cmapÄPb"D°head.61ûhheaD$åhmtx`JJmaxppPnamexÜn'□Âpost T
x□=□ÏoAÇg)['□ V 6AâÙk{7zÕ□w³u,.Ä□□Bøµ?x»;[X,□À□X
HP®Æ£Hû'x0MzÐYJÌ$o¾ù¾y□□¾7
Ã0»ù□÷1Ä gÞ£□□Üñõ0á□!tðjÿ/À¾³Zy□'°
T»□@è^µï'P8x*□□>fEÑJ¾Éå)□z÷¾°Q'oùÂ¦×mC□□QQ=G¬_@¯Ñ/ô□□FÜ□ËÆTUÑd ÇPMüqÎ*¦
FÀË°□□¯®□®}ÇÚ8□:³9k□4I¢□*vã7q□(□â#4EQð□~ìq(âþÜ°å[ñÙ7q7Î*dK'□ÐZê□&ÙÐ,□6Ë
°DÝ²dEÅG□W□¾#oé´ßÊ¥|0píÛÛ{□®j5HÃlµ[-Ìó¬4ÎÅÏb/k
□AÎV³Õ|¹(IÅr□Ê¬LmK8Ï□g□y$8□¾□/«□Õ©<□|;□æ□çËÖÝÌ¶®ss«□®□
□¦µ+ÅÒ®T□j□äHP$¥N®[îUs´_6FÍ□22î□*£ÞpÐi□=*;câþËÊTÄÜ1Ûjµ□f□4,µ0□îÒÉêt<4□□
#□□0Y8DÿÀÿF/a]_Ii)NRNÚám8àëi)%□"é:Îæi65□ôí²¾□5¬ì□³t&ö
Öxï□□¼x□c```d□sï®oÑç+êa4Zy³x□c`d``àb
`b`B0ñv7x□c`fb`□ÀÀÊÀÁèÃ□ÆÀÀà¥¿2H2´0001°r2□A#□R`□□□4×□DÅI□þ?`Ðc¼ýÿ6Dãm°J
F7Ëx□c```f□`F□òÁ| ÍÃÀÀÀd30$*()NP□ôÿ?XïÿâûïÝË□□□lèBÔ,T0□□
fà9-x□c`d``b;¯0ïx~□¯ÜL p¾°¢N«2¨2Þf¼ärO□¥ý¦
Rx□c`d``¼ÍÀÀ ÇÄ@6#*`0bí%%%%Px□m□=jÃ@□?Ù□C~H□"õÚ¥AB²\¦ÐR¸7f"B□□}□T®Rå9@□
åíz!E²°Ì÷□7ó¸ã□ÿ2®/¸byÉ□1r*Ï{ä□[>#¯¤¾åLÒkunÂ□ç÷<D^òÄ6r*Ïkä□G>"¯Ôÿbf çËÌ
CTyÆÒqVÿ□□´Ýy8Z&Ýv
ÕÉalÔ□ª;ýßM]Ñ□□_Ëãí4□ÚÉuÖÔEiv&$ªVMÞäuYÉò÷□¼□□ú}È÷{}&{ëæ~MU□ÿLý□ª2Ex□c`f
@□h□
```