netsparker®
web application security scanner

NETSPARKER SCAN REPORT SUMMARY

TARGET URL	http://18.209.2.175/
SCAN DATE	06/06/2021 14:05:04 (UTC+01:00)
REPORT DATE	06/06/2021 14:19:32 (UTC+01:00)
SCAN DURATION	00:04:08
NETSPARKER VERSION	4.9.5.17367-4.9.5-hf1-7f4b384

Total Requests
2041

Average Speed
8.23 req/sec.

8
Identified

1
Confirmed

0
Critical

5
Informational

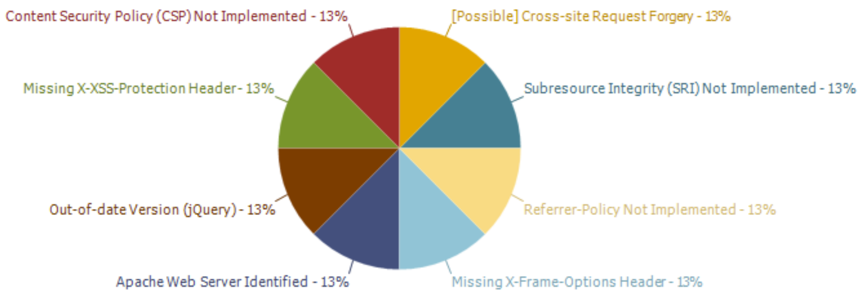
SCAN SETTINGS

ENABLED ENGINES	SQL Injection, SQL Injection (Boolean), SQL Injection (Blind), Cross-site Scripting, Command Injection, Command Injection (Blind), Local File Inclusion, Remote File Inclusion, Code Evaluation, HTTP Header Injection, Open Redirection, Expression Language Injection, Web App Fingerprint, RoR Code Execution, WebDAV, Reflected File Download, Insecure Reflected Content, XML External Entity, File Upload, Windows Short Filename, Cross-Origin Resource Sharing (CORS), HTTP Methods, Server-Side Request Forgery (Pattern Based), Server-Side Request Forgery (DNS), SQL Injection (Out of Band), XML External Entity (Out of Band), Cross-site Scripting (Blind), Remote File Inclusion (Out of Band), Code Evaluation (Out of Band)
URL REWRITE MODE	Heuristic
DETECTED URL REWRITE RULES	None
EXCLUDED URL PATTERNS	(log|sign)\-?(out|off) exit endsession gtm\.js

Authentication

Scheduled

VULNERABILITIES



MEDIUM
13%

LOW
25%

INFORMATION
63%

VULNERABILITY SUMMARY

URL	Parameter	Method	Vulnerability	Confirmed
http://18.209.2.175/		GET	Out-of-date Version (jQuery)	No
		GET	Missing X-Frame-Options Header	No
		GET	Apache Web Server Identified	No
		GET	Missing X-XSS-Protection Header	No
		GET	Subresource Integrity (SRI) Not Implemented	No
		GET	Content Security Policy (CSP) Not Implemented	No
		GET	Referrer-Policy Not Implemented	Yes
http://18.209.2.175/add		GET	[Possible] Cross-site Request Forgery	No

1. Out-of-date Version (jQuery)

1 TOTAL
MEDIUM

Netsparker identified the target web site is using jQuery and detected that it is out of date.

Impact

Since this is an old version of the software, it may be vulnerable to attacks.

Remedy

Please upgrade your installation of jQuery to the latest stable version.

Remedy References

- Downloading jQuery

Known Vulnerabilities in this Version

jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.

External References

- CVE-2015-9251

jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the &#39;&lt;&#39; character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the &#39;&lt;&#39; character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

External References

- CVE-2012-6708

jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing &lt;option&gt; elements from untrusted sources - even after sanitizing it - to one of jQuery&#39;s DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

External References

- CVE-2020-11023

jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery&#39;s DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.

External References

- CVE-2020-11022

jQuery Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jquery prior to 1.9.0 allows Cross-site Scripting attacks via the load method. The load method fails to recognize and remove &quot;&lt;script&gt;&quot; HTML tags that contain a whitespace character, i.e: &quot;&lt;/script &gt;&quot;, which results in the enclosed script logic to be executed.

External References

- CVE-2020-7656

JQuery Prototype Pollution Vulnerability

jQuery before 3.4.0, as used in Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.

External References

- CVE-2019-11358

Classification

OWASP 2013-A9 PCI V3.1-6.2 PCI V3.2-6.2 CAPEC-310

1.1. http://18.209.2.175/

http://18.209.2.175/

Identified Version

1.8.3 (contains 6 medium vulnerabilities)

Latest Version

3.6.0

Vulnerability Database

Result is based on 5/25/2021 vulnerability database content.

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

2. Missing X-Frame-Options Header

1 TOTAL

LOW

Netsparker detected a missing `X-Frame-Options` header which means that this website could be at risk of a clickjacking attack.

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.

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.

Remedy

- 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 to be loaded in frame/iframe.
  - `X-Frame-Options: SAMEORIGIN` It allows only if the site which wants to load has a 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.

External References

- Clickjacking
- Can I Use X-Frame-Options

Remedy References

- Clickjacking Defense Cheat Sheet

Classification

OWASP 2013-A5 CWE-693 CAPEC-103

2.1. http://18.209.2.175/

http://18.209.2.175/

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

3. [Possible] Cross-site Request Forgery

1 TOTAL

LOW

Netsparker identified a possible Cross-Site Request Forgery.

CSRF is a very common vulnerability. It's an attack which forces a user to execute unwanted actions on a web application in which the user is currently authenticated.

Impact

Depending on the application, an attacker can mount any of the actions that can be done by the user such as adding a user, modifying content, deleting data. All the functionality that’s available to the victim can be used by the attacker. Only exception to this rule is a page that requires extra information that only the legitimate user can know (such as user’s password).

Remedy

- Send additional information in each HTTP request that can be used to determine whether the request came from an authorized source. This "validation token" should be hard to guess for attacker who does not already have access to the user's account. If a request is missing a validation token or the token does not match the expected value, the server should reject the request.
- If you are posting form in ajax request, custom HTTP headers can be used to prevent CSRF because the browser prevents sites from sending custom HTTP headers to another site but allows sites to send custom HTTP headers to themselves using XMLHttpRequest.
    - For native XMLHttpRequest (XHR) object in JavaScript;

      ```
      xhr = new XMLHttpRequest();
      xhr.setRequestHeader('custom-header', 'value');
      ```

      For JQuery, if you want to add a custom header (or set of headers) to

      a. **individual request**

      ```
      $.ajax({
          url: 'foo/bar',
          headers: { 'x-my-custom-header': 'some value' }
      });
      ```

      b. **every request**

      ```
      $.ajaxSetup({
          headers: { 'x-my-custom-header': 'some value' }
      });
      OR
      $.ajaxSetup({
          beforeSend: function(xhr) {
              xhr.setRequestHeader('x-my-custom-header', 'some value');
          }
      });
      ```

External References

- OWASP Cross-Site Request Forgery (CSRF)

Remedy References

- OWASP Cross-Site Request Forgery (CSRF) Prevention Cheat Sheet

Classification

OWASP 2013-A8 PCI V3.1-6.5.9 PCI V3.2-6.5.9 CWE-352 CAPEC-62 WASC-9 HIPAA-164.306(A)

3.1. http://18.209.2.175/add

http://18.209.2.175/add

Form Action(s)

▌/add

Certainty

Request

```
GET /add HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Referer: http://18.209.2.175/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:11 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Share Your Thought!</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Share Your Thought!</h1>



<form action="/add" method="post" class="form-horizontal" enctype="multipart/form-data">
<div class="control-group">
<label class="control-label" for="thoughtMessage">Your Thought</label>
<div class="controls">
<textarea rows="3" name="thoughtMessage" id="thoughtMessage" class="input-xxlarge"></textarea>
</div>
</div>
<div class="control-group">
<label class="control-label" for="thoughtAuthor">Your Name</label>
<div class="controls">
<input type="text" name="thoughtAuthor" id="thoughtAuthor" class="input-xxlarge" maxlength="63">
</div>
</div>
<div class="control-group">
<div class="controls">
<button type="submit" class="btn btn-primary"><b class="icon-ok icon-white"></b> Submit Your Thought</button>
<a href="/" class="btn"><b class="icon-chevron-left"></b> Go Back</a>
</div>
</div>
</form>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

4. Apache Web Server Identified

1 TOTAL
INFORMATION

Netsparker identified a web server (Apache) in the target web server's HTTP response.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

External References

- Apache ServerTokens Directive

Classification

OWASP-PC-C7

CVSS 3.0

CVSS Vector String: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:H/RL:O/RC:C
Base: 5.3 (Medium)
Temporal: 5.1 (Medium)
Environmental: 5.1 (Medium)

4.1. http://18.209.2.175/

http://18.209.2.175/

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

5. Missing X-XSS-Protection Header

1 TOTAL
INFORMATION

Netsparker detected a missing `X-XSS-Protection` header which means that this website could be at risk of a Cross-site Scripting (XSS) attacks.

Impact

This issue is reported as additional information only. There is no direct impact arising from this issue.

Remedy

Add the X-XSS-Protection header with a value of "1; mode= block".

- `X-XSS-Protection: 1; mode=block`

External References

- MSDN - Internet Explorer 8 Security Features
- Internet Explorer 8 XSS Filter

Classification

HIPAA-164.308(A) OWASP-PC-C9

5.1. http://18.209.2.175/

http://18.209.2.175/

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

6. Subresource Integrity (SRI) Not Implemented

1 TOTAL
INFORMATION

Subresource Integrity (SRI) provides a mechanism to check integrity of the resource hosted by third parties like Content Delivery Networks (CDNs) and verifies that the fetched resource has been delivered without unexpected manipulation.

SRI does this using hash comparison mechanism. In this way, hash value declared in HTML elements (for now only script and link elements are supported) will be compared with the hash value of the resource hosted by third party.

Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source.

Remedy

Using Subresource Integrity is simply to add *integrity* attribute to the *script* tag along with a base64 encoded cryptographic hash value.

```
<script src="https://code.jquery.com/jquery-2.1.4.min.js" integrity="sha384-R4/ztc4ZlRqWjqIuvf6RX5yb/v90qNGx6fS48N0tRxiGkqveZETq72KgDVJCp2TC" crossorigin="anonymous"></script>
```

The hash algorithm must be one of **sha256**, **sha384** or **sha512**, followed by a '-' character.

External References

- Subresource Integrity
- Do not let your CDN betray you: Use Subresource Integrity
- Web Application Security with Subresource Integrity
- SRI Hash Generator

Classification

6.1. http://18.209.2.175/

http://18.209.2.175/

Identified Sub Resource(s)

- http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

7. Content Security Policy (CSP) Not Implemented

1 TOTAL
INFORMATION

CSP is an added layer of security that helps to mitigate mainly Cross-site Scripting attacks.

CSP can be enabled instructing the browser with a Content-Security-Policy directive in a response header;

```
 Content-Security-Policy: script-src 'self';
```

or in a meta tag;

```
<meta http-equiv="Content-Security-Policy" content="script-src 'self';">
```

In the above example, you can restrict script loading only to the same domain. It will also restrict inline script executions both in the element attributes and the event handlers. There are various directives which you can use by declaring CSP:

- **script-src:** Restricts the script loading resources to the ones you declared. By default, it disables inline script executions unless you permit to the evaluation functions and inline scripts by the unsafe-eval and unsafe-inline keywords.
- **base-uri:** Base element is used to resolve relative URL to absolute one. By using this CSP directive, you can define all possible URLs which could be assigned to base-href attribute of the document.
- **frame-ancestors**: It is very similar to X-Frame-Options HTTP header. It defines the URLs by which the page can be loaded in an iframe.
- **frame-src / child-src**: frame-src is the deprecated version of child-src. Both define the sources that can be loaded by iframe in the page. (Please note that frame-src was brought back in CSP 3)
- **object-src** : Defines the resources that can be loaded by embedding such as Flash files, Java Applets.
- **img-src**: As its name implies, it defines the resources where the images can be loaded from.
- **connect-src**: Defines the whitelisted targets for XMLHttpRequest and WebSocket objects.
- **default-src**: It is a fallback for the directives that mostly ends with -src suffix. When the directives below are not defined, the value set to default-src will be used instead:
  - child-src
  - connect-src
  - font-src
  - img-src
  - manifest-src
  - media-src
  - object-src
  - script-src
  - style-src

When setting the CSP directives, you can also use some CSP keywords:

- **none**: Denies loading resources from anywhere.
- **self** : Points to the document's URL (domain + port).
- **unsafe-inline**: Permits running inline scripts.
- **unsafe-eval**: Permits execution of evaluation functions such as eval().

In addition to CSP keywords, you can also use wildcard or only a scheme when defining whitelist URLs for the points. Wildcard can be used for subdomain and port portions of the URLs:

```
Content-Security-Policy: script-src https://*.example.com;
```

```
Content-Security-Policy: script-src https://example.com:*;
```

```
Content-Security-Policy: script-src https;
```

It is also possible to set a CSP in Report-Only mode instead of forcing it immediately in the migration period. Thus you can see the violations of the CSP policy in the current state of your web site while migrating to CSP:

```
Content-Security-Policy-Report-Only: script-src 'self'; report-uri: https://example.com;
```

Impact

There is no direct impact of not implementing CSP on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. By not implementing CSP you'll be missing out this extra layer of security.

Actions to Take

- Enable CSP on your website by sending the `Content-Security-Policy` in HTTP response headers that instruct the browser to apply the policies you specified.
- Apply the whitelist and policies as strict as possible.
- Rescan your application to see if Netsparker identifies any weaknesses in your policies.

Remedy

Enable CSP on your website by sending the `Content-Security-Policy` in HTTP response headers that instruct the browser to apply the policies you specified.

External References

- An Introduction to Content Security Policy
- Content Security Policy (CSP)

Classification

OWASP-PC-C9

7.1. http://18.209.2.175/

http://18.209.2.175/

Certainty

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```

8. Referrer-Policy Not Implemented

1 TOTAL

INFORMATION

CONFIRMED

1

Netsparker detected that no Referrer-Policy header implemented.

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

Impact

Referer 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 site's itself

Actions to Take

In a response header:

```
Referrer-Policy: no-referrer | same-origin | origin | strict-origin | no-origin-when-downgrading
```

In a META tag

```
<meta name="Referrer-Policy" value="no-referrer | same-origin"/>
```

In an element attribute

```
<a href="http://crosssite.example.com" rel="noreferrer"></a>
```

or

```
<a href="http://crosssite.example.com" referrerpolicy="no-referrer | same-origin | origin | strict-origin | no-origin-when-downgrading"></a>
```

Remedy

Please 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.

External References

- Referrer Policy
- Referrer-Policy - MDN
- A New Security Header: Referrer Policy
- Can I Use Referrer-Policy

Classification

OWASP 2013-A6 CWE-200 OWASP-PC-C9

8.1. http://18.209.2.175/ Confirmed

http://18.209.2.175/

Request

```
GET / HTTP/1.1
Host: 18.209.2.175
Cache-Control: no-cache
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
Accept-Language: en-us,en;q=0.5
X-Scanner: Netsparker
Accept-Encoding: gzip, deflate
```

Response

```
HTTP/1.1 200 OK
Server: Apache
Connection: Keep-Alive
Keep-Alive: timeout=65, max=100
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 06 Jun 2021 12:05:05 GMT
Cache-Control: no-cache

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Your Thoughts</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="assets/css/bootstrap.min.css" rel="stylesheet">
<style>
body {background: url(assets/img/background.png) repeat;}
.hero-unit {background-color: white;}
</style>
<link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
<!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

<body>

<div class="container">

<h1>Your Thoughts</h1>


<p><a href="/add" class="btn"><b class="icon-pencil"></b> Share Your Thought</a></p>

<div class="hero-unit">
<div class="row-fluid">
<blockquote>
<p>test thought</p>
<small>doug</small>
</blockquote>
<hr>
</div>
</div>

</div> <!-- /container -->

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>

</html>
```