PCI SCAN REPORT SUMMARY

TARGET URL	https://s.aomiapp.com/sw/web/index.html
SCAN DATE	2019/6/6 9:13:02 (UTC+08:00)
REPORT DATE	2019/6/6 9:25:46 (UTC+08:00)
SCAN DURATION	00:11:12
NETSPARKER VERSION	5.3.0.23162-master-9c20172

Total Requests
13232

Average Speed
19.66 req/sec.

6
Identified

2
Confirmed

0
Critical

1
Informational

EXPLANATION

EXPLANATION	This report is generated based on PCI classification and it has no validity. **PCI DSS scans must be performed by an approved scanning vendor.** There are 12 more vulnerabilities that are not shown below. Please take a look at the detailed scan report to see them.

6 vulnerabilities listed in PCI classification were found on this web site.

VULNERABILITIES

	ISSUES	INSTANCES	CONFIRMED
CRITICAL	0	0	0
HIGH	0	0	0
MEDIUM	3	3	0
LOW	2	2	2
INFORMATION	1	1	0
BEST PRACTICE	0	0	0
TOTAL	**6**	**6**	**2**

VULNERABILITIES BY PCI

PCI v3.2

6.2 - CRITICAL SECURITY PATCHES

URL	Severity	Vulnerability
https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html	Information	Out-of-date Version (jQuery)
https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html	Medium	Out-of-date Version (Vue.js)

6.5.10 - BROKEN AUTHENTICATION AND SESSION MANAGEMENT

URL	Severity	Vulnerability
https://s.aomiapp.com/sw/web/index.html	Low	Cookie Not Marked as Secure

6.5.4 - INSECURE COMMUNICATIONS

URL	Severity	Vulnerability
https://s.aomiapp.com/sw/web/index.html	Low	Insecure Transportation Security Protocol Supported (TLS 1.0)
https://s.aomiapp.com/sw/web/index.html	Medium	[Possible] Password Transmitted over Query String

6.5.7 - CROSS SITE SCRIPTING (XSS)

URL	Severity	Vulnerability
https://s.aomiapp.com/sw/web/login	Medium	[Possible] Cross-site Scripting

1. Out-of-date Version (Vue.js)

1 TOTAL
MEDIUM

Netsparker identified that the target web site is using Vue.js and detected that it is out of date.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 Vue.js to the latest stable version.

Remedy References

- Downloading Vue.js

Known Vulnerabilities in this Version

vue.js Cross-site Scripting (XSS) Vulnerability

Affected versions of the package are vulnerable to Cross-site Scripting (XSS). Cross-Site Scripting (XSS) attacks occur when an attacker tricks a user's browser to execute malicious JavaScript code in the context of a victim's domain. Such scripts can steal the user's session cookies for the domain, scrape or modify its content, and perform or modify actions on the user's behalf, actions typically blocked by the browser's Same Origin Policy.

vue.js Cross-site Scripting (XSS) Vulnerability

Affected versions of the package are vulnerable to Cross-site Scripting (XSS). Cross-Site Scripting (XSS) attacks occur when an attacker tricks a user's browser to execute malicious JavaScript code in the context of a victim's domain. Such scripts can steal the user's session cookies for the domain, scrape or modify its content, and perform or modify actions on the user's behalf, actions typically blocked by the browser's Same Origin Policy.

Classification

PCI V3.2-6.2

1.1. https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html

https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html

Identified Version

2.2.2

Latest Version

Vdb_LatestVersion_Branch

Vulnerability Database

Vdb_Version_Info

Certainty

Request

```
GET /sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html HTTP/1.1
Host: s.aomiapp.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: sharing-sid=5d4cd1d7-6bc4-48fd-8d6c-721efae5fc41
Referer: https://s.aomiapp.com/sw/web/index.html
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
X-Scanner: Netsparker
```

Response

```
HTTP/1.1 200
Server: nginx
Connection: keep-alive
Content-Encoding:
Content-Language: en-US
Content-Type: text/html;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 06 Jun 2019 01:13:17 GMT
Vary: Accept-Encoding

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>运营登录</title>
<meta name="renderer" content="webkit">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=0">
<link rel="stylesheet" href="/sw/web/layuiadmin/layui/css/layui.css?v=">
<link rel="stylesheet" href="/sw/web/layuiadmin/style/admin.css?v=">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-contextmenu/2.7.0/jquery.contextMenu.min.css?v=">
<link rel="icon" href="">

<script>
//全局异常输出
// window.onerror = function (msg,url,line) {
// var text = url + '\n\n';
// text += (msg) + '\n\n';
// text += ('line:' + line + '\n');
// alert(text)
// return false;
// }
</script>
<style type="text/css">

</style> <link rel="stylesheet" href="/sw/web/layuiadmin/style/login.css" media="all">
<style type="text/css">
h2{
margin: 40px 0 0 0;
}

@media screen and (min-width: 768px) {
.layadmin-user-login-main{
background: #ffffff;
}
}

</style>
</head>
<body>
<div class="layadmin-user-login layadmin-user-display-show" id="vue1">
<div class="layadmin-user-login-main">
<div class="layadmin-user-login-box layadmin-user-login-header">
<h2>代理商服务平台</h2>
<p>———— 共享充电系统 ————</p>
</div>
<form class="layadmin-user-login-box layadmin-user-login-body layui-form">
<div class="layui-form-item">
<label class="layadmin-user-login-icon layui-icon layui-icon-userna
…
```

2. [Possible] Cross-site Scripting

1 TOTAL
MEDIUM

Netsparker detected possible cross-site scripting, which allows an attacker to execute a dynamic script (*JavaScript, VBScript*) in the context of the application.

This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser. Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users' sessions, an attacker might attack an administrator to gain full control over the application.

Although Netsparker believes there is a cross-site scripting in here, it could **not confirm it**. We strongly recommend investigating the issue manually to ensure it is cross-site scripting and needs to be addressed.

Impact

There are many different attacks that can be leveraged through the use of XSS, including:

- Hijacking user's active session.
- Changing the look of the page within the victim's browser.
- Mounting a successful phishing attack.
- Intercepting data and performing man-in-the-middle attacks.

Remedy

This issue occurs because the browser interprets the input as active HTML, JavaScript or VBScript. To avoid this, all input and output from the application should be filtered / encoded. Output should be filtered / encoded according to the output format and location.

There are a number of pre-defined, well structured whitelist libraries available for many different environments. Good examples of these include OWASP Reform and Microsoft Anti-Cross Site Scripting libraries.

External References

- OWASP - Cross-site Scripting
- Cross-site Scripting Web Application Vulnerability
- XSS Shell
- XSS Tunnelling

Remedy References

- [ASP.NET] - Microsoft Anti-XSS Library
- OWASP XSS Prevention Cheat Sheet

Classification

PCI V3.2-6.5.7

CVSS 3.0

CVSS Vector String: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N
Base: 7.4 (High)
Temporal: 7.4 (High)
Environmental: 7.4 (High)

2.1. https://s.aomiapp.com/sw/web/login

https://s.aomiapp.com/sw/web/login

Parameters

Parameter	Type	Value
username	POST	Smith
password	POST	N3tsp@rker-
vercode	POST	3
rememberMe	**POST**	**'"--></style></scRipt><scRipt>netsparker(0x0017AC)</scRipt>**

Notes

Due to the Content-type header of the response, exploitation of this vulnerability might not be possible in all browsers or might not be possible at all. The Content-type header indicates that there is a possibility of exploitation by changing the attack. However Netsparker does not support confirming these issues. You need to manually confirm this problem. Generally lack of filtering in the response can cause Cross-site Scripting vulnerabilities in browsers with auto mime sniffing such as Internet Explorer.

Certainty

Request

```
POST /sw/web/login HTTP/1.1
Host: s.aomiapp.com
Accept: application/json, text/javascript, */*; q=0.01
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Content-Length: 118
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Cookie: sharing-sid=5d4cd1d7-6bc4-48fd-8d6c-721efae5fc41
Origin: https://s.aomiapp.com
Referer: https://s.aomiapp.com/sw/web/login.html?url=https:%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html&username=&password=&vercode=&rememberMe=true
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
X-Requested-With: XMLHttpRequest
X-Scanner: Netsparker

username=Smith&password=N3tsp%40rker-&vercode=3&rememberMe='"--></style></scRipt><scRipt>netsparker(0x0017AC)</scRipt>
```

Response

```
HTTP/1.1 470
Server: nginx
Connection: keep-alive
Content-Language: en-US
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Thu, 06 Jun 2019 01:21:14 GMT
Cache-Control: no-store

{"msg":"[rememberMe]Failed to convert property value of type 'java.lang.String' to required type 'boolean' for property 'rememberMe'; nested exception is
java.lang.IllegalArgumentException: Invalid boolean value ['\"--></style></scRipt><scRipt>netsparker(0x0017AC)</scRipt>]","code":470,"time":1559784074893,"data":null,"type":0}
```

3. [Possible] Password Transmitted over Query String

Netsparker detected that your web application is transmitting passwords over query string.

6 TOTAL

MEDIUM

Impact

A password is sensitive data and shouldn't be transmitted over query string. There are several information-leakage scenarios:

- If your website has external links or even external resources (such as image, javascript, etc), then your query string would be leaked.
- Query string is generally stored in server logs.
- Browsers will cache the query string.

Remedy

Do not send any sensitive data through query string.

Classification

PCI V3.2-6.5.4

CVSS 3.0

CVSS Vector String: CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N
Base: 6.5 (Medium)
Temporal: 6.5 (Medium)
Environmental: 6.5 (Medium)

3.1. https://s.aomiapp.com/sw/web/index.html

https://s.aomiapp.com/sw/web/index.html

Notes

PasswordMayNotBeTransmittedInQueryString

Input Name

password

Certainty

Request

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

Response

```
…
</div>
<div class="layui-form-item">
<label class="layadmin-user-login-icon layui-icon layui-icon-password" for="LAY-user-login-password"></label>
<input type="password" name="password" placeholder="密码" lay-verify="required" class="layui-input">
</div>
<div class="layui-form-item">
<div class="layui-row">
<div class="layui-col-xs7">
<label class="layadmi
…
```

4. Cookie Not Marked as Secure

1 TOTAL
LOW
CONFIRMED
1

Netsparker identified a cookie not marked as secure, and transmitted over HTTPS.

This means the cookie could potentially be stolen by an attacker who can successfully intercept and decrypt the traffic, or following a successful man-in-the-middle attack.

Impact

This cookie will be transmitted over a HTTP connection, therefore if this cookie is important (*such as a session cookie*), an attacker might intercept it and hijack a victim's session. If the attacker can carry out a man-in-the-middle attack, he/she can force the victim to make an HTTP request to steal the cookie.

Actions to Take

1. See the remedy for solution.
2. Mark all cookies used within the application as secure. *(If the cookie is not related to authentication or does not carry any personal information, you do not have to mark it as secure.)*

Remedy

Mark all cookies used within the application as secure.

Required Skills for Successful Exploitation

To exploit this issue, the attacker needs to be able to intercept traffic. This generally requires local access to the web server or to the victim's network. Attackers need to be understand layer 2, have physical access to systems either as waypoints for the traffic, or have locally gained access to to a system between the victim and the web server.

External References

- Netsparker - Security Cookies - Secure Flag
- .NET Cookie.Secure Property
- How to Create Totally Secure Cookies

Classification

PCI V3.2-6.5.10

CVSS 3.0

CVSS Vector String: CVSS:3.0/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Base: 2.0 (Low)
Temporal: 2.0 (Low)
Environmental: 2.0 (Low)

4.1. https://s.aomiapp.com/sw/web/index.html Confirmed

https://s.aomiapp.com/sw/web/index.html

Identified Cookie(s)

`sharing-sid`

Cookie Source

`CustomField_CookieSourceHeader`

Request

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

Response

```
HTTP/1.1 302
Set-Cookie: sharing-sid=5d4cd1d7-6bc4-48fd-8d6c-721efae5fc41; Path=/sw; HttpOnly

Server: nginx
Connection: keep-alive
Content-Length: 0
Location: https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html
Date: Thu, 06 Jun 2019 01:13:07 GMT
```

5. Insecure Transportation Security Protocol Supported (TLS 1.0)

1 TOTAL
LOW
CONFIRMED
1

Netsparker detected that insecure transportation security protocol (TLS 1.0) is supported by your web server.

TLS 1.0 has several flaws. An attacker can cause connection failures and they can trigger the use of TLS 1.0 to exploit vulnerabilities like BEAST (Browser Exploit Against SSL/TLS).

Websites using TLS 1.0 are considered non-compliant by PCI since 30 June 2018.

Impact

Attackers can perform man-in-the-middle attacks and observe the encryption traffic between your website and its visitors.

Remedy

Configure your web server to disallow using weak ciphers. You need to restart the web server to enable changes.

- For Apache, adjust the SSLProtocol directive provided by the mod_ssl module. This directive can be set either at the server level or in a virtual host configuration.

  ```
  SSLProtocol +TLSv1.1 +TLSv1.2
  ```

- For Nginx, locate any use of the directive ssl_protocols in the `nginx.conf` file and remove `TLSv1`.

  ```
  ssl_protocols TLSv1.1 TLSv1.2;
  ```

- For Microsoft IIS, you should make some changes on the system registry.
  1. Click on Start and then Run, type `regedt32` or `regedit`, and then click OK.
  2. In Registry Editor, locate the following registry key or create if it does not exist:

     ```
     HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.0\
     ```

  3. Locate a key named `Server` or create if it doesn't exist.
  4. Under the `Server` key, locate a DWORD value named `Enabled` or create if it doesn't exist and set its value to "0".

External References

- How to disable TLS v1.0
- OWASP - Insecure Configuration Management
- OWASP - Insufficient Transport Layer Protection
- How to disable PCT 1.0, SSL 2.0, SSL 3.0, or TLS 1.0 in Internet Information Services
- IIS Crypto is a free tool that gives administrators the ability to enable or disable protocols, ciphers, hashes and key exchange algorithms on Windows Server 2003, 2008 and 2012
- Date Change for Migrating from SSL and Early TLS
- Browser Exploit Against SSL/TLS Attack (BEAST)

Classification

PCI V3.2-6.5.4

5.1. https://s.aomiapp.com/sw/web/index.html Confirmed

https://s.aomiapp.com/sw/web/index.html

Request

```
[NETSPARKER] SSL Connection
```

Response

```
[NETSPARKER] SSL Connection
```

6. Out-of-date Version (jQuery)

1 TOTAL
INFORMATION

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

Classification
PCI V3.2-6.2

6.1. https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html

https://s.aomiapp.com/sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html

Identified Version
1.12.3

Latest Version
Vdb_LatestVersion_Branch

Vulnerability Database
Vdb_Version_Info

Certainty

Request

```
GET /sw/web/login.html?url=https%3A%2F%2Fs.aomiapp.com%2Fsw%2Fweb%2Findex.html HTTP/1.1
Host: s.aomiapp.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: sharing-sid=5d4cd1d7-6bc4-48fd-8d6c-721efae5fc41
Referer: https://s.aomiapp.com/sw/web/index.html
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.99 Safari/537.36
X-Scanner: Netsparker
```

Response

```
…
</div>
<div class="layui-trans layui-form-item layadmin-user-login-other" ></div>
</form>
</div>
</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.3/jquery.min.js?v=1812211501"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.min.js?v=$1812211501"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jqu
…
```