Acunetix by Invicti

Comprehensive Report

High

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Scan Detail

Target	http://10.136.108.179/
Scan Type	Full Scan
Start Time	Nov 17, 2025, 5:58:58 PM GMT
Scan Duration	9 minutes
Requests	7671
Average Response Time	1ms
Maximum Response Time	56127ms
Application Build	v24.6.240626115
Authentication Profile	-

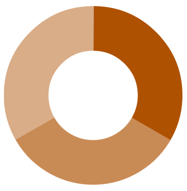
0	3	3	2	3
Critical	High	Medium	Low	Informational

Severity	Vulnerabilities	Instances
Critical	0	0
High	1	3
Medium	3	3
Low	2	2
Informational	3	3
Total	9	11

High Severity

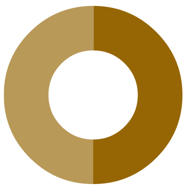
		Instances
🟥	Cross-site Scripting	3

Medium Severity



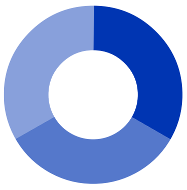
		Instances
🟧	Insecure HTTP Usage	1
🟧	Password transmitted over HTTP	1
🟧	SSL/TLS Not Implemented	1

Low Severity



		Instances
🟫	Cookies Not Marked as HttpOnly	1
🟫	Cookies with missing, inconsistent or contr...	1

Informational



		Instances
🟦	Content Security Policy (CSP) Not Implem...	1
🟦	Error page web server version disclosure	1
🟦	Permissions-Policy header not implemented	1

Impacts

SEVERITY	IMPACT	
High	3	Cross-site Scripting
Medium	1	Insecure HTTP Usage
Medium	1	Password transmitted over HTTP
Medium	1	SSL/TLS Not Implemented
Low	1	Cookies Not Marked as HttpOnly
Low	1	Cookies with missing, inconsistent or contradictory properties
Informational	1	Content Security Policy (CSP) Not Implemented
Informational	1	Error page web server version disclosure
Informational	1	Permissions-Policy header not implemented

Cross-site Scripting

Cross-site Scripting (XSS) refers to client-side code injection attack wherein an attacker can execute malicious scripts into a legitimate website or web application. XSS occurs when a web application makes use of unvalidated or unencoded user input within the output it generates.

Impact

Malicious JavaScript has access to all the same objects as the rest of the web page, including access to cookies and local storage, which are often used to store session tokens. If an attacker can obtain a user's session cookie, they can then impersonate that user.

Furthermore, JavaScript can read and make arbitrary modifications to the contents of a page being displayed to a user. Therefore, XSS in conjunction with some clever social engineering opens up a lot of possibilities for an attacker.

http://10.136.108.179/index.php Verified

URL encoded POST input **username** was set to **RDFYjolf'"()&%<zzz><ScRiPt >hNyX(9923)</ScRiPt>**

Request

```
POST /index.php HTTP/1.1
Referer: http://10.136.108.179/
Cookie: PHPSESSID=iqv6a0tikvtisdekigvl0bucb5
Content-Type: application/x-www-form-urlencoded
Content-Length: 88
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive

password=u]H[ww6KrA9F.x-F&username=RDFYjolf'"()%26%25<zzz><ScRiPt%20>hNyX(9923)</ScRiPt>
```

http://10.136.108.179/register.php Verified

URL encoded POST input **confirm_password** was set to **u]H[ww6KrA9F.x-F'"()&%<zzz><ScRiPt >pr0t(9054)</ScRiPt>**

Request

```
POST /register.php HTTP/1.1
Referer: http://10.136.108.179/
```

```
Cookie: PHPSESSID=iqv6a0tikvtisdekigvl0bucb5
Content-Type: application/x-www-form-urlencoded
Content-Length: 122
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive

confirm_password=u]H[ww6KrA9F.x-F'"()%26%25<zzz><ScRiPt%20>pr0t(9054)
</ScRiPt>&password=u]H[ww6KrA9F.x-F&username=RDFYjolf
```

http://10.136.108.179/register.php Verified

URL encoded POST input **password** was set to **u]H[ww6KrA9F.x-F'"()&%<zzz><ScRiPt >pr0t(9750)</ScRiPt>**

Request

```
POST /register.php HTTP/1.1
Referer: http://10.136.108.179/
Cookie: PHPSESSID=iqv6a0tikvtisdekigvl0bucb5
Content-Type: application/x-www-form-urlencoded
Content-Length: 122
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive

confirm_password=u]H[ww6KrA9F.x-F&password=u]H[ww6KrA9F.x-F'"()%26%25<zzz><ScRiPt%20>pr0t(9750)
</ScRiPt>&username=RDFYjolf
```

Recommendation

Apply context-dependent encoding and/or validation to user input rendered on a page

References

Cross-site Scripting (XSS) Attack - Acunetix
https://www.acunetix.com/websitesecurity/cross-site-scripting/

Types of XSS - Acunetix
https://www.acunetix.com/websitesecurity/xss/

XSS Filter Evasion Cheat Sheet
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

Excess XSS, a comprehensive tutorial on cross-site scripting
https://excess-xss.com/

Cross site scripting
https://en.wikipedia.org/wiki/Cross-site_scripting

Insecure HTTP Usage

It was detected that your web application uses HTTP protocol, but doesn't automatically redirect users to HTTPS.

Impact

In some circumstances, it could be used for a man-in-the-middle (MitM) attack

http://10.136.108.179/

Request

```
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

It's recommended to implement best practices of HTTP Redirection into your web application. Consult web references for more information

References

HTTP Redirections
https://infosec.mozilla.org/guidelines/web_security#http-redirections

Password transmitted over HTTP

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

Impact

A third party may be able to read the user credentials by intercepting an unencrypted HTTP connection.

http://10.136.108.179/

Forms with credentials sent in clear text:

- http://10.136.108.179/

```
Form name: <empty>
Form action: /index.php
Form method: POST
Password input: password
```

- http://10.136.108.179/register.php

```
Form name: <empty>
Form action: /register.php
Form method: POST
Password input: password
```

- http://10.136.108.179/index.php

```
Form name: <empty>
Form action: /index.php
Form method: POST
Password input: password
```

- http://10.136.108.179/reset-password.php

```
Form name: <empty>
Form action: /reset-password.php
Form method: POST
Password input: new_password
```

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

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

SSL/TLS Not Implemented

This scan target was connected to over an unencrypted connection. A potential attacker can intercept and modify data sent and received from this site.

Impact

Possible information disclosure.

http://10.136.108.179/ Verified

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

The site should send and receive data over a secure (HTTPS) connection.

Cookies Not Marked as HttpOnly

One or more cookies don't have the HttpOnly flag set. When a cookie is set with the HttpOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.

Impact

Cookies can be accessed by client-side scripts.

http://10.136.108.179/ Verified

Cookies without HttpOnly flag set:

- http://10.136.108.179/

```
Set-Cookie: PHPSESSID=iqv6a0tikvtisdekigvl0bucb5; path=/
```

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

If possible, you should set the HttpOnly flag for these cookies.

Cookies with missing, inconsistent or contradictory properties

At least one of the following cookies properties causes the cookie to be invalid or incompatible with either a different property of the same cookie, of with the environment the cookie is being used in. Although this is not a vulnerability in itself, it will likely lead to unexpected behavior by the application, which in turn may cause secondary security issues.

Impact

Cookies will not be stored, or submitted, by web browsers.

http://10.136.108.179/ Verified

List of cookies with missing, inconsistent or contradictory properties:

- http://10.136.108.179/

  Cookie was set with:

  ```
  Set-Cookie: PHPSESSID=iqv6a0tikvtisdekigvl0bucb5; path=/
  ```

  This cookie has the following issues:

  ```
  - Cookie without SameSite attribute.
  When cookies lack the SameSite attribute, Web browsers may apply different and
  sometimes unexpected defaults. It is therefore recommended to add a SameSite
  attribute with an appropriate value of either "Strict", "Lax", or "None".
  ```

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

Ensure that the cookies configuration complies with the applicable standards.

References

MDN | Set-Cookie
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie

Securing cookies with cookie prefixes
https://www.sjoerdlangkemper.nl/2017/02/09/cookie-prefixes/

Cookies: HTTP State Management Mechanism
https://tools.ietf.org/html/draft-ietf-httpbis-rfc6265bis-05

SameSite Updates - The Chromium Projects
https://www.chromium.org/updates/same-site

draft-west-first-party-cookies-07: Same-site Cookies
https://tools.ietf.org/html/draft-west-first-party-cookies-07

Content Security Policy (CSP) Not Implemented

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

Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:

```
Content-Security-Policy:
default-src 'self';
script-src 'self' https://code.jquery.com;
```

It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

Impact

CSP can be used to prevent and/or mitigate attacks that involve content/code injection, such as cross-site scripting/XSS attacks, attacks that require embedding a malicious resource, attacks that involve malicious use of iframes, such as clickjacking attacks, and others.

http://10.136.108.179/

Paths without CSP header:

- http://10.136.108.179/
- http://10.136.108.179/register.php
- http://10.136.108.179/config.php
- http://10.136.108.179/welcome.php
- http://10.136.108.179/reset-password.php

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

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

References

Content Security Policy (CSP)
https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP

Implementing Content Security Policy
https://hacks.mozilla.org/2016/02/implementing-content-security-policy/

Error page web server version disclosure

Application errors or warning messages may disclose sensitive information about an application's internal workings to an attacker.

Acunetix found the web server version number and a list of modules enabled on the target server. Consult the 'Attack details' section for more information about the affected page.

Impact

Error messages information about an application's internal workings may be used to escalate attacks.

http://10.136.108.179/

Request

```
GET /eUX3zdDDnj HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

Recommendation

Properly configure the web server not to disclose information about an application's internal workings to the user. Consult the 'Web references' section for more information.

References

Custom Error Responses (Apache HTTP Server)
https://httpd.apache.org/docs/current/custom-error.html

server_tokens (Nginx)
http://nginx.org/en/docs/http/ngx_http_core_module.html#server_tokens

Remove Unwanted HTTP Response Headers (Microsoft IIS)
https://blogs.msdn.microsoft.com/varunm/2013/04/23/remove-unwanted-http-response-headers/

Permissions-Policy header not implemented

The Permissions-Policy header allows developers to selectively enable and disable use of various browser features and APIs.

Impact

http://10.136.108.179/

Locations without Permissions-Policy header:

- http://10.136.108.179/
- http://10.136.108.179/register.php
- http://10.136.108.179/index.php
- http://10.136.108.179/config.php
- http://10.136.108.179/welcome.php
- http://10.136.108.179/reset-password.php

Request

```
GET / HTTP/1.1
Referer: http://10.136.108.179/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/125.0.0.0 Safari/537.36
Host: 10.136.108.179
Connection: Keep-alive
```

References

Permissions-Policy / Feature-Policy (MDN)
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Feature-Policy

Permissions Policy (W3C)
https://www.w3.org/TR/permissions-policy-1/

Coverage

- http://10.136.108.179
  - config.php
  - index.php
    - Inputs
      - POST password, username
  - logout.php
  - register.php
    - Inputs
      - POST confirm_password, password, username
      - GET do
  - reset-password.php
  - welcome.php
    - Inputs
      - POST submit, url