CYBER PRIVILEGE

CYBER PRIVILEGE

PRIVATE LIMITED

CIN : U80300TS2024PTC181676

Vulnerability Assessment & Penetration Testing

For

Web Application

(https://acuverconsulting.com)

Email

hello@cyberprivilege.com

Phone

+91 - 8977308555

Cyber Privilege Private Limited was founded in 2024 with a core mission to deliver top-tier Information Security services.

Statement of Confidentiality

This document contains information that is proprietary and Confidential to "**acuver consulting**" & "**Cyber Privilege Private Limited**", which shall not be disclosed outside to "**acuver consulting**" and "**Cyber Privilege Private Limited**", transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without explicit written permission of "**acuver consulting**" and "**Cyber Privilege Private Limited**" is prohibited.

ALL FINDINGS IN THIS REPORT ARE BASED ON THE INFORMATION SHARED WITH "ACUVER CONSULTING " *TEAM DURING ASSESSMENT DURATION MENTIONED IN THE ASSESSMENT INFORMATION.*
ANY CHANGES IN THE SYSTEMS IN SCOPE POST ASSESSMENT DATES MAY NOT REFLECT IN THIS REPORT AND ***CYBER PRIVILEGE*** *TEAM IS NOT RESPONSIBLE FOR SUCH ALTERNATIONS"*

Version

1.0	18/6/2024	*Web Application Security Assessment Report for acuver consulting*

Table of Contents

1 EXECUTIVESUMMARY

This report presents the results and findings of the Web Application Security Assessment conducted for "acuverconsulting.com" identified as the Vulnerability Assessment and Penetration Testing program. This assessment was performed by "cyber privilege Pvt Ltd". The purpose of this assessment was to identify vulnerabilities and other security issues that could affect the Web Application of "acuverconsulting.com". The findings in this report reflect the conditions found during the testing,

and do not necessarily reflect current conditions.

BACKGROUND

As part of secure solution and delivery model, "acuver consulting" wanted to identify security weaknesses or vulnerabilities affecting the Web Application of "acuver consulting". The security assessment was carried out as per the proposed methodology and also as briefly stated in section 2 of this report.

This report will provide "acuver consulting" the findings and observations from the Web Application Security Assessment along with the recommendations to fix the vulnerabilities.

2 ASSESSMENT METHODOLOGY

METHODOLOGY OVERVIEW

Web site/Application Security Assessment is a form of security testing used to analyse security posture of a web site/application. That built on OWASP Web Application Top 10 vulnerability standard. Web site/application security methodology is kept up-to-date according to changes in the threat environment and industry best practices provides consistency and structure to security testing. ITORIZIN TECHNOLOGY SOLUTIONS PVT LTD keeps its Web site/Application Security Assessment methodology updated with new tools, processes, techniques, or as trend develops. Our methodology is a comprehensive blend of the following methodologies and IT Security industry best practices:

Open-Source Security Testing Methodology Manual (OSSTMM) from the Institute for Security and Open Methodologies (ISECOM);
NIST SP 800-115 Technical Guide to Information Security Testing and Assessment.

TOOLS USED

Typical tools that were used in the vulnerability assessment:

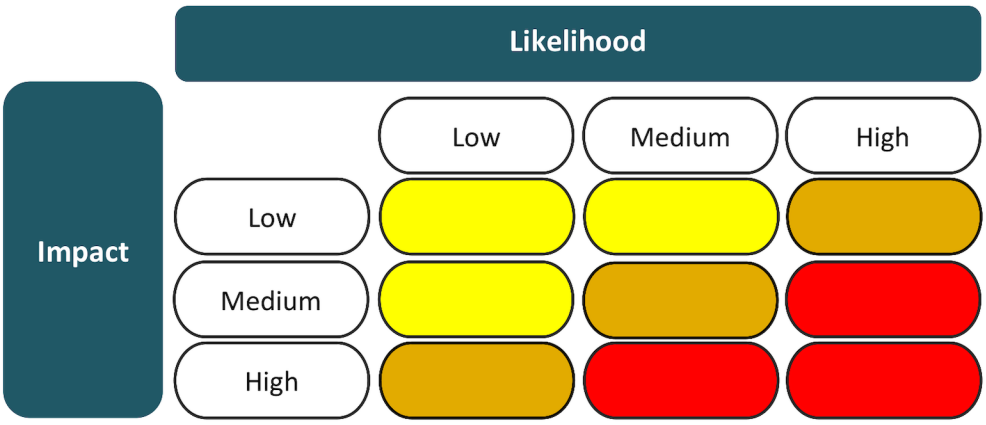
- **OWASP Zap**
- **Nmap**
- **Custom Scripts**

3 RISK RANKING APPROACH

Each assessment finding is assigned a risk rating. The risk rating used in this report is based on the following criteria:

High	Under specific conditions, these vulnerabilities can potentially make the system unusable and lead to serious security breaches. These vulnerabilities violate basic security design parameters and should be handled with the highest priority.
Medium	Though the threat is not critical at the moment, it has the potential to become a High-risk threat in the future under certain circumstances if not mitigated. Medium risk vulnerabilities require significant mitigation to lower the impact of
Low	The information found is useful to the attacker, but is not a threat in itself. Existing security controls are likely to be adequate or the risk is acceptable, but over the period this may give rise to more serious problems.
Info	The data revealed is an additional piece of information and there are no serious security implications related to it.

The likelihood of an attack occurring and the impact of a successful attack are used in calculating the risk rating:



4 SCOPEOFSECURITYASSESSMENT

Following “acuver consulting” targets were tested during the Vulnerability Assessment and Penetration Testing:

Target URLs
https://acuverconsulting.com/

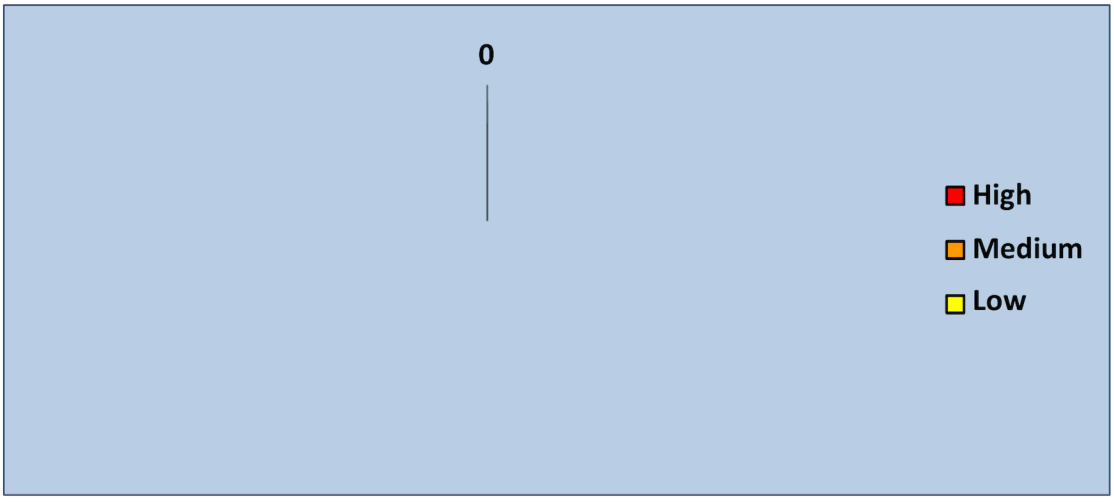
5 SUMMARYOFSECURITYASSESSMENTFINDINGS

The below sections list the count of total number of vulnerabilities found that are perceived as “High”, “Medium”, “Low” risks and also those which are informational in nature. The security assessment findings are also summarized in the below section.

SUMMARY ASSESSMENT RESULT:

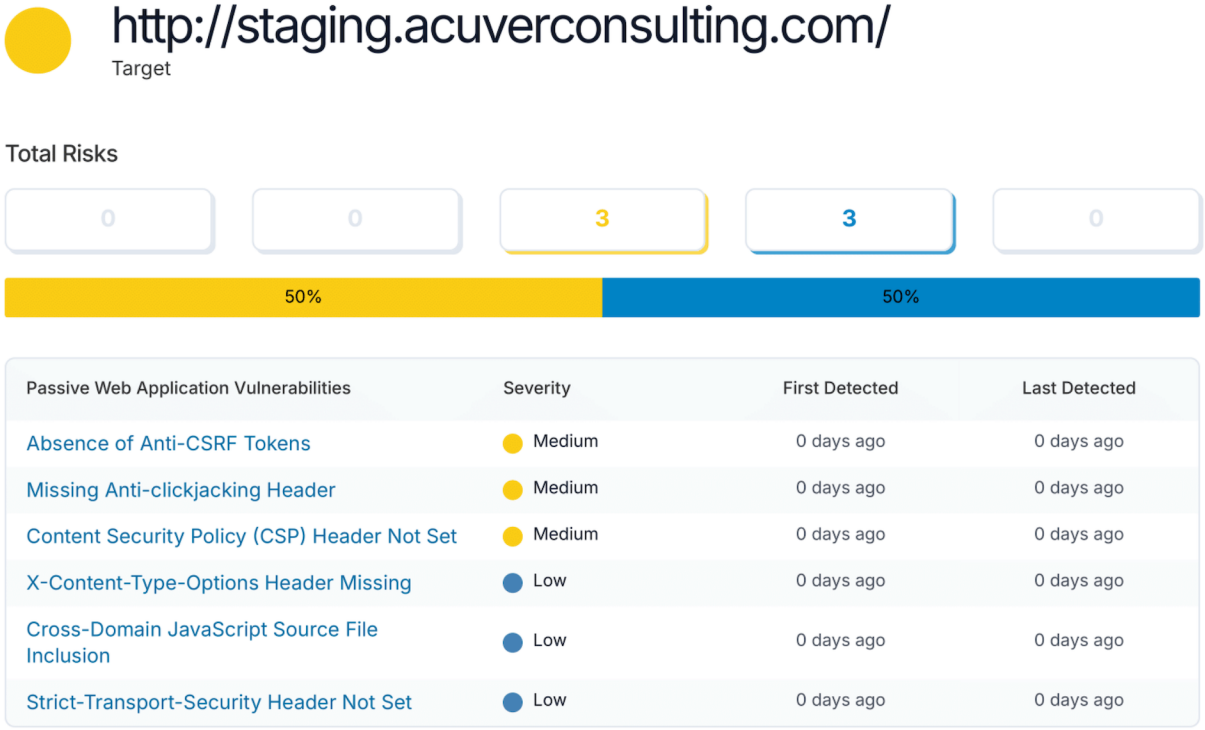
SUMMARY: TOTAL VULNERABILITY COUNT

1st Round A total of **twelve (9)** vulnerabilities were discovered during the assessment of which **Three (3)** vulnerabilities are of Medium-risk category and **Three (3)** vulnerabilities are of Low-risk category and **four (4)** vulnerabilities are of Informational-risk category. **2nd Round** No Vulnerabilities remains.



OWASP Top 2024 Ten Most Critical Web Application Vulnerabilities Mapping

S.No.	Security Risk	Present in Web Application
1	**Broken Authentication**	No (Examined & Not Found)
2	**Cryptographic Failures**	No (Examined & Not Found)
3	**Injection**	No (Examined & Not Found)
4	**Insecure Design**	No (Examined & Not Found)
5	**Security Misconfiguration**	No (Examined & Not Found)
6	**Vulnerable and Outdated Components**	No (Examined & Not Found)
7	**Identification and Authentication Failures**	No (Examined & Not Found)
8	**Software and Data Integrity Failures**	No (Examined & Not Found)
9	**Security Logging and Monitoring Failures**	No (Examined & Not Found)
10	**Server-Side Request forgery (SSRF)**	No (Examined & Not Found)



VULNERABILITY SUMMARY

The vulnerability findings have been summarized below along with their respective severity rating:

Absence of Anti-CSRF Tokens

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

No Anti-CSRF tokens were found in a HTML submission form.

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

CSRF attacks are effective in a number of situations, including:

* The victim has an active session on the target site.

* The victim is authenticated via HTTP auth on the target site.

* The victim is on the same local network as the target site.

CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Solution

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Phase: Implementation

Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.

Phase: Architecture and Design

Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).

Note that this can be bypassed using XSS.

Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Note that this can be bypassed using XSS.

Use the ESAPI Session Management control.

This control includes a component for CSRF.

Do not use the GET method for any request that triggers a state change.

Phase: Implementation

Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Instances (1 of 44)

uri: https://staging.acuverconsulting.com/
method: GET
evidence: <form class="search-form" action="https://staging.acuverconsulting.com/" method="get">
otherinfo: No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret, __csrf_magic, CSRF, _token, _csrf_token, data[_Token][key]] was found in the following HTML form: [Form 1: "search-field"].

References

https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
https://cwe.mitre.org/data/definitions/352.html

Missing Anti-clickjacking Header

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description
The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

Solution
Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Instances (1 of 27)
uri: https://staging.acuverconsulting.com/
method: GET
param: x-frame-options

References
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Content Security Policy (CSP) Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Solution
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Instances (1 of 28)
uri: https://staging.acuverconsulting.com/
method: GET

References
https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/
https://w3c.github.io/webappsec-csp/
https://web.dev/articles/csp
https://caniuse.com/#feat=contentsecuritypolicy
https://content-security-policy.com/

X-Content-Type-Options Header Missing

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description
The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution
Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Instances (1 of 100)
uri: https://staging.acuverconsulting.com/
method: GET
param: x-content-type-options
otherinfo: This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

References
https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
https://owasp.org/www-community/Security_Headers

Cross-Domain JavaScript Source File Inclusion

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description
The page includes one or more script files from a third-party domain.

Solution
Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Instances (1 of 26)
uri: https://staging.acuverconsulting.com/
method: GET
param: https://www.googletagmanager.com/gtag/js?id=UA-211005508-1
evidence: <script src="https://www.googletagmanager.com/gtag/js?id=UA-211005508-1" id="google_gtagjs-js" async></script>

Strict-Transport-Security Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description
HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Solution
Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Instances (1 of 100)
uri: https://staging.acuverconsulting.com/
method: GET

References
https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
https://caniuse.com/stricttransportsecurity
https://datatracker.ietf.org/doc/html/rfc6797

6 SECURITY FINDINGS AND ASSESSMENT DETAILS

DETAILED LIST: FINDINGS & RECOMMENDATIONS

This section presents the details of all the issues/findings which were identified during the assessment.

Out-of-date Version (core-js))	Risk level: low
Business Impact: Since this is an old version of the software, it may be vulnerable to attacks.	
Description: Since this is an old version of the software, it may be vulnerable to attacks. Description: The target website is using core-js, and it was detected that it is out of date. The web application is using version 3.31.0 of core-js. It is advised to keep the versions up-to-date A patch is available in version 3.37.0	

Out-of-date Version (wordpress))	Risk level: low
Business Impact: Since this is an old version of the software, it may be vulnerable to attacks.	
Description: Since this is an old version of the software, it may be vulnerable to attacks. Description: The target website is using **Wordpress**, and it was detected that it is out of date. The web application is using version 6.4.4 of Wordpress . It is advised to keep the versions up-to-date A patch is available in version 6.5	

OPEN PORTS	Risk level: low

Business Impact:

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

Description:

Since this is an old version of the software, it may be vulnerable to attacks.
Description:
The target website is using core-js, and it was detected that it is out of date. The web application is using version 3.31.0 of core-js. It is advised to keep the versions up-to-date
A patch is available in version 3.37.0

Out-of-date Version (wordpress))	Risk level: low

Business Impact:

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

Description:

Since this is an old version of the software, it may be vulnerable to attacks.
Description:
The target website is using **Wordpress**, and it was detected that it is out of date. The web application is using version 6.4.4 of Wordpress . It is advised to keep the versions up-to-date
A patch is available in version 6.5

Open Ports	Risk level: low

There are 3 open ports and that can be potentially harmfull in the near future the ports are
SSH(22)
HTTP(80)
HTTPS(443)

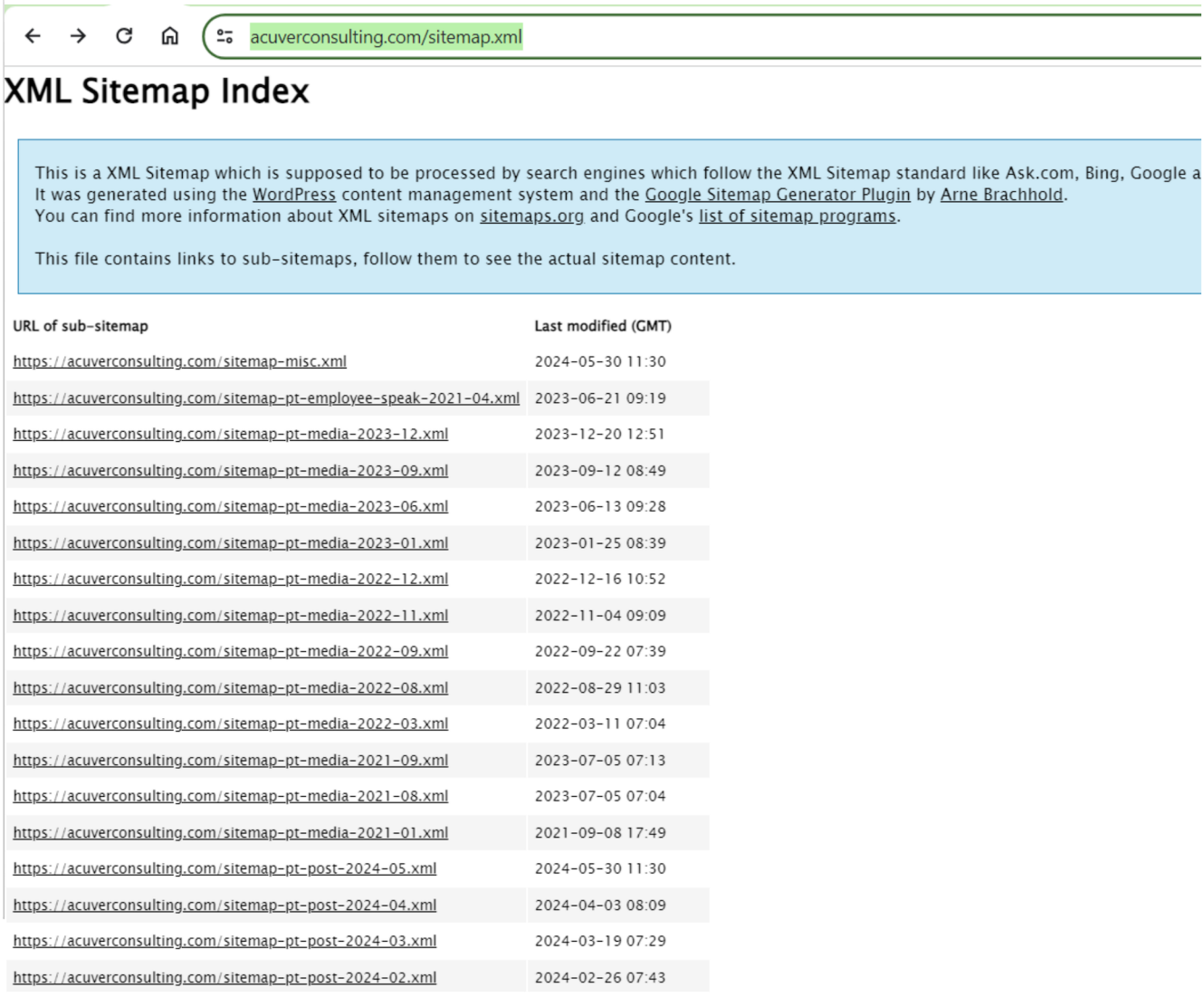
```
┌──(root㉿kali)-[~]
└─# nmap acuverconsulting.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-17 11:45 EDT
Nmap scan report for acuverconsulting.com (134.209.146.99)
Host is up (0.090s latency).
Other addresses for acuverconsulting.com (not scanned): 64:ff9b::86d1:9263
rDNS record for 134.209.146.99: 949702.cloudwaysapps.com
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
```

IF THESE ARE THE PORTS WHICH NEEDS TO KEPT OPEN IN THE WEB APPLICATION FOR THE FLOW OF CERTAIN SERVICES THEN IT SHOULD BE PROPERLY ENCRYPTED

FIX: If SSH, HTTP, and HTTPS ports are open on your website, it's important to ensure they are secured:

- SSH (port 22): Use strong passwords or SSH keys, and consider changing the default SSH port to reduce the risk of automated attacks.
- HTTP (port 80): Redirect all traffic to HTTPS to ensure data is encrypted.
- HTTPS (port 443): Use SSL/TLS certificates to secure your site.

Regularly update and patch your services, and consider using firewalls or intrusion detection systems for additional security



Sitemap Index **https://acuverconsulting.com/sitemap.xml**	Risk level: **Medium**	
Business Impact: It can reveal the structure of your site to competitors. and lead to Crawler error		

Description

WordPress Plugin Google XML Sitemaps is prone to a cross-site scripting vulnerability because it fails to properly sanitize user- supplied input. An attacker may leverage this issue to execute arbitrary script code in the browser of an unsuspecting user in the context of the affected site. This can allow the attacker to steal cookie-based authentication credentials and launch other attacks.

Remediation

Edit the source code to ensure that input is properly sanitised or disable the plugin until a fix is available

7 ASSESSMENT CONCLUSION

The overall analytical report is based on the technologies and known threats as of date of pen testing. We suggest that all recommendations mentioned in this Web Application Security Assessment Report should be undertaken in order to ensure the overall security of the concerned web application.

As a result of the assessment, we found that the current security posture of the Web application within the scope has no OWASP Top 10 vulnerability. We suggest that **“acuver consulting”** team to maintain the web application secured by conducting security audit at least once a year.