HACKEN

Web Infrastructure Risk Assessment Report

For Acme

Generated by Hacken

Sunday, November 22, 2020

This document includes confidential information regarding the IT systems and network infrastructure of the client, as well as information about potential vulnerabilities and the ways to exploit them.
The confidential information is for internal use by the client only and shall not be disclosed to third parties.

Document:

Name:	Web Infrastructure Risk Assessment (WIRA)
Type:	Automated Cyber Readiness Assessment Report
Revision:	Version 1
Date:	11/22/2020

Contents

Introduction

We thank Acme for giving us the opportunity to carry out the Automated Cyber Readiness Assessment. This document outlines the scope of work, our methodology, limitations, and outcomes of the assessment.

Executive Summary

Hacken OÜ (Consultant) was contracted by Acme to carry out the Automated Cyber Readiness Assessment of staging environment web application.

This report presents the findings of the security assessment of Network, Web & API security assessment that was carried out between 11/22/2020 - 11/23/2020.

The purpose of the engagement was to utilize active exploitation techniques to evaluate the security mechanisms of infrastructure and applications against best practices.

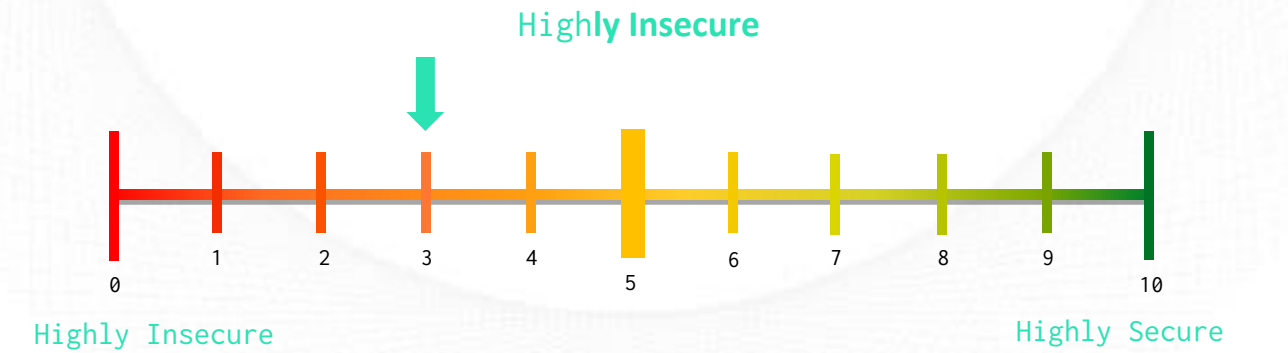
The assessment included an automated review of security controls and requirements listed in the OWASP Application Security Verification Standard (ASVS). This report isn't triaged by cybersecurity analytics so we cannot guarantee that it has covered all issues which are presented in the scope, and it can include false-positive results.

Next vulnerabilities and mistakes were identified during the assessment.

	High	Medium	Low	Informational
Overall	3	5	4	3

According to our research conducted after performing the security assessment, Infrastructure was identified as Low Secure level.

Overall security benchmark



The Overall rating of Acme, after the completion of the automated security assessment by the Consultant's Automated Assessment Tool, stands out to be 3 out of 10. The security assessment was carried out using automated tools.

Scope of the Security Assessment

The following list of the information systems constituted the scope of the Security Assessment.

- https://www.acme.com

Security Assessment start and end dates were communicated by email according to the following table:

Testing start date:	11/22/2020
Testing end date:	11/23/2020
Reporting:	11/23/2020

The security assessments main vectors are:

- Automated security assessment
  - Vulnerability Identification
  - Version Enumeration
  - Information Leakage
  - Vulnerability Exploitation
  - Brute Force Attacks
- Preparation of the final report with a detailed listing of findings, along with the associated risks and recommendations

Objectives

The assessment was conducted in an Automated mode (with an approved account) and had the following objectives:

- Identify technical and functional vulnerabilities.
- Estimate their severity level (ease of use, impact on information systems, etc.)
- Model the “most likely” attack vectors against the Customer’s Information System.
- Prove the concept and exploitation of vulnerabilities.
- Draw up a prioritized list of recommendations to address identified weaknesses.

WIRA Methodology

The methodology of Web Infrastructure Risk Assessment

Our methodology for Security Assessment is based on our own expertise, best practices in the area of information security, international methodologies, and guides such as PTES and OWASP.

Within the scope of this project, we have investigated the following functional domains:

- Intelligence gathering activities against a target;
- Service detection and identification;
- Vulnerabilities detection, verification, and analysis;
- The exploitation of vulnerabilities;
- Provision of recommendations aimed at addressing a security weakness.

Limitations and Assumptions

This project is limited by the scope of this document

During the implementation of the project, the Consultant will adhere to the following limitations:

- The operational impact on the networks will be maintained minimal and coordinated with the client;
- No active backdoor or Trojans will be installed;
- No client data will be copied, modified or destroyed.

The following security tests shall be considered Out of the Scope of this assessment:

- Internal networks assessment;
- Physical Social Engineering testing.

Disclaimer

This assessment was conducted for Acme prod environment and valid on the date of the report submission hereto. The description of findings, recommendations, and risks was valid on the date of the submission of the report hereto. Any projection to the future of the report's information is subject to risk due to changes in the Infrastructure architecture, and it may no longer reflect its logic and controls.

Definitions & Abbreviations

The level of criticality of each risk is determined based on the potential impact of loss from successful exploitation as well as ease of exploitation, the existence of exploits in public access, and other factors.

Risk Level	Description
High	High-level vulnerabilities are easy to exploit and may give an attacker full control of the affected systems, which may also lead to significant data loss or downtime. There are exploits or PoC available in public access.
Medium	Medium-level vulnerabilities are much harder to exploit and may not provide the same access to affected systems. No exploits or PoCs are available in public access. Exploitation provides only very limited access.
Low	Low-level vulnerabilities provide an attacker with information that may assist him in conducting further attacks against target information systems or against other information systems, which belong to an organization. Exploitation is extremely difficult and the impact is minimal.
Informational	These vulnerabilities are informational and can be ignored.

Summary of Findings

Value	Number of risks
High	3
Medium	5
Low	4
Informational	3

Based on our understanding of the environment, as well as the nature of the vulnerabilities discovered, their exploitability, and the potential impact we have assessed the level of risk for your organization as Low.

Key Findings

Risk level color map

High	Medium	Low	Informational

Cross site scripting

Description	*Type: Real*
Cross-site scripting (XSS) refers to a client-side code injection attack in which an attacker can execute malicious scripts into legitimate websites or web applications. XSS occurs when a web application uses unauthenticated or unencoded user input in its output.	
Details	URL encoded GET input sk was set to 19445'();}]9638 The input is reflected inside a <script> tag between single quotes.
GET /jobs/search?sk=19445'();}]9638 HTTP/1.1 Referer: https://www.acme.com/ Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Apply context-sensitive encoding and/or validation to user input presented on page.

Dotenv .env file

Description	*Type: Real*
The dotenv file (.env) was found in this directory. The Dotenv file is used to load environment variables from the .env file into the running process. The file may disclose sensitive information, which may help malicious users	

prepare for more advanced attacks. It is recommended to delete or restrict access to such files from the production system.	
Details	File: .env Pattern found: DB_HOST=
GET /.env HTTP/1.1 Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Delete or restrict access to all configuration files accessible from the Internet.

Git repository found

Description	*Type: Real*
The Git metadata directory (.git) was found in this folder. An attacker can extract sensitive information by requesting a hidden metadata directory created by the version control tool Git. The metadata catalog is used for development purposes to track development changes to a set of source code before submitting it back to the central repository (and vice versa). When the code is rolled from the repository to the active server, it should be done as an export instead of a local working copy, so this problem occurs.	
Details	Git files found at : /.git/config Repository files/directories: ▪ .gitignore ▪ .htaccess ▪ README.md ▪ admins/stats.php ▪ classes/calculations.php ▪ classes/category.php ▪ classes/chat.php

	▪ classes/company.php ▪ classes/cronjob.php ▪ classes/engine.php ▪ ...
GET /.git/config HTTP/1.1 Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Remove these files from production systems or restrict access to the .git directory. To deny access to all the .git folders you need to add the following lines in the appropriate context (either global config, or vhost/directory, or from .htaccess): <Directory ~ "\.git"> Order allow,deny Deny from all </Directory>

Development configuration file

Description	*Type: Real*
Find a configuration file (such as Vagrantfile, Gemfile, Rakefile, etc.) in this directory. The file may disclose sensitive information, which may help malicious users prepare for more advanced attacks. It is recommended to delete or restrict access to this type of file from the production system.	
Details	File info: composer.lock => Composer lock file. Composer is a dependency manager for PHP. Pattern found: "name": "phpmailer/phpmailer"

GET /composer.lock HTTP/1.1 Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive

Recommendations	Remove or restrict access to all configuration files acessible from internet.

HTML form without CSRF protection

Description	*Type: Real*
This alarm requires manual confirmation. Cross-site request forgery (CSRF, or XSRF) is a vulnerability in which an attacker can trick the victim into making a request that the victim does not want. Therefore, using CSRF, an attacker can abuse the trust of the web application in the victim's browser. WIRA found that HTML forms did not implement obvious anti-CSRF protection. Please refer to the "Attack Details" section for more information about the affected HTML forms.	

Details	Form name: <empty> Form action: /#wpcf7-f15514-o1 Form method: POST Form inputs: ▪ _wpcf7 [hidden] ▪ _wpcf7_version [hidden] ▪ _wpcf7_locale [hidden] ▪ _wpcf7_unit_tag [hidden] ▪ _wpcf7_container_post [hidden] ▪ _wpcf7_posted_data_hash [hidden] ▪ _wpcf7cf_hidden_group_fields [hidden] ▪ _wpcf7cf_hidden_groups [hidden] ▪ _wpcf7cf_visible_groups [hidden] ▪ _wpcf7cf_repeaters [hidden] ▪ _wpcf7cf_steps [hidden] ▪ _wpcf7cf_options [hidden]

	▪ your-email [email] ▪ <empty> [submit]
GET / HTTP/1.1 Referer: https://www.acme.com/ Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. ▪ The anti-CSRF token should be unique for each user session ▪ The session should automatically expire after a suitable amount of time ▪ The anti-CSRF token should be a cryptographically random value of ▪ significant length ▪ The anti-CSRF token should be cryptographically secure, that is, generated ▪ by a strong Pseudo-Random Number Generator (PRNG) algorithm ▪ The anti-CSRF token is added as a hidden field for forms, or within URLs ▪ (only necessary if GET requests cause state changes, that is, GET requests ▪ are not idempotent) ▪ The server should reject the requested action if the anti-CSRF token fails ▪ validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.

HTML form without CSRF protection

Description	*Type: Real*
This alarm requires manual confirmation. Cross-site request forgery (CSRF, or XSRF) is a vulnerability in which an attacker can trick the victim into making a request that the victim does not want. Therefore, using CSRF, an attacker can abuse the trust of the web application in the victim's browser. WIRA found that HTML forms did not implement obvious anti-CSRF protection. Please refer to the "Attack Details" section for more information about the affected HTML forms.	
Details	Form name: <empty> Form action: <empty> Form method: GET Form inputs: ▪ <empty> [email] ▪ <empty> [password] ▪ <empty> [checkbox] ▪ <empty> [submit]
GET /wp-content/what-input/ HTTP/1.1 Referer: https://www.acme.com/ Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. ▪ The anti-CSRF token should be unique for each user session ▪ The session should automatically expire after a suitable amount of time ▪ The anti-CSRF token should be a cryptographically random value of ▪ significant length

	▪ The anti-CSRF token should be cryptographically secure, that is, generated ▪ by a strong Pseudo-Random Number Generator (PRNG) algorithm ▪ The anti-CSRF token is added as a hidden field for forms, or within URLs ▪ (only necessary if GET requests cause state changes, that is, GET requests ▪ are not idempotent) ▪ The server should reject the requested action if the anti-CSRF token fails ▪ validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.

Vulnerable Javascript library

Description	*Type: Real*
You are using a vulnerable Javascript library. One or more vulnerabilities have been reported in this version of Javascript library. For more information on the affected libraries and the reported vulnerabilities, please consult the attack details and web reference.	
Details	Detected Javascript library jquery version 1.12.4. The version was detected from file content. References: ▪ https://github.com/jquery/jquery/issues/2432

```
GET /wp-includes/js/jquery/jquery.js HTTP/1.1
Referer: https://www.acme.com/
Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: www.acme.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Recommendations	Upgrade to the latest version.

Error message on page

Description	*Type: Real*
TThis alarm requires manual confirmation. Application error or warning messages may expose sensitive information about the internal workings of the application to an attacker. WIRA found errors or warning messages that could reveal sensitive information. The message may also contain the location of the file that generated the unhandled exception. Please refer to the "Attack Details" section for more information on the affected pages.	
Details	Pattern found: Fatal error
GET /defaults/header.php HTTP/1.1 Referer: https://www.acme.com/ Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a;searchSortBy=2 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Verify that this page is disclosing error or warning messages and properly configure the application to log errors to a file instead of displaying the error to the user.

Cookie(s) without HttpOnly flag set

Description	*Type: Real*

This cookie does not 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.	
Details	Set-Cookie: wp-postpass_bbd7e1dc5dcde6eb76f0f7681465b526=+; expires=Wed, 20-Nov-2019 16:42:41 GMT; Max-Age=0; path=/
GET / HTTP/1.1 Referer: https://www.acme.com/ User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com Connection: Keep-alive	
Recommendations	If possible, you should set the HttpOnly flag for this cookie.

Clickjacking: X-Frame-Options header missing

Description	*Type: Real*
Clickjacking (User Interface Correction Attack, UI Correction Attack, UI Correction) is a malicious technology that tricks Web users to click on content that is different from what the user thinks to click, which may leak confidential information or control the computer. Click on the seemingly harmless web page. The server did not return the X-Frame-Options header, which means that the site may be at risk of clickjacking attacks. The X-Frame-Options HTTP response header can be used to indicate whether the browser should be allowed to render the page in a frame or iframe. Websites can avoid clickjacking attacks by ensuring that their content is not embedded in other websites.	
Details	

```
GET / HTTP/1.1
Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813;wp-settings-
0=+;wpsettings-time-0=+;wp-postpass_bbd7e1dc5dcde6eb76f0f7681465b526=+
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: www.acme.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Recommendations	Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.

Clickjacking: X-Frame-Options header missing

Description	*Type: Real*
Clickjacking (User Interface Correction Attack, UI Correction Attack, UI Correction) is a malicious technology that tricks Web users to click on content that is different from what the user thinks to click, which may leak confidential information or control the computer. Click on the seemingly harmless web page. The server did not return the X-Frame-Options header, which means that the site may be at risk of clickjacking attacks. The X-Frame-Options HTTP response header can be used to indicate whether the browser should be allowed to render the page in a frame or iframe. Websites can avoid clickjacking attacks by ensuring that their content is not embedded in other websites.	
Details	

```
GET /admin/ HTTP/1.1
Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813;wp-settings-
0=+;wpsettings-time-0=+;wp-postpass_bbd7e1dc5dcde6eb76f0f7681465b526=+
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: www.acme.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Recommendations	Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.

Possible sensitive files

Description	*Type: Real*
A possible sensitive file has been found. This file is not directly linked from the website. This check looks for common sensitive resources like password files, configuration files, log files, include files, statistics data, database dumps. Each one of these files could help an attacker to learn more about his target.	
Details	
GET /test.php HTTP/1.1 Accept: acunetix/wvs Cookie: __cfduid=d6235b9bd2001ca89757792678fae7e581605542121;PHPSESSID=d1oh0lcvn2efnotdq4tt64qoe a Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	Restrict access to this file or remove it from the website.

Password type input with auto-complete enabled

Description	*Type: Real*
When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.	

Details	Form name: <empty> Form action: <empty> Form method: GET Form input: ▪ <empty> [password]

```
GET /wp-content/themes/what-input/ HTTP/1.1
Referer: https://www.acme.com/
Cookie: __cfduid=d450b241d72ad420a43de0b05c8e0088b1605803813
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: www.acme.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like
Gecko) Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: <INPUT TYPE="password" AUTOCOMPLETE="off">

Content Security Policy (CSP) not implemented

Description	*Type: Real*

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 ContentSecurity-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.

Details	
GET / HTTP/1.1 Referer: https://www.acme.com/ Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the ContentSecurity-Policy HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.

Reverse proxy detected

Description	*Type: Real*
This server uses a reverse proxy, a load balancer or a CDN (Content Delivery Network) or it's hosted in a cloud provider. WIRA detected this by sending various payloads and detecting changes in headers and body.	
Details	Detected reverse proxy: CloudFlare
GET / HTTP/1.1 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept-Encoding: gzip,deflate Host: www.acme.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36 Connection: Keep-alive	
Recommendations	None

HACKEN

Have any questions?

This report also provides for a free session with Hacken's cybersecurity specialist to help you understand the report and guide on how to avoid future security issues.
Please use the link below to book a timeslot.

GET FREE CONSULTATION

Generated by Hacken