Packet33

DEMO ACMEHR SAAS WEB APP P33

WEB APP PENTEST

Acme Corporation

02/12/2026

This assessment was performed in accordance with the Statement of Work, and the procedures were limited to those described in that agreement. The findings and the recommendations resulting from this assessment are provided in the attached report. Given the time-boxed scope of this assessment and its reliance on client-provided information, the findings in this report should not be taken as a comprehensive listing of all security issues.

This report is intended solely for information and use of **Acme Corporation**.

Street name 81

1000 Amsterdam

The Netherlands

DOCUMENT CONTROL

VERSIONS

Version	Status	Date	Editor	Comments
v1	FINAL	02/12/2026		Final report

PENTEST TEAM

Name	Email
[redacted]	[redacted]@packet33.com

DISTRIBUTION

Individual	Method
() –	Cyver Platform
() –	Cyver Platform
() –	Cyver Platform
() –	Cyver Platform

TABLE OF CONTENTS

1. EXECUTIVE REPORT

The assessment team conducted a security assessment of the Demo AcmeHR SaaS Web App P33. The objective of the security assessment was to identify and evaluate the security vulnerabilities present in the Demo AcmeHR SaaS Web App P33 and provide recommendations for improvement.

Approach

The assessment team primarily utilized a manual testing approach to test the web application's functionalities, business logic, and vulnerabilities in accordance with the OWASP Top 10 and OWASP Application Security Verification Standard (ASVS).

Throughout the web application penetration testing, we used a well-defined checklist to ensure comprehensive coverage and to reduce the likelihood of missing any potential vulnerabilities. This checklist was used as a systematic approach to assess the application's security posture.

The checklist was thoroughly put together based on industry best practices, widely accepted security standards such as OWASP (Open Web Application Security Project), and our significant expertise doing penetration testing assessments. It covered a wide range of potential flaws, attack vectors, and security controls.

Each item on the checklist was thoroughly inspected, and relevant tests were run to uncover any flaws or vulnerabilities. This method allows us to cover numerous areas of web application security in a methodical manner, such as authentication techniques, access controls, input validation, session management, data protection, and so on.

Goals

The objective of this assessment was to evaluate the security posture of the AcmeHR SaaS platform from the perspective of an authenticated and unauthenticated attacker. The testing focused on identifying vulnerabilities that could allow unauthorized access to customer data, cross-tenant data exposure, account compromise, or abuse of application functionality.

Special attention was given to authentication flows, authorization controls, data handling, and common web application risks defined by the OWASP Top 10. The goal was not only to identify technical weaknesses, but to determine their potential business impact to the confidentiality, integrity, and availability of customer information.

Results from this assessment are intended to help the organization prioritize remediation efforts and reduce the likelihood of a customer-impacting security incident.

EXECUTIVE SUMMARY

The security assessment of the **Demo AcmeHR SaaS Web App P33** was conducted between **February 2nd and February 9th, 2026**. The assessment focused on identifying vulnerabilities that could lead to unauthorized data access, tenant isolation breaches, or service disruption.

Key Findings & Security Posture

The assessment identified several critical areas where the application's security posture is vulnerable to exploitation:

Infrastructure Pivot Risks (SSRF): A **High-severity** Server-Side Request Forgery (SSRF) vulnerability was identified (F-2026-0003). An attacker can force the application to make unauthorized requests to internal-only systems or cloud metadata services (IMDSv2), potentially leading to a full environment takeover.

Data Isolation Failures (IDOR): A **High-severity** Insecure Direct Object Reference (IDOR) vulnerability was discovered (F-2026-0002). By manipulating resource identifiers, an authenticated user can access sensitive data belonging to other organizations, representing a high-risk failure of data isolation.

Authentication Weaknesses: The platform lacks strict rate limiting on authentication endpoints (F-2026-0004), exposing the application to automated brute-force attacks and mass account takeover (ATO) risks.

Supply Chain Risks: The use of an outdated version of *bootstrap-vue* (F-2026-0005) introduces known Cross-Site Scripting (XSS) vulnerabilities that could allow session token theft.

Identified vulnerabilities

Identified vulnerabilities	
Critical	0
High	2
Medium	2
Low	2
Info	1

Total: 7 findings

RECOMMENDATIONS

Packet33 recommends prioritizing the implementation of a **centralized authorization module** to validate object ownership for every request. Furthermore, strict **allow-listing for outbound requests** must be enforced to mitigate SSRF risks. Addressing these high-severity findings will significantly harden the platform's security posture and maintain client trust.

2. SCOPE

2.1 Scope

The assessment was conducted against the following primary assets and environments as defined in the Statement of Work (SOW).

Acme Corporation has mandated us to perform security tests on the following scope:

Asset	Type
APP API [https://api.acmehr.com]	Web Application
APP AUTH [https://auth.acmehr.com]	Web Application
App URL [https://app.acmehr.com]	Web Application

The testing activities were performed between 02/02/2026 and 02/09/2026.

2.2 Methodology & Depth

Assessment Type: Grey Box (Authenticated testing with 'Manager' and 'Employee' roles provided).

Testing Period: February 2nd – February 9th, 2026.

Source Code Access: No (Dynamic Analysis / DAST only).

2.3 Explicit Exclusions

To ensure business continuity, the following activities were strictly out of scope:

Denial of Service (DoS/DDoS): No resource exhaustion testing was performed.

Social Engineering: No phishing or physical security tests were conducted against Acme Corp employees.

Third-Party Integrations: Testing was limited to the AcmeHR codebase; external providers (e.g., Auth0, SendGrid) were not directly targeted.

3. ASSESSMENT REPORT

The assessment team performed Demo AcmeHR SaaS Web App P33 on the targets that are provided in the Scope.

VULNERABILITIES SUMMARY

Following vulnerabilities have been discovered:

Vulnerability	Severity	CVSS 3.1	Status
F-2026-0003 - Server-Side Request Forgery (SSRF)	High	8.6	Pending Fix
F-2026-0002 - Insecure Direct Object Reference (IDOR) / Broken Object Level Authorization	High	7.7	Pending Fix
F-2026-0005 - Vulnerable Third-Party Dependencies	Medium	6.1	Pending Fix
F-2026-0004 - Broken Rate Limiting on Authentication Endpoints	Medium	5.3	Pending Fix
F-2026-0006 - Missing Security Headers	Low	3.7	Pending Fix
F-2026-0007 - Cookie Missing 'Secure' and 'HttpOnly' Flags	Low	3.1	Pending Fix
F-2026-0008 - Exposure of Non-Sensitive Internal Documentation	Info	0.0	Pending Fix

4. IDENTIFIED ISSUES

F-2026-0003 - SERVER-SIDE REQUEST FORGERY (SSRF) - HIGH

Description

The application accepts user-supplied URLs to process images or webhooks but does not sufficiently validate the destination. This allows an attacker to "pivot" through the application server to make unauthorized requests to internal-only systems, cloud metadata services (e.g., IMDSv2), or other private infrastructure that is not directly accessible from the internet.

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

High

Impact Description

A successful SSRF attack can lead to the exposure of sensitive cloud environment credentials and internal network mapping. In cloud-hosted environments, this is often a precursor to a full environment takeover, as attackers can use stolen metadata tokens to escalate privileges across the cloud infrastructure..

Recommendations

Remediation

Enforce a strict "Allow-list" of permitted domains and IP ranges for outbound requests. If possible, utilize a dedicated egress proxy to isolate outbound traffic and disable access to internal metadata endpoints (e.g., `169.254.169.254`) at the network layer.

External References

- OWASP: SSRF Prevention Cheat Sheet
- CWE-918: Server-Side Request Forgery (SSRF)

F-2026-0002 - INSECURE DIRECT OBJECT REFERENCE (IDOR) / BROKEN OBJECT LEVEL AUTHORIZATION - HIGH

Description

The application fails to properly enforce authorization checks when accessing resource identifiers. By modifying request parameters (such as `user_id` or `org_id`) referencing specific resources, an authenticated user can access data belonging to other users or organizations. This indicates a failure in the application's "Object-Level" authorization logic.

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

High

Impact Description

This represents a high-risk failure of data isolation. An attacker could systematically "scrape" the database of sensitive customer records, leading to a massive data breach, loss of customer trust, and significant regulatory fines under frameworks like GDPR or CCPA.

Recommendations

Remediation

Implement a centralized authorization module that validates the requesting user's ownership of an object before returning data. Server-side logic must verify permissions for every request; never rely on user-supplied identifiers as the sole source of truth for access control.

External References

- OWASP: Authorization Cheat Sheet
- CWE-639: Access Control Bypass Through User-Controlled Key

F-2026-0005 - VULNERABLE THIRD-PARTY DEPENDENCIES - MEDIUM

Description

The frontend utilizes an outdated version of `bootstrap-vue` (v2.1.0) which is susceptible to multiple Cross-Site Scripting (XSS) vulnerabilities.

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

Medium

Impact Description

Exploitation could allow an attacker to execute malicious scripts in a user's browser, potentially stealing session tokens or manipulating SaaS dashboard data.

Recommendations

Remediation

Update `bootstrap-vue` to version 2.23.0 or higher. Integrate a Software Composition Analysis (SCA) tool into the CI/CD pipeline to flag vulnerable packages before deployment.

External References

- OWASP: Vulnerable and Outdated Components
- Snyk Advisor: bootstrap-vue Security Report

F-2026-0004 - BROKEN RATE LIMITING ON AUTHENTICATION ENDPOINTS - MEDIUM

Description

The application does not enforce strict rate limits on the `/api/v1/login` and `/api/v1/password-reset` endpoints. An attacker can perform automated brute-force attacks to guess user credentials or exhaust system resources.

Assets

- App URL [https://app.acmehr.com]
- APP API [https://api.acmehr.com]
- APP AUTH [https://auth.acmehr.com]

Classification

Severity

Medium

Impact Description

High risk of account takeover (ATO), leading to unauthorized access to tenant data and potential reputational damage due to "credential stuffing" incidents.

Recommendations

Remediation

Implement a sliding-window rate limiting strategy (e.g., using Redis or an API Gateway). Block or throttle IPs/accounts after 5 failed attempts within a 5-minute window.

External References

- OWASP: Blocking Brute Force Attacks
- CWE-307: Improper Restriction of Excessive Authentication Attempts

F-2026-0006 - MISSING SECURITY HEADERS - LOW

Description

The web server fails to implement critical security headers, specifically `Content-Security-Policy` (CSP) and `Strict-Transport-Security` (HSTS).

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

Low

Impact Description

Increases the platform's susceptibility to "clickjacking" and "Man-in-the-Middle" (MitM) attacks, though it requires specific user conditions to exploit.

Recommendations

Remediation

Configure the web server or Load Balancer to include a strict CSP and an HSTS header with a `max-age` of at least one year.

External References

- OWASP: Secure Headers Project
- CWE-693: Protection Mechanism Failure

F-2026-0007 - COOKIE MISSING 'SECURE' AND 'HTTPONLY' FLAGS - LOW

Description

The `marketing_preference` cookie is set without the `Secure` and `HttpOnly` attributes, making it accessible via client-side scripts and over unencrypted channels.

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

Low

Impact Description

While this specific cookie is not sensitive, the lack of these flags indicates a systemic configuration weakness that could eventually affect session cookies.

Recommendations

Remediation

Update the cookie-setting logic to include `HttpOnly; Secure; SameSite=Lax` for all cookies.

External References

- OWASP: Cookie Database - Secure Flag

F-2026-0008 - EXPOSURE OF NON-SENSITIVE INTERNAL DOCUMENTATION - INFO

Description

The `/docs/internal-style-guide` path is publicly accessible. It does not contain credentials but outlines internal UI components.

Assets

- App URL [https://app.acmehr.com]

Classification

Severity

Info

Impact Description

Negligible risk, but represents "low-hanging fruit" for reconnaissance during a targeted attack.

Recommendations

Remediation

Restrict access to this path to internal IP ranges or require SSO authentication.

External References

- OWASP: Information Disclosure Prevention
- CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

5. STEPS TO REPRODUCE

F-2026-0002: Insecure Direct Object Reference - IDOR (High)

Steps to Reproduce:

Log into the application as **User A** and navigate to a private resource (e.g., `/api/v1/invoices/123`).

Open a second browser session as **User B** (an unrelated user) and identify a resource ID belonging to them (e.g., `/api/v1/invoices/456`).

As **User A**, modify your API request to target User B's ID: `/api/v1/invoices/456`.

Observe that the server returns the sensitive data belonging to User B without verifying if User A has the authority to view it.

F-2026-0003: Server-Side Request Forgery (High)

Steps to Reproduce:

Log into the AcmeHR platform and navigate to the **Organization Settings** or **Document Import** feature.

Locate a field that requests a URL for an external resource (e.g., a logo or a webhook).

Enter the following internal cloud metadata endpoint: `http://169.254.169.254/latest/meta-data/`.

Submit the request while observing the traffic in **Burp Suite**.

Confirm that the application returns internal server information (like Instance ID or IAM Role names) in the response body.

F-2026-0004: Lack of Rate Limiting (Medium)

Steps to Reproduce:

Navigate to the login endpoint at `/auth/login`.

Capture a login request in **Burp Suite** and send it to the **Intruder** tool.

Configure a simple brute-force attack (e.g., testing 50 common passwords for a single username).

Run the attack and observe that the server returns a `200 OK` or `401 Unauthorized` response for every attempt.

The absence of a `429 Too Many Requests` status or an account lockout confirms the lack of rate limiting.

F-2026-0005: Vulnerable Third-Party Dependency (Medium)

Steps to Reproduce:

Access the web application and open the browser's **Developer Tools (F12)**.

Navigate to the **Network** or **Sources** tab and find the `bootstrap-vue.js` file.

Identify the version header (e.g., v2.21.0).

Search the **NVD (National Vulnerability Database)** for this version to identify known CVEs like **CVE-2021-42321**.

Check the console for warnings related to deprecated or insecure functions associated with this library.

F-2026-0006: Missing Security Headers (Low)

Steps to Reproduce:

Send a request to the application root (e.g., `https://app.acmehr.com/`).

Review the **HTTP Response Headers** in Burp Suite or browser DevTools.

Verify that `Content-Security-Policy`, `Strict-Transport-Security`, and `X-Frame-Options` are not present.

F-2026-0007: Insecure Cookie Flags (Low)

Steps to Reproduce:

Log in to the application and intercept the `Set-Cookie` header in the response.

Observe the attributes of the session cookie (e.g., `session_id`).

Confirm that the `HttpOnly` and `Secure` flags are missing from the string.

Verify by typing `javascript:alert(document.cookie)` in the browser address bar; if the session cookie is visible, the flag is not set.

F-2026-0008: Information Disclosure (Informational)

Steps to Reproduce:

Use a directory discovery tool (like `dirsearch` or `ffuf`) against the application root.

Identify a non-linked sensitive path, such as `/docs/api-internal.pdf` or `/tmp/config.bak`.

Request the file directly in a browser and observe that the server serves the file without authentication.

APPENDIX A - MEASUREMENT SCALES

Vulnerability Risk Severity

Packet33 utilizes a risk-based approach to categorize findings. Severity is determined by calculating the intersection of **Technical Impact** and **Exploitability**.

The risk severity of each finding in this report was determined independently from other findings. Vulnerabilities with a higher risk severity have more significant technical and business impact.

Severity	Description
Critical	Vulnerabilities that lead to full system compromise, administrative access, or large-scale data breaches. These represent an existential risk to the business and require **immediate remediation**.
High	Flaws that allow unauthorized access to sensitive data or the ability to bypass core security boundaries. These are high-priority risks that are easily exploitable by an attacker.
Medium	Vulnerabilities that could result in limited data exposure or a partial breakdown of security controls. These usually require specific configurations or user interaction to be successful.
Low	Issues that increase the "attack surface" or provide reconnaissance data to an attacker. They typically cannot be used to compromise the system on their own.
Info	Best practice recommendations and "Defense-in-Depth" findings. These have no direct security impact but improve the overall resilience of the environment.

CONFIDENTIALITY NOTICE

This report contains confidential and sensitive security information intended solely for Acme Corporation. Distribution to third parties should be limited to individuals with a legitimate business need, such as customers, auditors, or regulatory bodies. Public disclosure of specific vulnerabilities prior to remediation is not recommended.

REPORT VALIDITY

Security assessments reflect the state of the application at the time of testing only. Changes to the application, infrastructure, or configuration after the assessment date may introduce new vulnerabilities or invalidate previously reported results.

RETESTING

Packet33 offers verification testing to confirm remediation of identified vulnerabilities. Upon request, remediated findings can be validated and an updated verification letter can be issued.

CONTACT INFORMATION

Packet33, LLC
security@packet33.com
www.packet33.com

For questions regarding this report or remediation guidance, please contact the security team.