Acme Inc.

Bugcrowd Standard Pen Test (SPT)
Acme Ordering API

Created On
March 01, 2023
Prepared By
Advanced Security Group (ASG)
Reviewed By
Advanced Security Group (ASG)

Contents

Executive Summary

Acme Inc., based in San Francisco, California, is a large importer/exporter of numerous products. As a requirement of their business, Acme Inc. maintains their API that is used for ordering with the ordering web and mobile applications. These APIs are responsible for gathering and storing all types of data. Acme Inc. has a requirement and obligation to ensure that this application is resilient to cyber-attack in order to protect the privacy of their customers. To assist with this, Acme Inc. employed Bugcrowd to perform a Standard Pen Test (SPT) which took place from February 20, 2023, through February 25, 2023.

The purpose of this engagement was to identify security vulnerabilities in the assets listed under Targets and Scope. Once identified, each vulnerability was rated for technical impact defined in the Findings Summary section of the report.

To perform this test, our researcher leveraged several common tools to help identify and exploit vulnerable findings in the environment.

Manual testing of the scope was performed, evaluating the assets for weaknesses as per the Bugcrowd methodology. In support of this, active scanners and scripts were used in an attempt to identify any commonly found, known vulnerabilities.

At the time of this report, 5 findings were identified, including 1 Critical, 1 High, 1 Medium, 1 Low and 1 Informational vulnerability.

The highest priority vulnerabilities include:

- Broken Authentication and Session Management where the token management allows for multiple malicious activities.
- Server Security Misconfiguration where the token management allows for the token limit to be bypassed by using a Race Condition.

Bugcrowd has rated the overall risk to Acme Inc. – Orders API as Critical based on the Broken Authentication and Session Management, and the Server Security Misconfiguration. Our rating is based on the severity of the findings disclosed within this report.

It is recommended that Acme Inc. focus on critical and high severity issues first, with medium, low and informational findings being fixed once all high and critical issues are remediated.

Bugcrowd recommends that all critical, high and medium severity findings are retested once remediation activities are completed.

If not already implemented, Bugcrowd recommends taking the following high-level actions to further improve the overall security posture of the organization:

- Implement a secure development lifecycle such as Microsoft Secure Development Lifecycle (MSDL).
- Implement a static code analysis (SAST) tool into the development lifecycle to minimize the introduction of vulnerabilities in code.
- Provide regular secure development training to developers to ensure that they are aware of secure development practices and emerging threats.

The continuation of this document contains technical details of the specific vulnerabilities that were discovered throughout the PTS engagement. It should be noted that many of the details, including comments, up-to-date remediation status, images and additional contexts are not present in this document and are only available in the Bugcrowd Customer Portal.

If you have any questions or concerns as you move to remediate the items raised in this report, please do not hesitate to contact us. Bugcrowd would like to thank Acme Inc. for this engagement and look forward to working together in the future.

This report is just a summary of the information available and is a 'snapshot' in time of the state for the tested environment.

All details of the program's findings (comments, code, and any researcher provided remediation information) can be found in the Bugcrowd Crowdcontrol platform.

Reporting and Methodology

Bugcrowd Standard Pen Test (SPT) is an on-demand methodology-driven penetration test that delivers real-time results and 24/7 reporting in support of a variety of compliance initiatives. A pay-per-project model powered by CrowdMatch technology enables Bugcrowd to draw from a global network of continuously vetted pentesters to deliver faster setup without compromising on skill or experience. To support accelerated remediation and streamline integrated business processes, vulnerabilities discovered during the methodology are viewable live in the Bugcrowd Customer Console as soon as they are submitted by the pentester. Bugcrowd's in-house team of Security Engineers works in parallel to validate, prioritize, and push streaming vulnerabilities through customer-chosen SDLC integrations like GitHub, JIRA, or ServiceNow.

The Bugcrowd Pen Test service was designed and independently assessed by a leading QSA to ensure alignment with key compliance and regulatory standards.

Our unique BugHunter testing methodology blends key organizational and operational elements of leading industry standards to create a unified methodology that satisfies auditor and reviewer requirements.

By leading with a best-in-class testing approach, our methodology provides enhanced risk reduction while supporting critical compliance initiatives. A review of these standards follows.

Organizational Methodology Standards

Executing a penetration test involves a proven workflow that is split into phases. Each of these phases is run in a cyclical manner allowing penetration testers to build upon findings and potentially uncover significant risk. An organizational methodology also ensures that high-level coverage of testing is done. When reviewing common organizational methodologies, Bugcrowd found similarities in the general workflow.

Bugcrowd pen testers adhere to these standards in a common workflow as shown:

Reviewed Organizational Methodology Standards:

- PCI DSS Requirement 11.2, 11.3.1, 11.3.3, 11.3.4
- NIST 800-115 - Technical Guide to Information Security Testing and Assessment 2.1 “Information Security Assessment Methodology”
- Open Source Security Testing Methodology Manual (OSSTMM)
- Penetration Testing Execution Standard (PTES)

Operational Methodology Standards

Many penetration testing models fail to provide both results and coverage. To bring value to the customer, Bugcrowd has reviewed the most common and in-depth operational Pen Test methodologies. Operational methodologies provide details on what exactly needs to be tested in a security assessment, for each endpoint.

The methodology assigned to researchers includes application and infrastructure level testing domains. Each domain contains several tests for the tester to cover in both manual and automated methods.

In order to create a complete testing methodology, Bugcrowd has pulled from the following industry standard operational methodologies:

- OWASP Testing Guide (OTG)
- Web Application Hacker Handbook Methodology (WAHHM)
- Others where applicable (SANS Top 25, CREST, WASC, PTES)

Findings Summary

Targets and Scope

Prior to penetration test launching, Bugcrowd worked with Acme Inc. to define the rules of the engagement, commonly known as the program brief, which includes the scope of work.

The following targets were considered explicitly in scope for testing:

- Acme Inc. Ordering API

The following items are explicitly out-of-scope:

- Denial of Service attacks (DoS)

All details of the program scope and full program brief are available in the Program Brief found on the Bugcrowd Crowdcontrol platform.

Risk and Priority Key

The following priority keys are used to explain how Bugcrowd rates valid vulnerability submissions and their technical severity. As a trusted advisor, Bugcrowd provides common next steps for program owners per severity category.

Priority	Technical Severity	Example Vulnerability Types
P1	Critical Severity vulnerabilities are escalated as soon as they are validated. These issues warrant the highest security consideration and should be addressed immediately.	• Remote Code Execution • Vertical Authentication Bypass • SQL Injection • Insecure Direct Object Reference for a critical function
P2	High Severity vulnerabilities should be slated for a fix very soon. These issues still warrant prudent consideration but are often not availability or "breach level" submissions.	• Lateral Authentication Bypass • Stored Cross-Site Scripting • Cross-Site Request Forgery for a critical function • Insecure Direct Object Reference for an important function
P3	Medium Severity vulnerabilities should be slated for remediation in a major release cycle.	• Reflected Cross-Site Scripting • Cross-Site Request Forgery for an important function • Insecure Direct Object Reference for an unimportant function
P4	Low Severity vulnerabilities should be considered for remediation within the next six months.	• Cross-Site Scripting with limited impact • Cross-Site Request Forgery for an unimportant function • External Server-Side Request Forgery
P5	Informational findings are environmental information, best practices or potential accepted business risk.	• All open ports • Lack of Code Obfuscation • Autocomplete enabled • Non-exploitable SSL issues

Findings Table

Index	Title	VRT	Priority
F001	Generate, Enable, and Delete Token Using Expired Token	Broken Authentication and Session Management	P1
F002	Generate a Token Using Disabled Token	Broken Authentication and Session Management	P2
F003	API Token Limit Bypass Using Race Condition, Leads to Misuse or Financial Loss to the Company	Server Security Misconfiguration	P3
F004	Insufficient Rate Limit on Create a User Resource	Server Security Misconfiguration	P4
F005	Session Timeout Too Long	Server Security Misconfiguration	P5

Vulnerability Details

This section outlines the full submission data for each valid finding. These findings are rarely altered from their original state from the researcher. Due to the nature of crowd-sourced security assessments, some typos or grammar errors may occur. Each finding is headlined with the submission title and priority followed by more detailed vulnerability information based on the type of finding submitted. Several other fields may appear based on the context and Vulnerability Rating Taxonomy (VRT) classification selected by a researcher.

The details may include the following:

Description

This section appears after the "Bug URL" as a free form area for the researcher to describe the context of the submission.

Bug URL

This is the specific URL path or IP target location where the vulnerability was found.

Submission Reference Number

The unique identifier for the submission visible to researchers.

CVSS Rating

The CVSS vector string for this submission, if provided, and the score calculated from that vector string.

Vulnerability Rating Taxonomy (VRT)

The Vulnerability Rating Taxonomy is the baseline guide used for classifying technical severity.

Additional Details

Several other fields may appear based on the context and VRT classification selected by a researcher, such as but not limited to Bugcrowd Application Security Engineer (ASE) curated proof of concepts, comments to the researcher or Bugcrowd, assignees, attachments, and state change metadata. These can be viewed in the Crowdcontrol platform.

F001 - Generate, Enable, and Delete Token Using Expired Token

P1

Submission Reference Number
Dc61db1253b1a6887c52fb923c58923c81d602f10fc272a8ddb

Vulnerability Rating Taxonomy (VRT)
Broken Authentication and Session Management > Privilege Escalation

Description

Authentication and session management controls can be bypassed through a variety of ways including, calling an internal post authentication page, modifying the given URL parameters, by manipulating the form, or by counterfeiting sessions. The authentication method for this application can be bypassed by an attacker which enables them to access a privileged user's account and functionality, giving them access to more resources or functionality within the application. This could include viewing or editing sensitive customer data, and viewing or editing other user permissions.

Business Impact:

The impact of privilege escalation through broken authentication controls can vary in severity depending on the degree of access to resources or functionality the malicious attacker is able to gain. An attacker with the ability to access, delete, or modify data from within the application could result in reputational damage for the business through the impact to customers' trust. This can also result in indirect financial cost to the business through fines and regulatory bodies if sensitive data is accessed. The severity of the impact to the business is dependent on the sensitivity of the data being stored in, and transmitted by the application.

Steps to Reproduce:

1. Enable a HTTP interception proxy, such as Burp Suite or OWASP ZAP
2. Use a browser to navigate to: REDACTED
3. Login to the application

Finding Truncated for Display Purposes

F002 - Generate a Token Using Disabled Token

P2

Submission Reference Number
727992fe76d053e67d68fcaeb77fa88ec280b1190ee935ed

Vulnerability Rating Taxonomy (VRT)
Broken Authentication and Session Management > Privilege Escalation

Description

Privilege Escalation wherein a disabled token can generate a new token.

Steps to Reproduce:

1. Enable a HTTP interception proxy, such as Burp Suite or OWASP ZAP
2. Navigate to the following endpoint using a browser: REDACTED
3. Generate a new token and then disable the given token
4. Load (GET or POST /v0/token) and replace auth bearer token using the disabled token
5. Using the HTTP intercept proxy, re-issue the captured request to generate a new token

HTTP Request:

Finding Truncated for Display Purposes

F003 - API Token Limit Bypass Using Race Condition, Leads to Misuse or Financial Loss to the Company

P3

Reference Number
4e7b75cbf48b0ffcf89e5ba098b85e7c1b134c186afafc06a

VRT
Server Security Misconfiguration > Race Condition

Description

Steps to Reproduce:

1. Login to your account
2. Click on generate Token and take this request to Burp Intruder
3. As a null payload run the attack to 100 payloads with 15 number of threads

Observe that more than 5 API Tokens are generated that are bypassed. Upon checking it is found that the limit is 5 and there have been 9 API tokens created. This is beyond the allowed limit.

Finding Truncated for Display Purposes

F004 - Insufficient Rate Limit on Create a User Resource

P4

Submission Reference Number
dacba07473fe7c26e8dc6a626f7da6a79e5d9be31d07b61f

Vulnerability Rating Taxonomy (VRT)
Server Security Misconfiguration > Lack of Security Headers > Strict-Transport-Security

Vulnerability

Rate limiting is a strategy to limit the frequency of a repeat action within a particular time frame. This ensures that a service doesn't become unresponsive or unavailable due to too many requests exhausting the application's resources. A lack of rate limiting on this endpoint allows an attacker to send a large number of requests to the server and potentially cause accelerated service usage for the business or exhaust the application resources.

Business Impact:

Insufficient rate limiting on an endpoint can result in reputational damage to the organization if the rate limiting prevents legitimate endpoint submissions and responses. It also has the potential to cause accelerated service usage, which can incur a direct financial cost in environments with SaaS services or pay on demand systems.

Steps to Reproduce:

1. Enable a HTTP interception proxy, such as Burp Suite or OWASP ZAP
2. Navigate to the following endpoints using a browser and Generate a token: REDACTED
3. Load (POST /users) and replace auth bearer token using generated token

Finding Truncated for Display Purposes

F005 - Session Timeout Too Long

P5

Submission Reference Number
D773d2e03558533a3e4f7c4dd7d4335bbc4f07499eb8e5a18

Vulnerability Rating Taxonomy (VRT)
Broken Authentication and Session Management > Failure to Invalidate Session > Long Timeout

Vulnerability

Session timeout defines the amount of time a session will remain active in case there is no activity in the session, closing and invalidating the session upon the defined idle period since the last HTTP request received by the web application for a given session ID.

Session termination is an important part of the session lifecycle. Reducing to a minimum the lifetime of the session tokens decreases the likelihood of a successful session hijacking attack. This can be seen as a control against preventing other attacks like Cross Site Scripting and Cross Site Request Forgery. Such attacks have been known to rely on a user having an authenticated session present. Not having a secure session termination only increases the attack surface for any of these attacks.

A secure session termination requires at least the following components:

- Availability of user interface controls that allow the user to manually log out. Session termination after a given amount of time without activity (session timeout)
- Proper invalidation of server-side session state

Finding Truncated for Display Purposes

Closing Statement

Bugcrowd Inc.
921 Front Street
Suite 100
San Francisco, CA 94111

March 01, 2023

Summary

This report shows testing of Acme Inc. – Ordering API from February 20, 2023, to February 25, 2023. The purpose of this assessment was to identify security issues that could adversely affect the integrity of Acme Inc. – Ordering API. The assessment was performed under the guidelines provided in the statement of work between Acme Inc. and Bugcrowd. This document provides a high-level overview of the testing performed and the test results.

Pen Test Portfolio Overview

The Bugcrowd Pen Test portfolio provides organizations with the power of the Crowd, through two unique engagement styles designed to fit a range of security workflows and objectives. Max Pen Test (MPT), Plus Pen Test (PPT) and Standard Pen Test (SPT) are all powered by the Bugcrowd platform, enabling rapid setup, launch, and real-time results.

While Bugcrowd offers both continuous and on-demand penetration testing options, it is important to note that this document represents a point-in-time evaluation of security posture. Security threats and attacker techniques evolve rapidly, and the results of this assessment are not intended to represent an endorsement of the adequacy of current security measures against future threats. This document contains information in summary form and is therefore intended for general guidance only; it is not intended as a substitute for detailed research or the exercise of professional judgment. The information presented here should not be construed as professional advice or service.

Testing Methods

This security assessment leveraged researchers that used a combination of proprietary, public, automated, and manual test techniques throughout the assessment. Commonly tested vulnerabilities include code injection, cross-site request forgery, cross-site scripting, insecure storage of sensitive data, authorization/authentication vulnerabilities, business logic vulnerabilities, and more.

The summary of Bugcrowd's findings are as follows:

1 Critical **1 High** **1 Medium** **1 Low** **1 Informational**