INTIGRITI

Letter of Attestation

Example Comp. Inc.

Creation Date: October 1, 2024

Version control

Version	Date	Editor	Overview of changes
0.1	September 27, 2024	John Doe	Created Example Comp. Inc. penetration test report draft.
0.2	September 28, 2024	Jane Doe	Reviewed Example Comp. Inc. penetration test report.
1.0	September 29, 2024	Jane Doe	Finalized Example Comp. Inc. penetration test report.

Table of contents

Introduction

This document is created as evidence for our customer Example Inc, explaining the results of their penetration test on the Intigriti platform.

Intigriti is a cloud solution, providing an ethical hacking platform to companies that desire a structured bug bounty & hybrid pentest program.

Intigriti's hybrid pentest is delivered via the crowdsourced security platform allowing vetted security researchers to engage and communicate with companies quickly, safely, and reliably, offering live updates and communication about found vulnerabilities.

Based on the customer's predefined scope of the hybrid pentest program, a hand-picked researcher has searched for vulnerabilities and reported their submissions through Intigriti's platform.

Benefits of hybrid pentesting

Penetration test with bug bounty benefits

Intigriti's hybrid pentest offering consists of a traditional penetration test but with the motivation, reporting, triage, and rewards of a bug bounty program.

Specialized skills

Penetration testers are hand-picked; selection is based on researcher specialism and activity as well as test criteria.

Transparent researcher selection

Researchers are selected based on previous ratings, quality, motivation, expertise, and skillset.

Work with experts in your field

Gain industry-tailored security insights from researchers who understand your sector's unique challenges and demands: Fintech, Retail, E-commerce, Media, Health, etc.

Highly motivated penetration testers

Researcher receives an effort-based fee based and a capped bounty fee on top for all accepted submissions.

Data-driven platform benefits

All submissions are real-time reported via the Intigriti platform.

Executive summary

In September 2024, Example Comp. Inc. engaged Intigriti to perform a hybrid penetration test with one of Intigriti's vetted researchers eligible for pentests (see researcher information below).

The hybrid pentest was executed as a black-box assessment, meaning that no access to source code was available. The Intigriti researcher had direct 24/7 access to communication with the Example Inc security and development team.

A total of 4 findings were reported during the assessment including 1 critical, 1 high, 1 medium and 1 low vulnerabilities. The key issue found presented a SQL Injection that was found in the public-facing web application, which could allow unauthorized access to the database.

The Acme Corporation team, together with Intigriti has identified all steps needed to remediate the found issues. Software fixes will be implemented in-line with the Acme Corporation vulnerability remediation program.

Delivery

Goals and objectives

The main objective of this penetration test was to test for the OWASP Top 10 and OWASP API Top 10 vulnerability categories.

A special focus was set on testing for:

- PII data leaks
- Vertical privilege escalations
- Cross-tenancy vulnerabilities

Assets

Assets in scope

- Domains
  - www.example.com/*
  - www.example2.com/users/*
  - www.subdomain1.example.com/*
  - www.subdomain2.example.com/*

- Android Applications
  - com.example.androidapplication

Timeframe

This penetration test was executed between September 1, 2024, and September 25, 2024. A total of 80 hours of testing was performed on all the assets that were in scope.

Methodology

Depending on the scope of the assessment, Intigriti's vetted researcher base is following the methodologies and standards discussed in this chapter.

Web application

During the security assessment of a web application, an extensive range of vulnerabilities is tested for, including those defined in the OWASP Top 10 – 2021:

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures
- A10:2021-Server-Side Request Forgery

A common methodology that is followed is the OWASP Web Security Testing Guide.

API

For the security assessments of API's, the focus of testing lies on the security vulnerabilities defined in the OWASP API Top 10 - 2019:

- API1:2019-Broken Object Level Authorization
- API2:2019-Broken User Authentication
- API3:2019-Excessive Data Exposure
- API4:2019-Lack of Resources & Rate Limiting
- API5:2019-Broken Function Level Authorization
- API6:2019-Mass Assignment
- API7:2019-Security Misconfiguration
- API8:2019-Injection
- API9:2019-Improper Assets Management
- API10:2019-Insufficient Logging & Monitoring

Mobile application

During the security assessment of mobile applications, the OWASP MAS checklist and the OWASP Mobile Application Security Testing Guide play a predominant role in test coverage. These include vulnerabilities out of the following categories:

- Architecture, Design and Threat Modelling
- Data Storage and Privacy
- Cryptography
- Authentication and Session Management
- Network Communication
- Platform Interaction
- Code Quality and Build Setting
- Resilience

Personnel

Researcher

	Researcher details
Username	Security_researcher_1
Intigriti ranking all time	#1
Reputation all time	1337 pts
Current submission streak	Exceptional
Profile	https://app.intigriti.com/profile/security_researcher_1

Findings

Finding overview

During this penetration test, a total of 18 vulnerabilities were identified. An overview of all vulnerabilities is broken down by severity and asset in the table below:

Assets in scope	Informative	Low	Medium	High	Critical	Exceptional	Total
https://example.com	0	2	2	0	3	1	8
www.subdomain1.example.com/*	0	0	4	1	1	0	6
com.example.androidapplication	0	2	0	0	1	1	4
Total	0	1	1	1	1	0	18

The accepted vulnerabilities broken down per vulnerability type:

Vulnerability type	Informative	Low	Medium	High	Critical	Exceptional	Total
Vertical Privilege Escalation	0	0	0	0	0	2	2
Blind SQL-Injection	0	0	0	0	1	0	1
Unauthenticated access to public MongoDB instance	0	0	0	0	1	0	1
Insecure Direct Object Reference	0	0	0	0	3	0	3
Subdomain Takeover	0	0	0	1	0	0	1
Business Logic Error	0	0	1	0	0	0	1
Improper Access Control	0	0	1	0	0	0	1
Stored Cross-Site Scripting	0	0	4	0	0	0	4
Broken Access Control	0	1	0	0	0	0	1
Sensitive Data Exposure	0	1	0	0	0	0	1
Security Misconfiguration	0	1	0	0	0	0	1
Path Traversal	0	1	0	0	0	0	1
Total	0	4	6	1	5	2	18

Finding Details

This section provides an in-depth look at all vulnerabilities which have been discovered throughout this penetration test engagement. The list of vulnerabilities has been prioritized following the severity level of each finding starting with the most critical one on top. Each vulnerability found is presented with the following information:

- Affected asset and endpoint
- Vulnerability type
- Severity of the vulnerability
- CVSS score
- Summarized Impact

EXCOMINC-ZBW33H4G

Metadata

Field	Value
Vulnerability type:	Vertical Privilege Escalation
Severity:	Exceptional
CVSS score (vector):	9.8
Endpoint / vulnerable component:	www.example.com/login

Summarized Impact:

User role “viewer” can become an admin by setting “rolePermMatrix” POST parameter to 1 within user administration settings. Admin role can be used to obtain access to and change all user data.

EXCOMINC-47ISHSLJ

Metadata

Field	Value
Vulnerability type:	Vertical Privilege Escalation
Severity:	Exceptional
CVSS score (vector):	9.5
Endpoint / vulnerable component:	com.example.androidapplication

Summarized Impact:

Admin privileges can be obtained for the user role “guest” by modifying a cookie set upon the site’s first visit. This results in the ability for any guest user to impact the data on the platform.

EXCOMINC-H74GGXBW

Metadata

Field	Value
Vulnerability type:	Blind SQL-Injection
Severity:	Critical
CVSS score (vector):	9.1
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

An attacker can dump all data from the database.

EXCOMINC-SU28DJUI

Metadata

Field	Value
Vulnerability type:	Unauthenticated access to public MongoDB instance
Severity:	Critical
CVSS score (vector):	9.0
Endpoint / vulnerable component:	com.example.androidapplication

Summarized Impact:

An attacker can anonymously login to the publicly exposed MongoDB instance to get access to all data.

EXCOMINC-8SI29UI

Metadata

Field	Value
Vulnerability type:	Insecure Direct Object Reference
Severity:	Critical
CVSS score (vector):	9.0
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

An attacker can get access to any user’s personal records.

EXCOMINC-OIC2SKL9

Metadata

Field	Value
Vulnerability type:	Insecure Direct Object Reference
Severity:	Critical
CVSS score (vector):	9.0
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

Confidential support chat transcripts can be retrieved by an attacker by modifying the URL.

EXCOMINC-UU7LF88U

Metadata

Field	Value
Vulnerability type:	Insecure Direct Object Reference
Severity:	Critical
CVSS score (vector):	9.0
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

By modifying the request sent to pull inventory data, an attacker can gain access to the inventory back-end database.

EXCOMINC-WUV722JK

Metadata

Field	Value
Vulnerability type:	Subdomain Takeover via dangling DNS record
Severity:	High
CVSS score (vector):	8.8
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

Controlling the subdomain, an attacker can serve malicious content trusted by company users, using the domain for phishing purposes or e.g. for stealing session cookies enabling account takeovers.

EXCOMINC-LO19FUQP

Metadata

Field	Value
Vulnerability type:	Business Logic Error
Severity:	Medium
CVSS score (vector):	6.9
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

2FA TOTP token is not bound to user allowing an attacker to create a valid token to log in.

EXCOMINC-KK27VK9U

Metadata

Field	Value
Vulnerability type:	Improper Access Control
Severity:	Medium
CVSS score (vector):	6.5
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

An attacker can use all endpoints under /api/example/read/* without having the right permission group set.

EXCOMINC-89SUIKQN

Metadata

Field	Value
Vulnerability type:	Stored Cross-Site Scripting
Severity:	Medium
CVSS score (vector):	6.3
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

An attacker can use this XSS vulnerability to post victim's private information as a comment in thread functionality.

EXCOMINC-NM23FLIO

Metadata

Field	Value
Vulnerability type:	Stored Cross-Site Scripting
Severity:	Medium
CVSS score (vector):	6.3
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

The XSS vulnerability can be used to run malicious JavaScript within the browser, by exploiting the commenting functionality.

EXCOMINC-QU7V7890

Metadata

Field	Value
Vulnerability type:	Stored Cross-Site Scripting
Severity:	Medium
CVSS score (vector):	6.0
Endpoint / vulnerable component:	www.subdomain1.example.com/*

Summarized Impact:

The commenting functionality can be exploited by the attacker to send malicious JavaScript to any user browsing the site.

EXCOMINC-12VHLSAOI

Metadata

Field	Value
Vulnerability type:	Stored Cross-Site Scripting
Severity:	Medium
CVSS score (vector):	5.9
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

Client-side scripts can be used to steal customer's cookies and personal information.

EXCOMINC-YXMI22HH

Metadata

Field	Value
Vulnerability type:	Broken Access Control
Severity:	Low
CVSS score (vector):	3.9
Endpoint / vulnerable component:	com.example.androidapplication

Summarized Impact:

Attacker can get information if a certain email address has already registered a user.

EXCOMINC-VHOSH2JK

Metadata

Field	Value
Vulnerability type:	Sensitive Data Exposure
Severity:	Low
CVSS score (vector):	2.9
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

Stacktrace exposes the tech stack and plugins used by the application.

EXCOMINC-VLSJU2KL

Metadata

Field	Value
Vulnerability type:	Security Misconfiguration
Severity:	Low
CVSS score (vector):	2.2
Endpoint / vulnerable component:	com.example.androidapplication

Summarized Impact:

CAPTCHA can be bypassed by setting setting “valid” parameter to “true”.

EXCOMINC-UVKL2LUU

Metadata

Field	Value
Vulnerability type:	Path Traversal
Severity:	Low
CVSS score (vector):	2.1
Endpoint / vulnerable component:	www.example.com

Summarized Impact:

Attacker can upload arbitrary files to any location.

Appendix

Evidence

Additional attachments to the penetration test can be requested by contacting the Intigriti pentest coordinator.

Glossary

Black box testing: A type of penetration testing where the researcher has no prior knowledge of the internal workings or architecture of the system being tested, simulating an outsider's perspective.

CVSS: Common Vulnerability Scoring System

Exploit: A piece of software, tool, or technique used to take advantage of a vulnerability in a system, typically to gain unauthorized access or execute malicious code.

Penetration testing, or pentesting: A security testing method used to identify vulnerabilities in a system, network, or application by attempting to exploit them, simulating real-world attacks.

Risk assessment: The process of identifying, analyzing, and prioritizing potential security risks to an organization's assets, considering the likelihood and impact of various threats.

Risk mitigation: The process of reducing or eliminating the potential impact of identified risks through proactive measures such as implementing security controls, policies, and procedures.

Vulnerability: A weakness or flaw in a system's design, implementation, or configuration that could be exploited by an attacker to compromise the confidentiality, integrity, or availability of the system.

White box testing: A type of penetration testing where the researcher has full knowledge of the internal workings and architecture of the system being tested, simulating an insider's perspective.