ENTERSOFT SECURITY

2020

API CRITIQUE

Issued by Entersoft Security R&D

Most Compreshensive Penetration Tester for APIs

Table of Contents

Preface

Preface

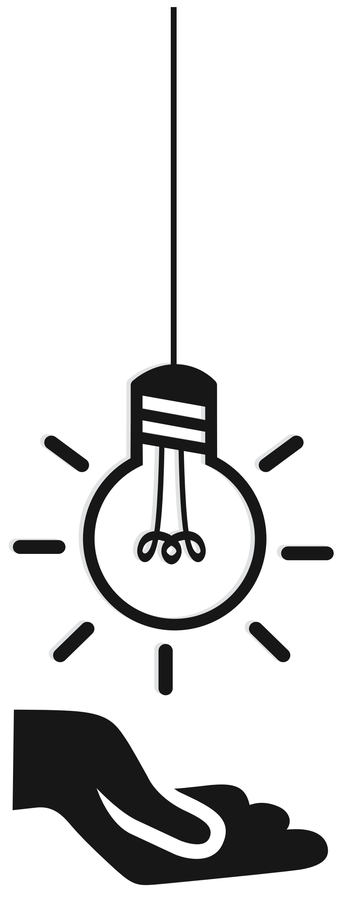
Due to the rapid growth of digital businesses and new emerging technology enablers, security should become a genuine concern for all organizations. Whether it is a cloud-based or internal owned by these organizations, any software interacts and integrates with other rapidly expanding environments of software components. They are indirectly providing the attackers an extra turf to play around. Simultaneously, Cloud service providers are taking care of the availability aspect to some extent but leaving all other areas open, which is merely a shared responsibility. Attackers are coming up with new patterns and payloads to exploit the unattended vulnerabilities overlooked by operations teams unintentionally. Given the context, Webservices/APIs have predominantly adopted across all these platforms providing ease of operating all the required functions or transactions. According to Gartner, "APIs have become foundational elements for the organization's digital transformation strategies. Hence securing APIs from attack and misuse is a concern for many security and risk management professionals". Entersoft research team has come up with a solution that exclusively focuses on proactively identifying those vulnerabilities in any environment.

The API security solution's Whitepaper intends to ensure that security professionals, developers, and organizations responsible for the development, maintenance, and operation of API endpoints have all required information to do their work correctly by absorbing the details. There are a few test cases that this white paper intended to address.

Besides, the Whitepaper intends to provide all information that is required.

Adoption of the product

APIs are a foundational element of organizations' digital transformation strategies. Hence, securing APIs from attack and misuse is a concern for many security and risk management professionals. API-specific testing, before and after development, builds a solid foundation for an overall API security strategy. The growing importance of APIs is leading to growing interest in and adoption of testing tools. Many tools are on offer, but functionality varies from one to another with specific products addressing different but sometimes overlapping use cases, which complicates evaluation and selection efforts.



Summary

This Whitepaper outlines the existing problems where leading automated REST API security scanners from the reputed organizations fail miserably in finding vulnerabilities. All the current automated scanners provide limited coverage because they fail in collecting necessary information from the customer at first glance. Most of the issues reported by these scanners are recognized as false positives because the scanner does not have the patterns which differentiate the false positives from actual vulnerabilities. Also, these companies fail to feed all the latest patterns into the scanners from time to time since it involves many research and development activities.

In this Whitepaper, we have discussed in-depth about our internally developed REST API scanner's functionalities and abilities, which helps organizations automatically detect OWASP Top 10 vulnerabilities with high accuracy and less manual intervention. Our tool can help organizations maintain optimal security by periodically scanning the REST APIs with ever-evolving latest attacks and patterns, which are the outcome of our research.

Introduction

Over the past few years, technology has revolutionized how applications interact and process information. Developers combine multiple APIs to create a new application. Nowadays, the interaction of users online is surrounded by APIs. With the simple words behind any web application or any kind of service, from simple blog posts through data collection and processing (e.g., from social media) to receive payments, REST APIs play a vital role.

REST became a trending technology in web architecture — it is now become a real web standard and replaced its predecessor – SOAP (Simple Object Access Protocol). REST APIs are changing web services' face by eliminating the heavy burden of integration for the developers. The way that everyone should think about REST is – it is simple and allows you to integrate faster.

The rapid growth and adaptation of the REST standard also increased the need for security. There were a lot of situations where a single REST API endpoint exposed the sensitive data of almost all the users. With our experience and research in testing the REST APIs for security, we collectively decided to design an automated scanner with the below features which other scanners have failed to provide

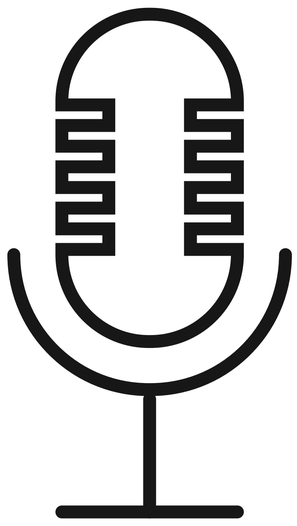
1. Customers usually prefer scanners that require fewer configuration efforts, but these scanners fail miserably in detecting complex vulnerabilities that require additional configuration details. We believe in "With ideal configuration comes the ideal outcome". Hence, we prefer to collect all the necessary information regarding the REST API endpoints and their respective constraints before scanning.

2. With the collected necessary information, our scanning engine will create payloads rather than fuzzing the REST API endpoint parameters with random values. The generated payloads are useful in identifying complex vulnerabilities, including input validations, injections, IDOR, Authorization, Authentication, and other Business Logic vulnerabilities.

Intended Audience

In the past few years, the IT world has seen rapid growth in digital services. Most of the technology companies are moving towards developing, implementing, and integrating the REST APIs into their infrastructure.

This Whitepaper helps the developers, architects, managers, and security professionals who are responsible for developing/securing/maintaining their REST APIs, used in processing the organization's business and critical customer data.



Key features of the product

1. Identifies OWASP REST API Security Risks
2. User-friendly API scanner
3. Easy API Conversions
4. Effective Role Mapping
5. Smart Payload Selection
6. Fastest Scanning Engine
7. Minimal False positives
8. Interactive Ticketing system
9. JIRA/Slack Integrations
10. Automated report

Key benefits of the features

1. Identifies OWASP REST API Security Risks

Discovers existing potential security vulnerabilities by automatically scanning an API for the below issues:

- Broken Object Level Authorization
- Broken Authentication
- Excessive Data Exposure
- Lack of Resources and Rate limiting
- Broken Function Level Authorization
- Mass Assignment
- Security Misconfiguration
- SQL Injection
- Improper Asset Management

2. User-friendly API Scanner

Identify potential vulnerabilities and secure your REST APIs by following the simple 5 step process in which our engine collects necessary details of the target APIs. The mandatory process includes providing the endpoint URI, roles, authorization headers (if any), body parameters (in case of POST, PUT, and DELETE) and verification of the target domain's ownership.

3. Easy API conversions

API's are developed for different purposes and are documented in multiple formats to adhere to the business requirement. Our scanning engine is a one-stop solution for getting these API's scanned and converting the documentations from one format to another with ease. The formats which we currently support includes:

- OpenAPI/Swagger
- RAML
- Postman Collection
- Insomnia Export Format
- HAR
- API Blueprint
- WSDL - W3C
- WADL - W3C
- Google Discovery
- I/O Docs - Mashery

4. Effective Role Mapping

To comprehensively identify the Authentication, Authorization, and Access Control misconfigurations on the target REST API, we collect mappings for the roles and respective endpoints to which each of the roles has access.

5. Smart Payload Selection

Instead of fuzzing the API endpoint parameters with random payloads, our engine will select the efficient payloads based on the context of the technology, vulnerability class, and parameter type; This helps identify the potential vulnerabilities with less effort.

6. Fastest Scanning Engine

Scanning a target REST API for the OWASP vulnerabilities with all the known and latest attack patterns is not a simple task to achieve in less time. Most of the existing commercial scanners take more time to complete a single scan. To overcome this problem, we have designed our scanner engine to scan every vulnerability with a unique algorithmic approach and the selected effective payloads.

7. Minimal False positives

Our engine thoroughly analyses the responses for every request, which identifies the vulnerabilities. For every vulnerability recognized, the engine feeds on multiple conditions that need to be satisfied before confirming the vulnerability. This process effectively reduces the occurrence of false positives.

8. Interactive Ticketing system

Customers can raise tickets to the support team for any of the issues encountered. The support team will be available round the clock to provide solutions for the tickets raised.

9. JIRA/Slack Integrations

Receive notifications for different actions on your tracking portals by simply integrating them into our portal. We currently support Jira and Slack integration to receive multiple notifications, including scan initiation,

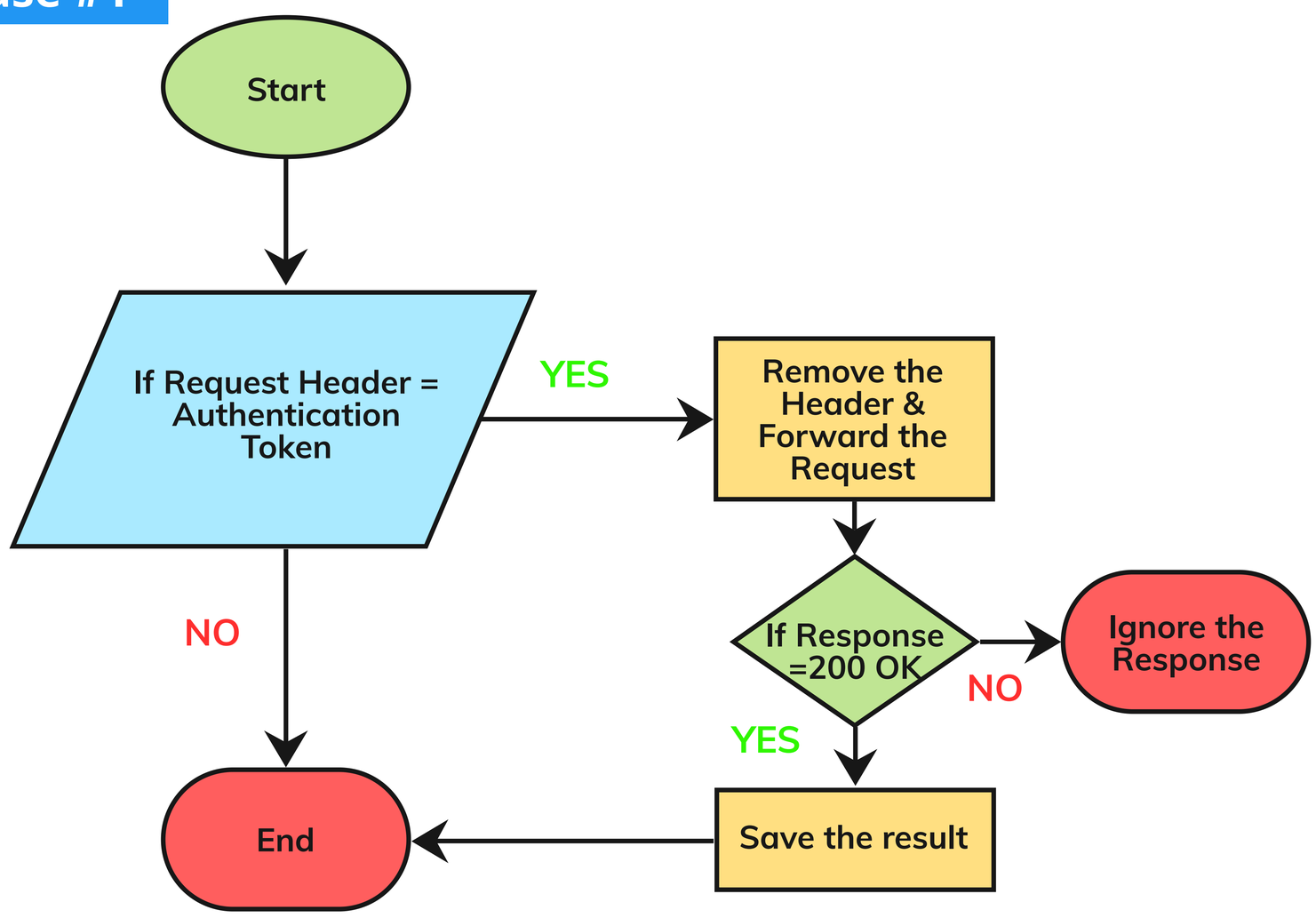
10. Automated report

You can export a PDF report containing all the identified vulnerabilities and detailed information, which includes description, location, parameters, remediations, and proof of concepts.

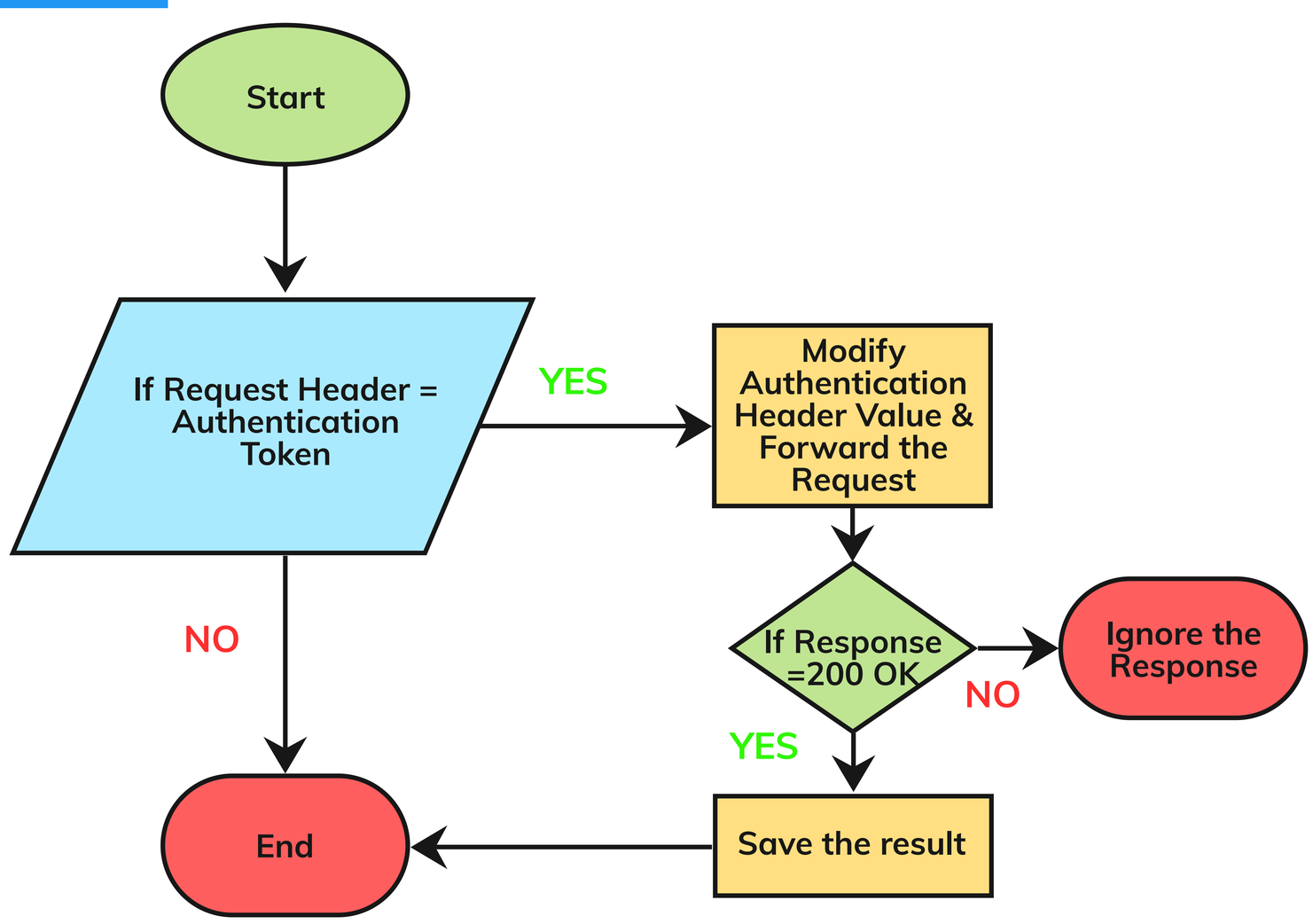
2. Broken Authentication

Authentication is the process of verifying the user's identity. Authentication is one of the five pillars of information assurance that resides with Integrity, Availability, Confidentiality, and Non-repudiation. Poorly implemented API authentication allows attackers to assume other user's identities. Lack of access token validation, Use of Weak, plain text, poorly hashed, default passwords are use cases of broken authentication.

Use Case #1



Use Case #2



How are we able to create a unique and reliable automated scanner?

With a cumulative experience of **15+** years, we have analyzed REST APIs related to various industries like Financial corporations, NBFCs, E-Commerce, Health, and SaaS. We have segregated all the vulnerability patterns identified, and the possible patterns which we have anticipated to be vulnerable. We have built an intelligent automated scanner to which we have fed all these patterns, making it one of its kind in terms of reliability and accuracy.

We have discussed a few vulnerability classes and the respective patterns below.

1. Broken Object Level Authorization

2. Broken Authentication

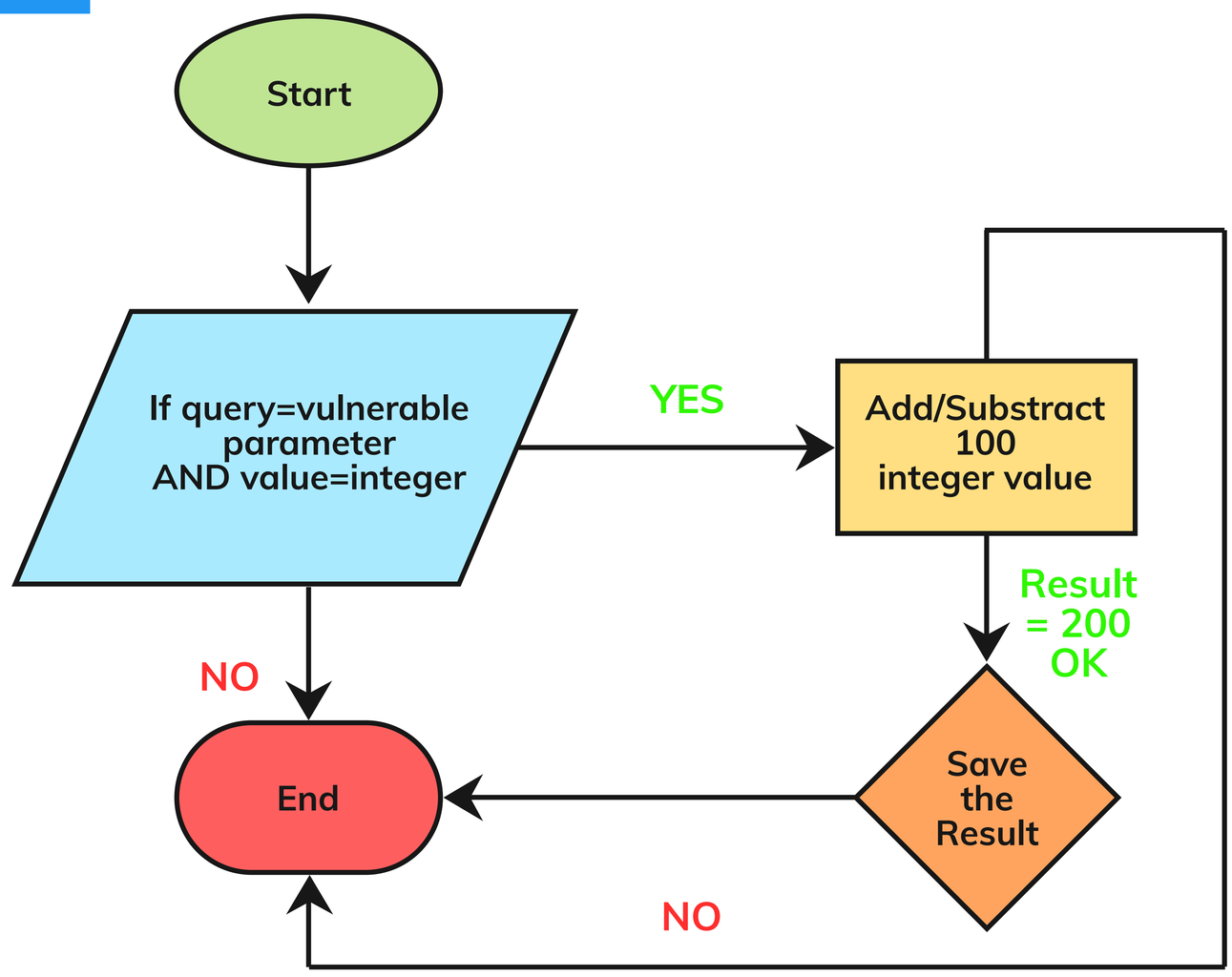
3. Excessive Data Exposure

4. Lack of Resources & Rate Limiting

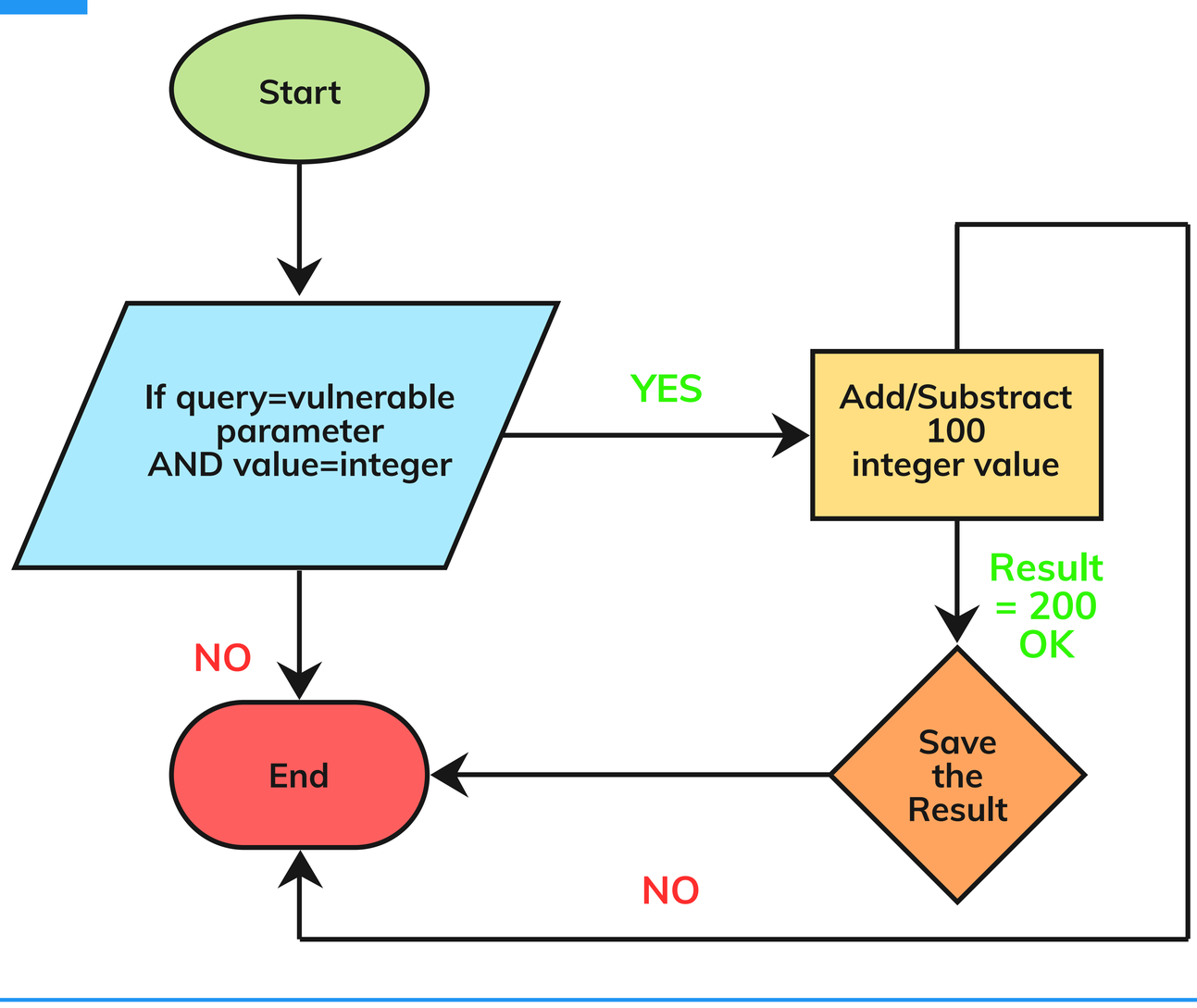
1. Broken Object Level Authorization

Broken Object Level Authorization is also known as Insecure Direct Object Reference. It occurs when an application provides direct access to the objects based on the user-supplied input. It allows attackers to bypass authorization and access resources directly by modifying a parameter used to direct an object.

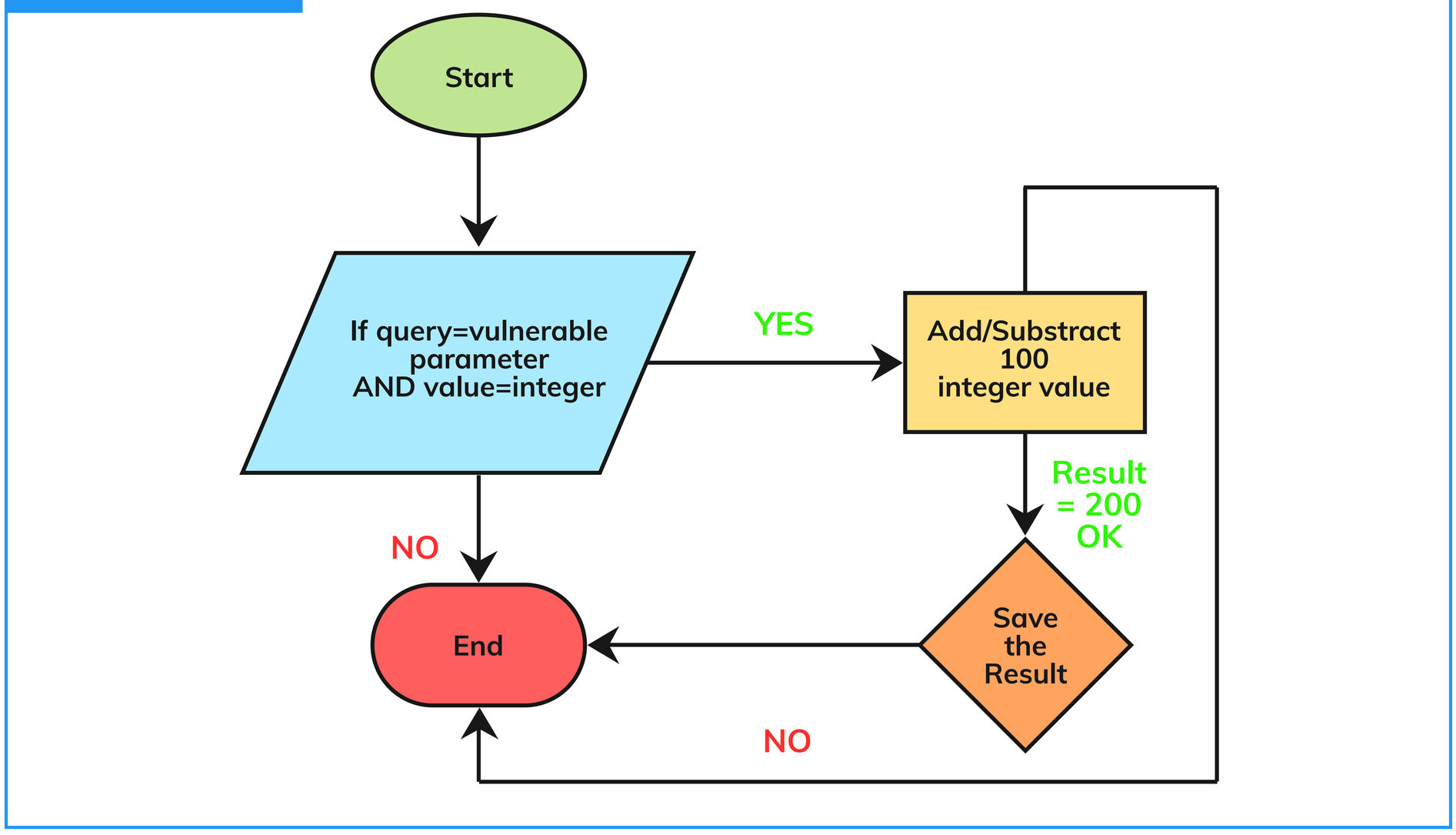
Use Case #1



Use Case #2



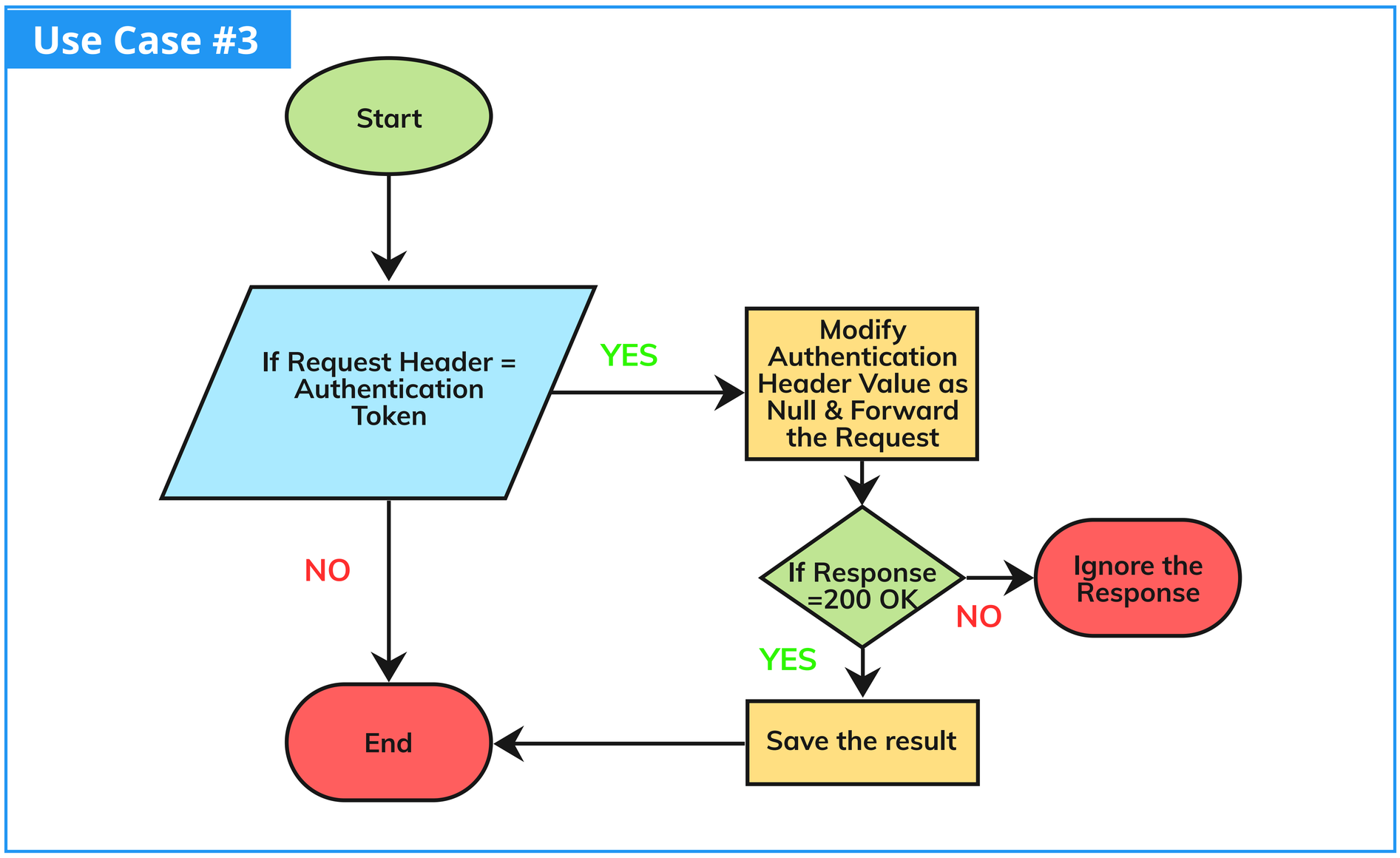
Use Case #3



Expected Result:

Display Broken Object Level Authorization is vulnerable if the response contains 200 status code.

Use Case #3



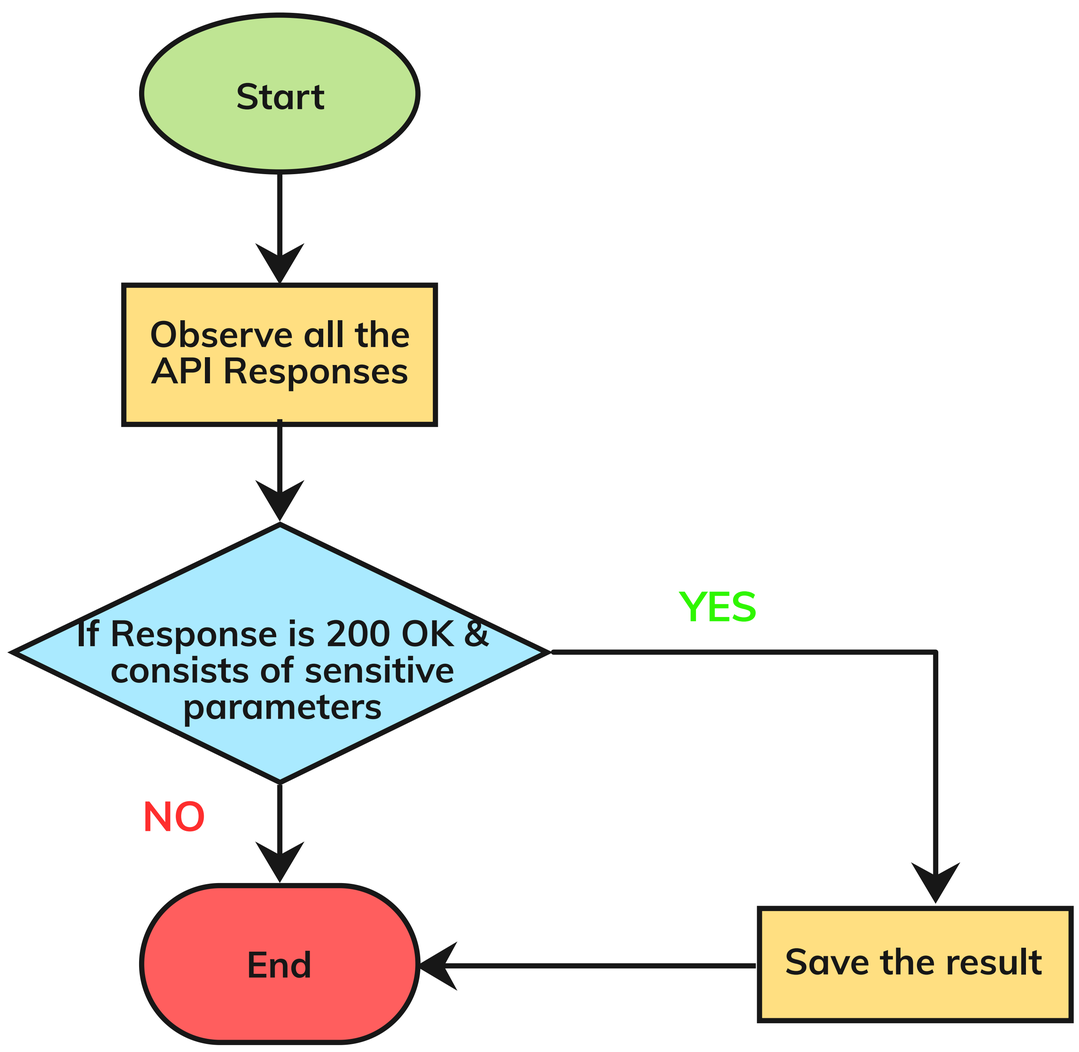
Expected Result:

Display Broken Authentication is vulnerable if the response contains 200 status code.

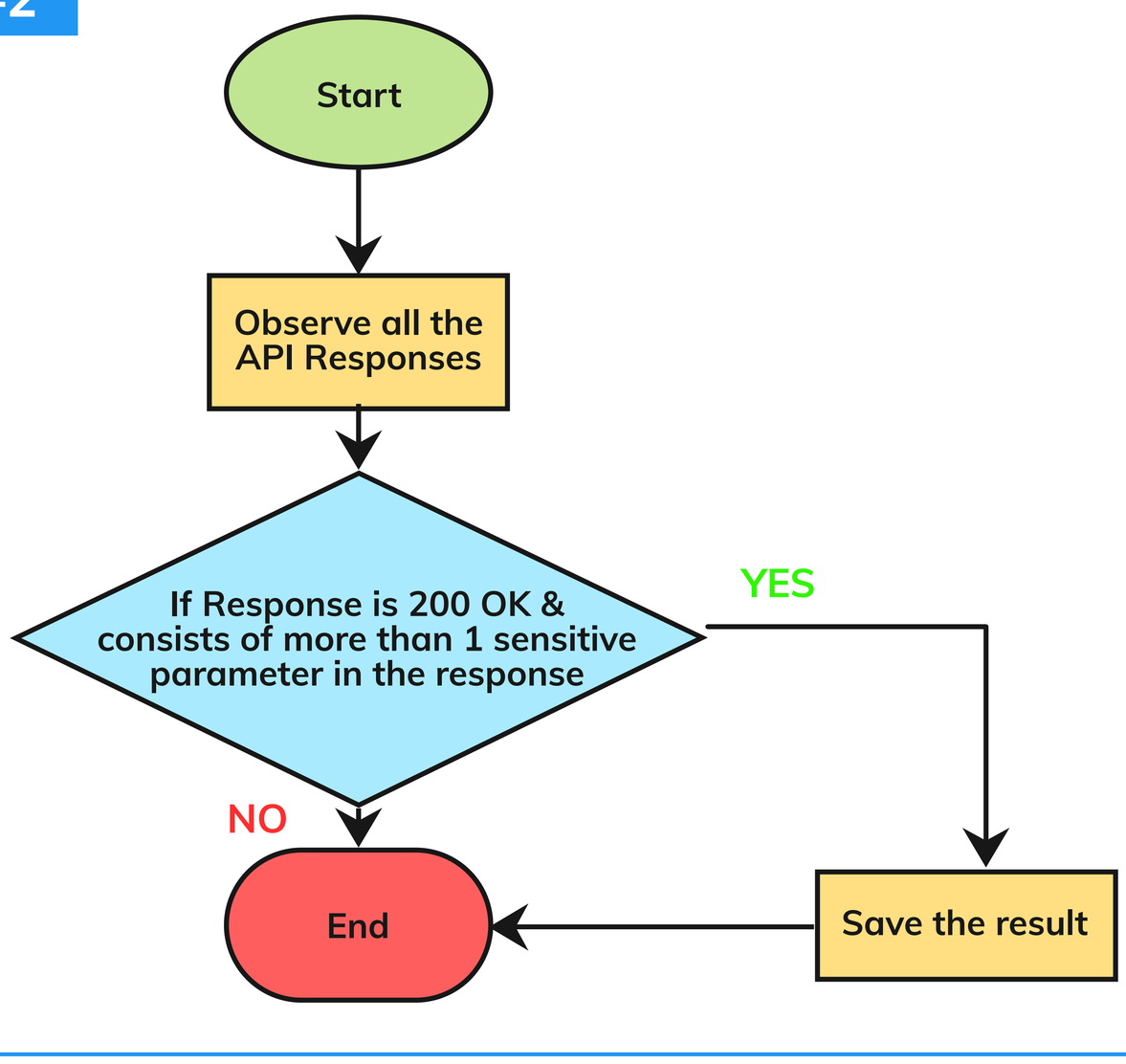
3. Excessive Data Exposure

Excessive Data Exposure occurs when an application or program, like a smartphone app or a browser, does not adequately protect information such a password, payment info, or personal data. For example: if you want to get some specific details on your profile, you will make a call(/profile/{id}) to the server which pulls the data from the database and responds to you with all the details available rather than giving the specific data you requested.

Use Case #1



Use Case #2

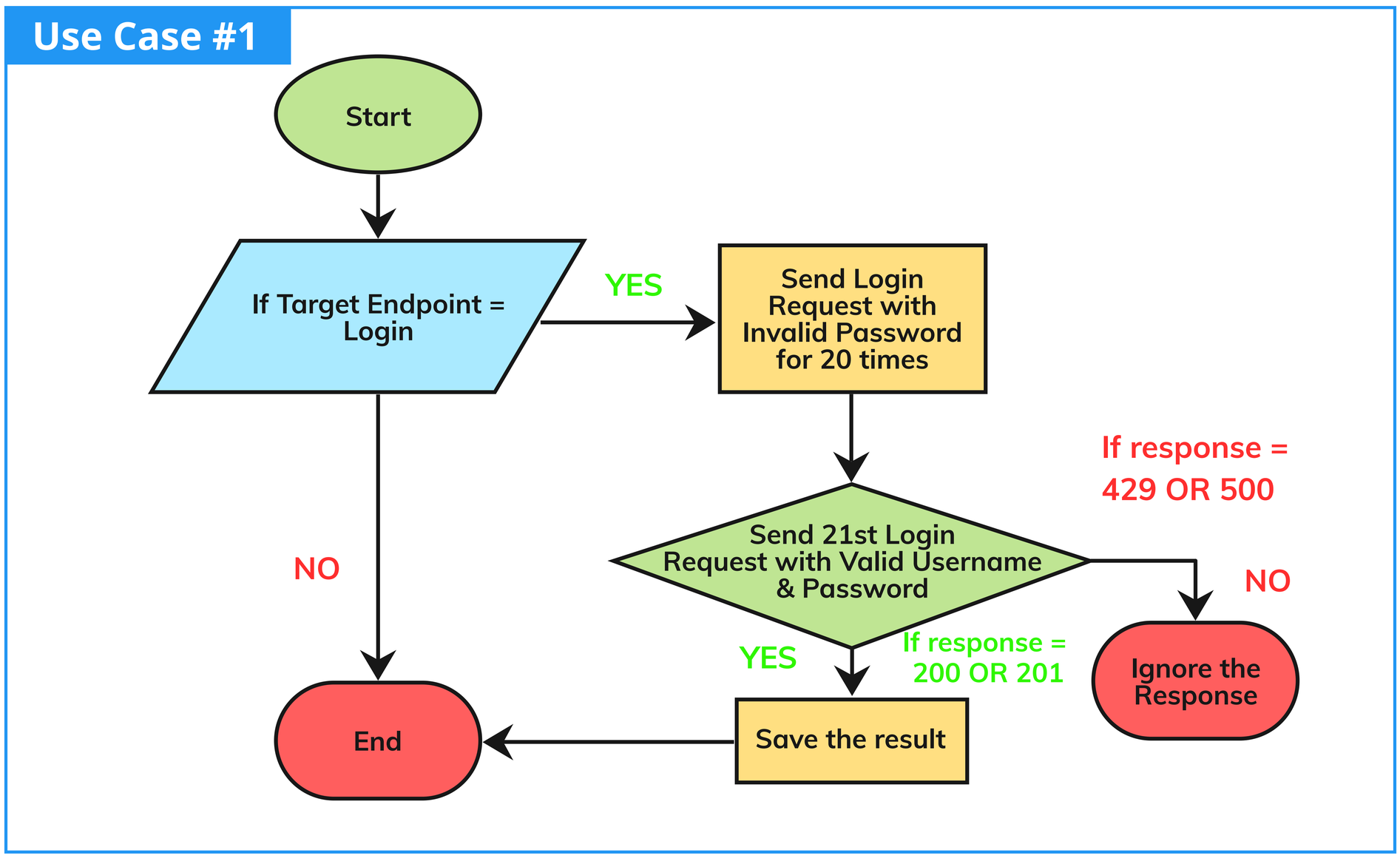


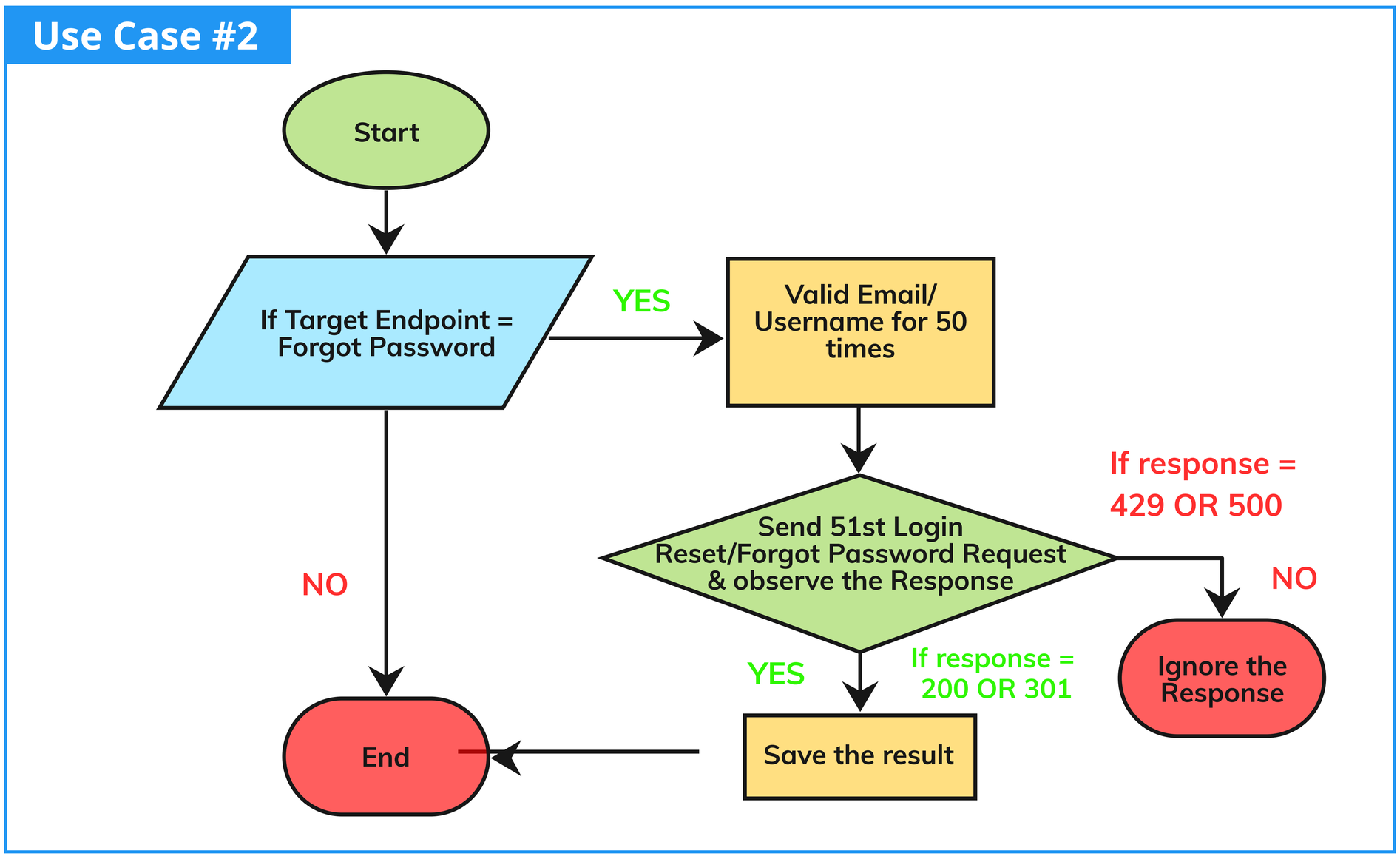
Expected Result:

If any endpoint response contains 200 Ok and any sensitive parameter matched in response (or) response contains the same sensitive parameter more than once, report as Excessive Data Exposure is vulnerable.

4. Lack of Resources & Rate Limiting

Unlimited access to an API can have severe consequences like Denial of Service (DoS) and authentication flaws like brute force attacks. Rate limiting prevents malicious code from abusing legitimate / illegitimate access to your API.





Expected Result:

Display Lack of Resource and Rate limiting is vulnerable if the 21st response in use case 1 and 51st response in Use cases 2 and 3 is 200 status code.

References

https://apisecurity.io/
https://www.owasp.org/index.php/OWASP_API_Security_Project#tab=Main

Action at the end

Demo? Request at: https://apicritique.com/

Trail usage? Contact us at: https://www.entersoftsecurity.com/contactus

Queries? Call us at +91 - 888 546 2220

Queries? Call us at info@entersoftsecurity.com

API CRITIQUE

Powered By

ENTERSOFT SECURITY

WWW.APICRITIQUE.COM

INFO@ENTERSOFTSECURITY.COM

INDIA, AUSTRALIA, U.S., HONG KONG, ISRAEL