wib

1
Broken object-level authorization (BOLA)
2
Broken User Authentication
3
Excessive d
exposure
4
Lack of Re
and Rate-
5
Broken Functio
Level Authorizatio
6
Mass Assignment
ecurity
isconfiguration
ection
Improper assets management
10
Insufficent logging & monitoring

OWASP API 8:

Injection

OWASP top 10

Injection is at the heart of several vulnerabilities, including SQL injection, OS command injection, and XML injection. As such, it accounts for a significant portion of the vulnerabilities discovered in real-world apps and APIs.

When an application cannot discriminate between untrusted user data and code, it becomes vulnerable. In injection attacks, hackers send malicious data as input to the API. The program's interpreter will confuse the user input as part of a command or query if the application does not correctly handle untrusted user data before putting it into a command or query. In this situation, attackers can provide data to a program that alters the meaning of the instructions it issues.

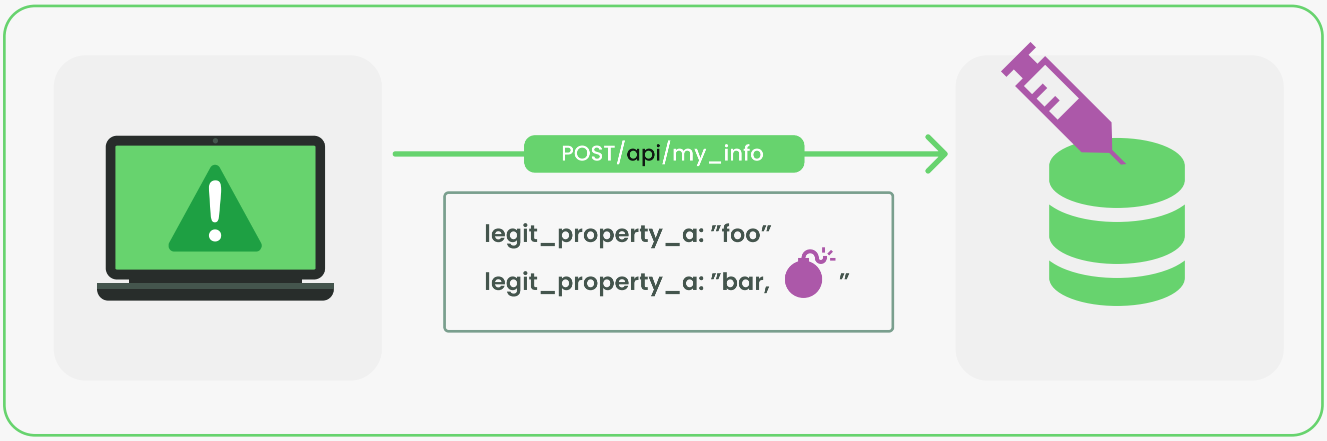
How Hackers Exploit It

By examining the web browser network traffic, an attacker might notice an API request like this, which starts the password recovery process:

POST /api/accounts/recovery {“username”: “matthew@yahoo.com”}
The attacker sends a new payload to the server:
POST /api/accounts/recovery {“username”: “matthew@yahoo.com’;WAITFOR DELAY ‘0:0:7’ — ”}

The attackers see that the response time has increased by -7 seconds, indicating that the API is vulnerable to SQL injection.

They may now concatenate phrases like SELECT * FROM users WHERE role='admin' and continue collecting valuable data from the destination.



Why You Should Care

Attackers exploit these injection vulnerabilities by sending malicious data to an API that is in turn processed by an interpreter or parsed by the application server and passed to some integrated service, such as a database management system (DBMS) or a database-as-a-service (DBaaS) in the case of SQL injection. They trick the interpreter or parser into executing unintended commands since they either lack the filtering directly or expect it to arrive filtered by other server-side code. Injection can lead to many impacts, including information disclosure, data loss, denial of service (DoS), or complete host takeover. In many cases, successful injection attacks expose large sets of unauthorized sensitive data. Attackers could also create new functionality, perform remote code execution, or bypass authentication and authorization mechanisms altogether.

Traditional Tools Will Not Protect You

Because WAFs and some API Gateways may employ signatures to pattern-match and detect known injection types, protecting against injection attacks is a frequent feature. Signatures are frequently created for off-the-shelf online and application software packages. However, these tools can miss new attack types if they lack the most recent signature changes. As threats become more complicated and sophisticated, these tools miss attacks that target the logic of APIs.

Another factor to consider is that WAFs generally protect web traffic, with API traffic being a subset and peripheral emphasis. This approach could result in only a positive security model to enforce traffic against an API schema or specified HTTP traffic patterns. They may not apply injection protections and other rulesets to API traffic.

How to Combat Injection Threats

You must do several things to protect your APIs from injection threats:

1. Validate, filter, and sanitize any data given by clients, as well as data from other interconnected systems
2. Validate data using a single, dependable, and regularly maintained library
3. Choose a secure API with a parameterized interface
4. Escape out special characters using the target interpreter's unique syntax
5. For all string arguments, define data types and strict patterns
6. Validate incoming data using enough filters to ensure that only valid values for each input parameter are allowed
7. Keep the number of records returned to a limit to avoid mass disclosure in the event of an injection

How Wib Can Help You

Wib is a full-lifecycle API security platform, defending against API security threats. The platform ensures comprehensive protection across the entire API software lifecycle from development to testing to production.

Wib utilizes state-of-the-art proprietary AI and ML to analyze millions of requests in real-time. It provides complete visibility of existing APIs, with actionable insights and comprehensive protection across the entire lifecycle. It learns your APIs inside and out and can provide complete visibility of your existing APIs, analyze their integrity, identify vulnerabilities, and detect attacks in real-time.

If you are building or using APIs in your applications (and you probably are), sign up for a demo to see Wib in action and learn more.