DVAPI Vulnerability Assessment Report

# API4:2023 – Unrestricted Resource Consumption

## Objective

To assess and demonstrate the exploitation of the Unrestricted Resource Consumption vulnerability in DVAPI (API4:2023) using Postman, resulting in the retrieval of the fourth flag.

## Overview of DVAPI

DVAPI (Damn Vulnerable API) is an intentionally insecure API developed for educational and training purposes. It replicates common API security flaws based on the OWASP API Security Top 10, providing a realistic environment for practicing vulnerability assessment and penetration testing.

## Vulnerability Description

Unrestricted Resource Consumption (API4:2023) refers to scenarios where APIs fail to enforce limits on the use of system resources such as memory, CPU, bandwidth, or storage. This can lead to degraded performance, denial-of-service (DoS), and increased operational costs.

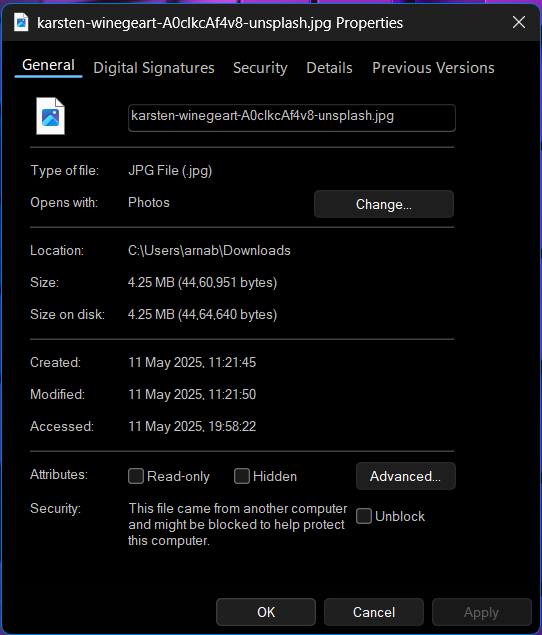
## Tools Used

**• DVAPI:** Local instance at http://127.0.0.1:8000  
**• Postman:** Used for crafting and sending API requests  
**• Online Image Resizer:** To increase image file size for testing upload limits

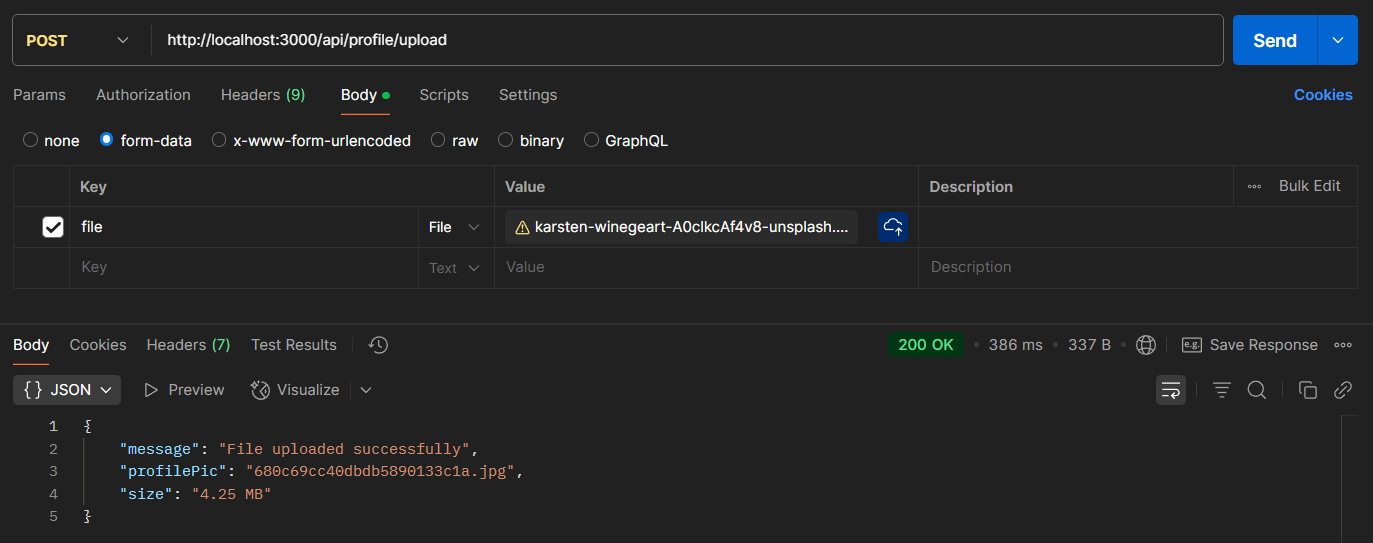
## Steps to Reproduce

The following steps demonstrate the exploitation process, as observed using Postman:

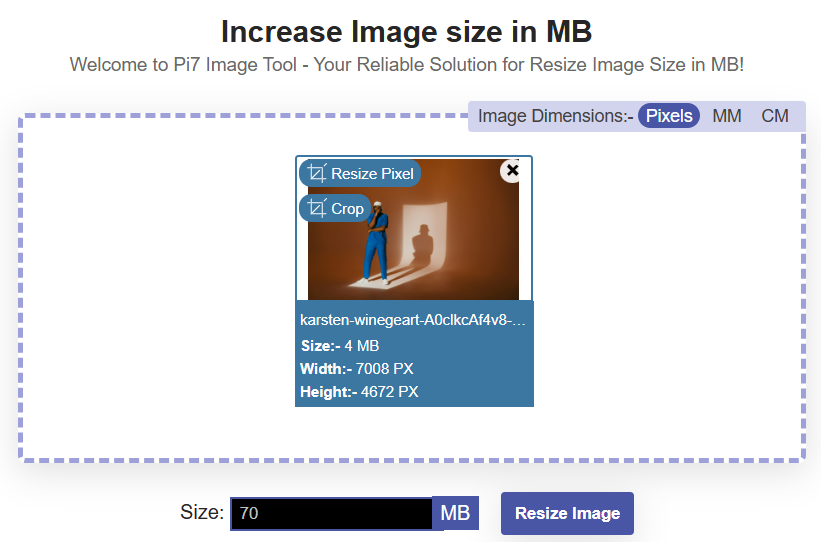
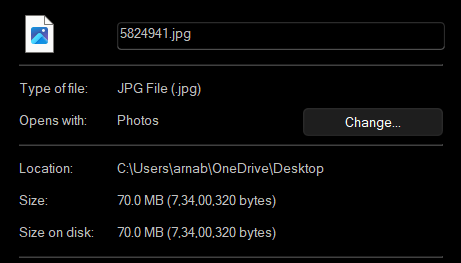
Sending an image above 2 mb to check if there is any restriction on the size.



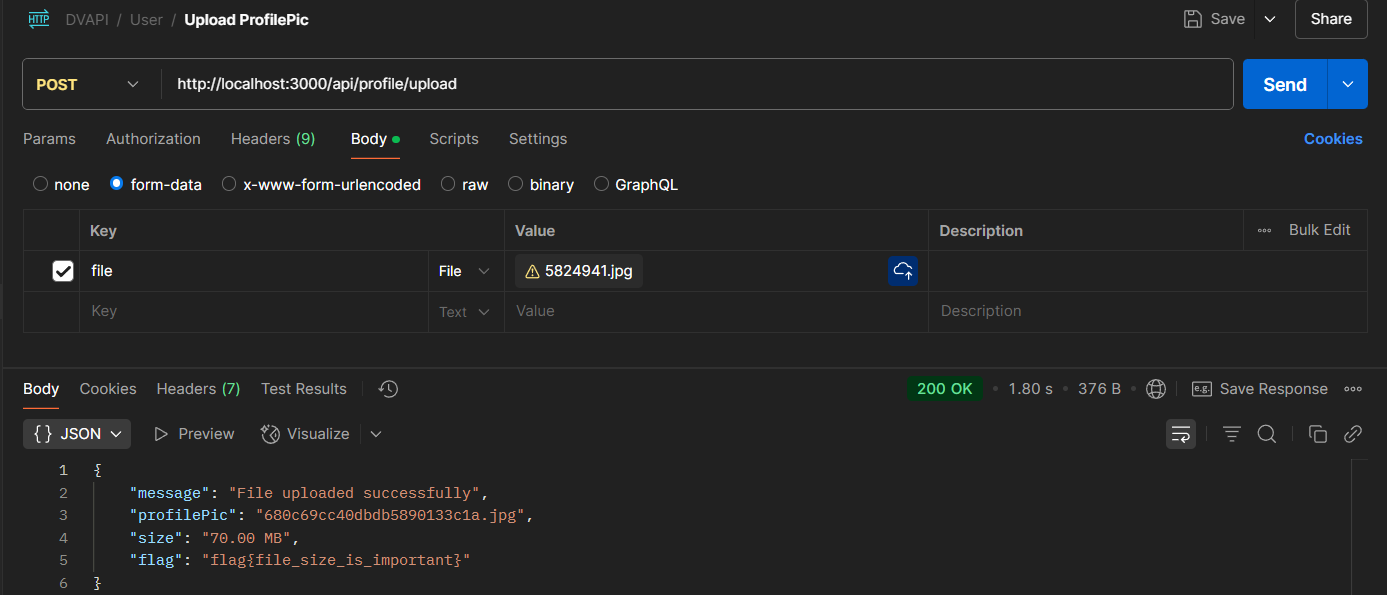
As we can see the image size is 4 mb



And when we upload it through postman, we don’t get see any restriction in file size. Rather it gets uploaded successfully implying that there is no limit on file size.

We use online image size increase tool to increase the size of this image to 70 mb



Now if we send this 50 mb size image we get the flag.

flag{file\_size\_is\_important}

## Impact

The lack of file size validation allows attackers to upload large files and consume server resources excessively. This could impact availability and performance, and could also lead to additional infrastructure costs.

## Mitigation Strategy

• Enforce file upload size limits at the API level.  
• Validate file types and content before processing uploads.  
• Implement rate limiting and user-level quotas.  
• Monitor and alert on unusual traffic patterns or large payloads.