# YUICHIRO ADVANCED SECURITY AUDIT

*Strict Semantic Analysis & Remediation Report*

==========

## 1. SCAN SUMMARY

Project Target : C:\jacoba-coffe2025

Timestamp : 2025-12-28 19:08:34

Total Findings : 33 issues detected

---

## 2. DETAILED FINDINGS AND ANALYSIS

### FINDING #1: [HIGH] Sensitive Data Leakage (High)

Location : .env (Line 3)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

### FINDING #2: [HIGH] Sensitive Data Leakage (High)

Location : .env (Line 27)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

### FINDING #3: [HIGH] Sensitive Data Leakage (High)

Location : .env (Line 35)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

### FINDING #4: [HIGH] Sensitive Data Leakage (High)

Location : .env (Line 72)

Description : Credentials or Debug mode exposed in config.

==========

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

## FINDING #5: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\EventBookingController.php (Line 22)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #6: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\EventBookingController.php (Line 34)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #7: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\EventBookingController.php (Line 45)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #8: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\OrderController.php (Line 19)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

# YUICHIRO ADVANCED SECURITY AUDIT

*Strict Semantic Analysis & Remediation Report*

==========

### FINDING #9: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\OrderController.php (Line 95)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #10: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\OrderController.php (Line 100)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #11: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\OrderController.php (Line 123)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #12: [CRITICAL] SQL Injection (Critical)

Location : app\Http\Controllers\ReportController.php (Line 32)

Description : Raw query detected. Potential SQLi if variables are not bound.

**REMEDIATION STEPS:**

1. Never concatenate user input directly into queries.
2. Use Eloquent methods where possible.
3. If raw SQL is needed, use '?' placeholders.

==========

### FINDING #13: [CRITICAL] SQL Injection (Critical)

Location : app\Http\Controllers\ReportController.php (Line 32)

Description : Raw query detected. Potential SQLi if variables are not bound.

**REMEDIATION STEPS:**

1. Never concatenate user input directly into queries.

2. Use Eloquent methods where possible.
3. If raw SQL is needed, use '?' placeholders.

==========

### FINDING #14: [HIGH] Broken Access Control (High)

Location : app\Http\Controllers\ReportController.php (Line 14)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #15: [HIGH] Broken Access Control (High)

Location : app\Models\Event.php (Line 11)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #16: [HIGH] Broken Access Control (High)

Location : app\Models\Event.php (Line 16)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #17: [HIGH] Broken Access Control (High)

Location : app\Models\EventPackage.php (Line 28)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #18: [HIGH] Broken Access Control (High)

==========

Location : app\Models\EventTable.php (Line 24)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #19: [HIGH] Broken Access Control (High)

Location : app\Models\EventTable.php (Line 32)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #20: [HIGH] Broken Access Control (High)

Location : app\Models\EventTable.php (Line 41)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #21: [HIGH] Broken Access Control (High)

Location : app\Models\EventTable.php (Line 54)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

## FINDING #22: [HIGH] Broken Access Control (High)

Location : app\Models\EventTableType.php (Line 17)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

# YUICHIRO ADVANCED SECURITY AUDIT

*Strict Semantic Analysis & Remediation Report*

==========

### FINDING #23: [HIGH] Broken Access Control (High)

Location : app\Models\EventTableType.php (Line 22)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #24: [HIGH] Broken Access Control (High)

Location : app\Models\Order.php (Line 23)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #25: [HIGH] Broken Access Control (High)

Location : app\Models\OrderItem.php (Line 23)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #26: [HIGH] Broken Access Control (High)

Location : app\Models\OrderItem.php (Line 28)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #27: [HIGH] Broken Access Control (High)

Location : app\Models\OrderItem.php (Line 33)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.

==========

2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #28: [HIGH] Broken Access Control (High)

Location : app\Models\OrderItem.php (Line 38)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #29: [HIGH] Broken Access Control (High)

Location : app\Services\ImageKitService.php (Line 15)

Description : Semantic Check: Method lacks $this->authorize() call.

**REMEDIATION STEPS:**

1. Implement Laravel Policies.
2. Call $this->authorize('update', $model) at the start of controller methods.

==========

### FINDING #30: [HIGH] Sensitive Data Leakage (High)

Location : config\cache.php (Line 84)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

### FINDING #31: [HIGH] Sensitive Data Leakage (High)

Location : config\filesystems.php (Line 53)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

### FINDING #32: [HIGH] Sensitive Data Leakage (High)

Location : config\queue.php (Line 59)

==========

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

## FINDING #33: [HIGH] Sensitive Data Leakage (High)

Location : config\services.php (Line 32)

Description : Credentials or Debug mode exposed in config.

**REMEDIATION STEPS:**

1. Ensure .env is in .gitignore.
2. Use Laravel Vault or Secret Manager for production.

==========

## 3. SECURITY RESEARCHER CHEATSHEET

**- XSS Prevention**

Gunakan {{ $data }} sebagai ganti {!! $data !!}. Gunakan Blade components untuk auto-escaping.

**- CSRF Protection**

Selalu gunakan @csrf di dalam form. Jangan matikan middleware VerifyCsrfToken secara global.

**- Session Security**

Atur SESSION_SECURE_COOKIE=true dan SESSION_HTTP_ONLY=true di file .env.

**- Mass Assignment**

Lebih baik gunakan $fillable daripada $guarded. Hindari Request::all() pada method create.

**- API Security**

Gunakan Sanctum atau Passport. Jangan ekspos ID internal; gunakan UUID untuk resource publik.