



# kelpompok 2

Finding Details: Faraj Muhiddin

CODE AUDITING

The image shows a terminal window with a dark background and light-colored text. It displays a C++ program for multiplying two matrices. The code includes user input for matrix dimensions and element entry. A portion of the code is highlighted in yellow, indicating it is being audited. The audit findings are listed in red text, pointing out potential security issues such as buffer overflows and lack of input validation.

```
/* Enter rows and columns for first matrix.
 * Enter rows and columns for second matrix.
 */
cout << "Enter elements of first matrix,
        one row at a time: ";
for(i = 0; i < m1; i++) {
    for(j = 0; j < n1; j++) {
        cout << "Enter element a[" << i << "][" << j << "]: ";
        cin >> a[i][j];
    }
}
cout << "Enter elements of matrix 1: " << endl;
for(i = 0; i < m1; i++) {
    for(j = 0; j < n1; j++) {
        cout << a[i][j] << " ";
    }
    cout << endl;
}

/* Enter rows and columns for second matrix.
 * Enter rows and columns for first matrix.
 */
cout << "Enter elements of matrix 2: " << endl;
for(i = 0; i < m2; i++) {
    for(j = 0; j < n2; j++) {
        cout << "Enter element b[" << i << "][" << j << "]: ";
        cin >> b[i][j];
    }
}
cout << "Enter elements of matrix 2: " << endl;
for(i = 0; i < m2; i++) {
    for(j = 0; j < n2; j++) {
        cout << b[i][j] << " ";
    }
    cout << endl;
}
```

# VULN-001

## BROKEN ACCESS CONTROL (IDOR) - CART QUANTITY MANIPULATION



Common Vulnerability Scoring System  
(CVSS-SIG)

- Calculator
- Specification Document
- User Guide
- Examples
- Frequently Asked Questions
- CVSS v4.0 Documentation & Resources
- CVSS v3.1 Archive
- CVSS v3.0 Archive
- CVSS v2 Archive
- CVSS v1 Archive
- JSON & XML Data Representations
- CVSS On-Line Training Course
- Identity & logo usage

## Common Vulnerability Scoring System Version 4.0 Calculator

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:N/VI:H/VA:N/SC:N/SI:N/SA:N

Reset

CVSS v4.0 Score: 7.1 / High

Hover over metric names and metric values for a summary of the information in the official CVSS v4.0 Specification Document. The Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, a set of Frequently Asked Questions (FAQ), and both JSON and XML Data Representations for all versions of CVSS.

### Base Metrics ?

#### Exploitability Metrics

Attack Vector (AV): Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC): Low (L) High (H)

Attack Requirements (AT): None (N) Present (P)

# Broken Access Control (IDOR) • Cart Quantity Manipulation

- Executive Summary – Attacker dapat memodifikasi quantity item keranjang pada user lain dengan mengirim patch request ke endpoint tanpa verifikasi dari pemilik akun.
- CVSS Score – 7.1
- Severity – High
- CVSS String – CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:N/VI:H/VA:N/SC:N/SI:N/SA:N

- User Korban sebelum diubah quantity

The screenshot shows a web browser window titled "Checkout - Beliin". The URL is <http://192.168.100.34:8000/checkout>. The page displays a "Keranjang Belanja" (Shopping Cart) with the following items:

Produk	Jumlah	Total
tv elektronik tv Rp 1.000.000	1	Rp 1.000.000
kalung Aksesoris kalung Rp 10	1	Rp 10

**Ringkasan Pesanan**

Subtotal	Rp 1.000.010
Pajak (0%)	Rp 0
<b>Total</b>	<b>Rp 1.000.010</b>

A large blue button labeled "Bayar Sekarang" is prominently displayed.

At the bottom of the browser window, the developer tools Network tab is selected, showing network activity details.

Instructions in the developer tools:

- Perform a request or Reload the page to see detailed information about network activity.
- Click on the button to start performance analysis.

- **User Korban setelah diubah quantity**

The screenshot shows a web browser window with the title "Checkout - Beliin". The address bar indicates the URL is "http://192.168.100.34:8000/checkout". The main content is a "Keranjang Belanja" (Shopping Cart) page. On the left, there's a table for "Produk" (Products) showing two items: "tv" (elektronik) and "kalung" (Aksesoris). The "tv" item has a quantity of 2, and the "kalung" item has a quantity of 1. The total amount for the "tv" item is "Rp 2.000.000" and for the "kalung" item is "Rp 10". To the right, a "Ringkasan Pesanan" (Order Summary) box shows a "Subtotal" of "Rp 2.000.010" and a "Pajak (0%)" of "Rp 0". The "Total" is highlighted with a red box and is "Rp 2.000.010". Below it is a blue button labeled "Bayar Sekarang" (Pay Now).

At the bottom, a developer tools Network tab is visible. It shows a single request: a PATCH method to the URL "192.168.100.34:8000/cart/update/17". The Headers section is highlighted with a red box and shows the following details:

- Scheme: http
- Host: 192.168.100.34:8000
- Filename: /cart/update/17

The Response section shows a 200 OK status with the following details:

- Status: 200 OK
- Version: HTTP/1.1
- Transferred: 1,18 kB (99 B size)
- Referrer Policy: strict-origin-when-cross-origin
- Resource Priority: Hintlight

- User penyerang login dan melihat id dari penyerang

The screenshot shows a Firefox browser window with a Kali Linux desktop environment in the background. The title bar says "Checkout - Beliin". The address bar shows "Not Secure http://192.168.100.34:8000/checkout". The page content is a "Keranjang Belanja" (Shopping Cart) with a red box around it. It lists a single item: "tv" under "Produk" and "Rp 1.000.000" under "Jumlah". To the right is a "Ringkasan Pesanan" (Order Summary) box with a red box around it. It shows "Subtotal Rp 1.000.000", "Pajak (0%) Rp 0", and a total of "Rp 1.000.000" with a "Bayar Sekarang" (Pay Now) button. Below the browser is the Kali Linux desktop with a Network tab open in the developer tools. A red box highlights the XHR section of the Network tab, which shows a PATCH request to "http://192.168.100.34:8000/cart/update/19" with the following headers:

```
PATCH
Scheme: http
Host: 192.168.100.34:8000
Filename: /cart/update/19
Address: 192.168.100.34:8000
```

The response status is 200 OK.

- **penyerang mencoba attack melalui terminal kali dengan merubah idnya**

```
tes [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Aksi Sunting Lihat Bantuan

(kali㉿kali)-[~]
$ curl 'http://192.168.100.34:8000/cart/update/17' \
-X PATCH \
-H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0' \
-H 'Accept: */*' \
-H 'Accept-Language: en-US,en;q=0.5' \
-H 'Accept-Encoding: gzip, deflate' \
-H 'Referer: http://192.168.100.34:8000/checkout' \
-H 'Content-Type: application/json' \
-H 'X-CSRF-TOKEN: 5iKBMKLIANMAFFxPyvIegONHJ0FWSJXRN8ujdAbe' \
-H 'Origin: http://192.168.100.34:8000' \
-H 'Connection: keep-alive' \
-H 'Cookie: XSRF-TOKEN=eyJpdiI6ImNvdmJ1MUDWRzlaemV0czNjM1E0UFE9PSIsInZhbHVlIjoiazRjU0t0emJ4K2RYZG1kN29xTddGTnVPWGLYRjF6R25haG5lbGZ1SUzsM0I4NUtZOFI3aGo1dXVyQ1NVZ3VMbHIzbXgwRTEzzVI1YkRVTjZWV2hTSzQrdWFkalFMV2cvRHBIVGFGY0J5cE9hOHFZbzFCRjZ1cnhTY0JnTG1GcGgiLCJtYWMiOii0NzliMtc0YWYzNTkwYmE1MzhmOGMxNWNhZDRjN2I4ZTEwY2RiNjMyMTg40GQxMWMyNWNgkYTFjMWE1MmQ0MDRmIiwidGFnIjoiIn0%3D; laravel-session=eyJpdiI6IkUxcExJbHQ0bmFIdTRIdDlwTDJubWc9PSIsInZhbHVlIjoiR1ovY2ltQ1A4NEJ0bkZICzdiRjBDYUlGL1BrZE9wcmxsckh4UGh0emdCVGxHbnpgYTFqchM2NFJWMWtWQkE1MXVPQ0hDSHRCRnhPcFFnbjcyLzB3cW9uYWRUdVhGNzZBRTJRNzRPL09UTWpGTDlQQm5WS1B6Ti9WN0hsOUJrQnQiLCJtYWMiOijjMjM1NGZkZWZjY1zje1MGNln2YwMDzlZjk4MTEXZTQwZDk30Thl0WM5YWM5YWNhYTRmOWYxOWJkOTJjMWM5IiwidGFnIjoiIn0%3D' \
-H 'Priority: u=0' \
--data-raw '{"change": "-1"}'
{"status": "success", "new_quantity": 1, "new_line_total": "Rp 1.000.000", "new_subtotal": "Rp 1.000.010"}

(kali㉿kali)-[~]
```

- **Hasil user korban setelah di serang ini diserangnya untuk mengurangi cart**

The screenshot shows a web browser window with a shopping cart page titled 'Keranjang Belanja'. The cart contains two items: a 'tv' priced at Rp 1.000.000 and a 'kalung' priced at Rp 10. A summary box on the right shows a subtotal of Rp 1.000.010. Below the cart is a large blue button labeled 'Bayar Sekarang'.

At the bottom of the browser window, the developer tools Network tab is open, showing a recent PATCH request to 'http://192.168.100.34:8000/cart/update/17'. The request details show the same JSON payload as the terminal command above, indicating the modification made to the cart.

Produk	Jumlah	Harga
tv	1	Rp 1.000.000
kalung	1	Rp 10

**Ringkasan Pesanan**

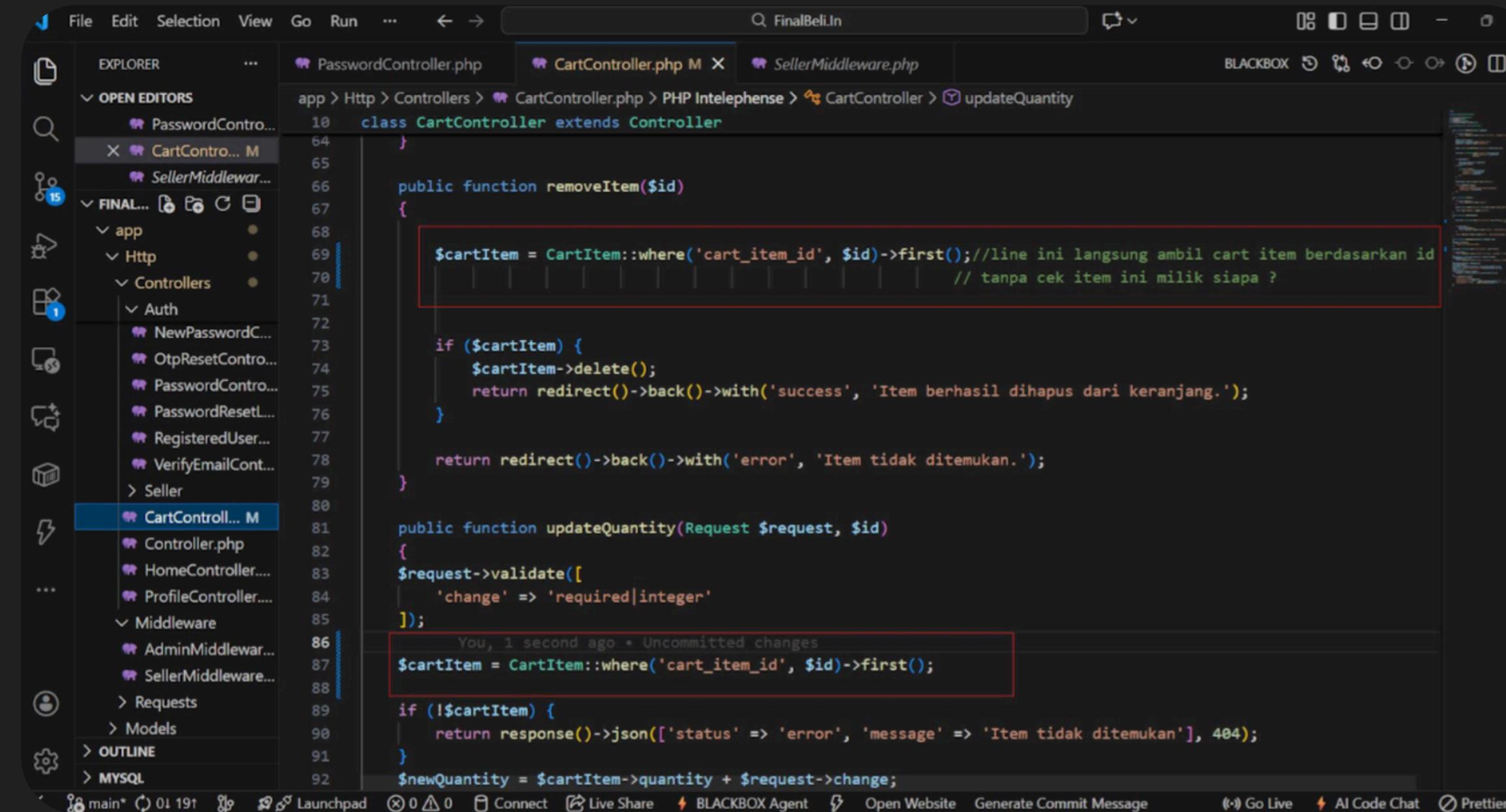
Subtotal	Rp 1.000.010
Pajak (0%)	Rp 0
<b>Total</b>	<b>Rp 1.000.010</b>

**Bayar Sekarang**

**Network Tab Details:**

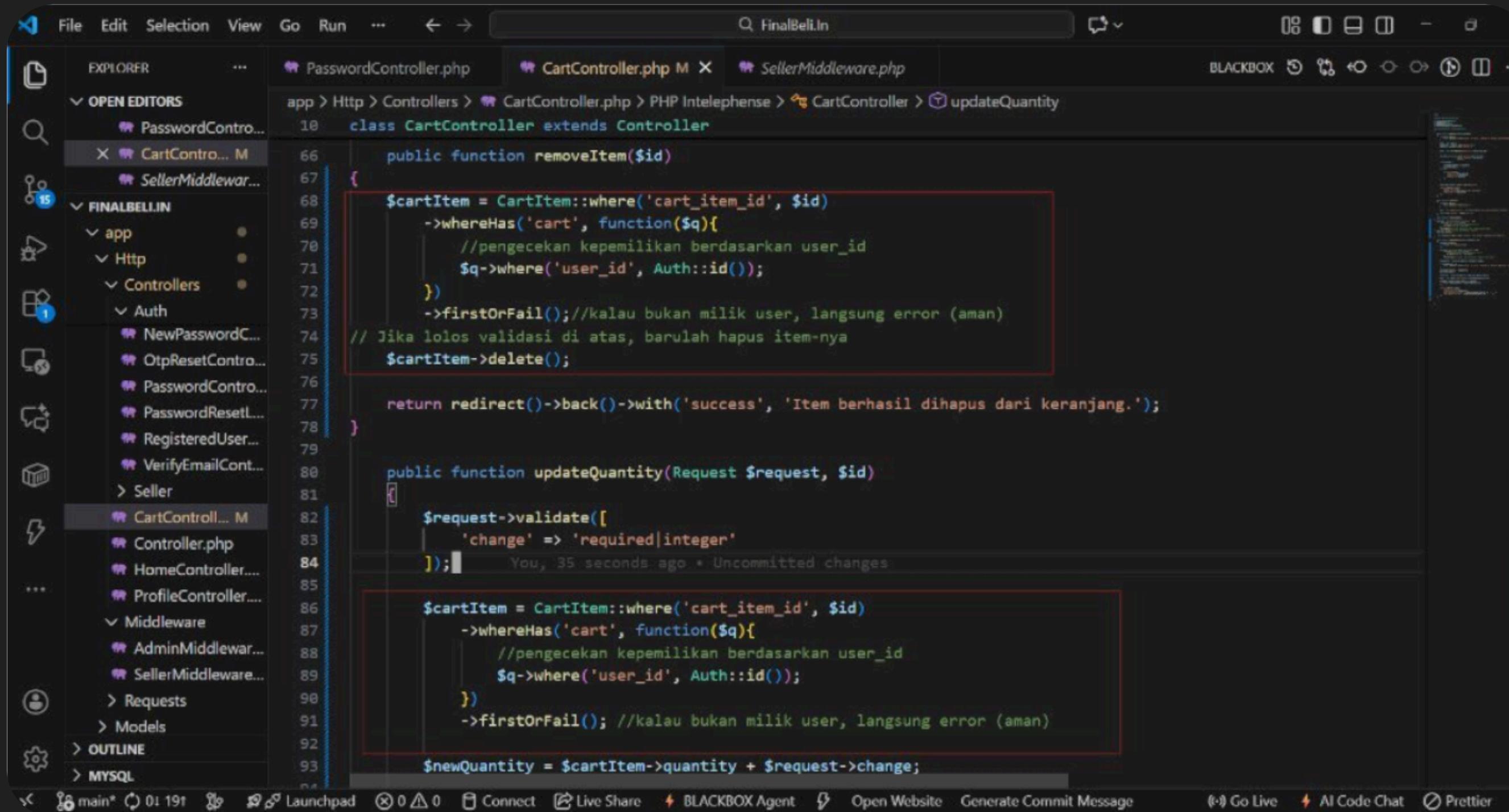
- Status: 200 OK
- Method: PATCH
- Domain: 192.168.100.34:8000
- File: /cart/update/17
- Initiator: checkout:28 (fetch)
- Type: json
- Transferred: 1,18 kB
- Size: 99 B
- Headers:
  - PATCH
  - Scheme: http
  - Host: 192.168.100.34:8000
  - Filename: /cart/update/17
  - Address: 192.168.100.34:8000
- Cookies: None
- Request: None
- Response: None
- Timings: None
- Stack Trace: None

- kodingan yang salah pada removeItem dan updateQuantity



```
File Edit Selection View Go Run ... ← → Q FinalBeli.In BLACKBOX - ...
EXPLORER PasswordController.php CartController.php M SellerMiddleware.php
OPEN EDITORS
>PasswordController.php
CartController.php M
SellerMiddleware.php
FINAL... app Http Controllers Auth
app
Http
Controllers
Auth
NewPasswordC...
OtpResetContro...
PasswordContro...
PasswordResetL...
RegisteredUser...
VerifyEmailCont...
Seller
CartController... M
Controller.php
HomeController...
ProfileController...
Middleware
AdminMiddleware...
SellerMiddleware...
Requests
Models
OUTLINE
MYSQL
app > Http > Controllers > CartController.php > PHP Intelephense > CartController > updateQuantity
10 class CartController extends Controller
11 {
12     public function removeItem($id)
13     {
14         $cartItem = CartItem::where('cart_item_id', $id)->first(); //line ini langsung ambil cart item berdasarkan id
15         // tanpa cek item ini milik siapa ?
16
17         if ($cartItem) {
18             $cartItem->delete();
19             return redirect()->back()->with('success', 'Item berhasil dihapus dari keranjang.');
20         }
21
22         return redirect()->back()->with('error', 'Item tidak ditemukan.');
23     }
24
25     public function updateQuantity(Request $request, $id)
26     {
27         $request->validate([
28             'change' => 'required|integer'
29         ]);
30
31         You, 1 second ago * Uncommitted changes
32         $cartItem = CartItem::where('cart_item_id', $id)->first();
33
34         if (!$cartItem) {
35             return response()->json(['status' => 'error', 'message' => 'Item tidak ditemukan'], 404);
36         }
37         $newQuantity = $cartItem->quantity + $request->change;
38
39     }
40 }
```

- **Remediation Strategy – Tambahkan validasi user\_id pada query, blokir akses unauthorized.**



The screenshot shows a code editor with two tabs open: PasswordController.php and CartController.php. The CartController.php tab is active, displaying PHP code for a Laravel application. Two specific sections of the code are highlighted with red boxes:

```
class CartController extends Controller
{
    public function removeItem($id)
    {
        $cartItem = CartItem::where('cart_item_id', $id)
            ->whereHas('cart', function($q){
                // pengecekan kepemilikan berdasarkan user_id
                $q->where('user_id', Auth::id());
            })
            ->firstOrFail(); // kalau bukan milik user, langsung error (aman)
        // Jika lolos validasi di atas, barulah hapus item-nya
        $cartItem->delete();

        return redirect()->back()->with('success', 'Item berhasil dihapus dari keranjang.');
    }

    public function updateQuantity(Request $request, $id)
    {
        $request->validate([
            'change' => 'required|integer'
        ]);
        $cartItem = CartItem::where('cart_item_id', $id)
            ->whereHas('cart', function($q){
                // pengecekan kepemilikan berdasarkan user_id
                $q->where('user_id', Auth::id());
            })
            ->firstOrFail(); // kalau bukan milik user, langsung error (aman)

        $newQuantity = $cartItem->quantity + $request->change;
    }
}
```

The first highlighted section is in the `removeItem` method, where it checks if the item belongs to the authenticated user before deleting it. The second highlighted section is in the `updateQuantity` method, where it performs a similar check before updating the item's quantity.

**VULN-002**

**BROKEN ACCESS CONTROL (IDOR) –  
BOLAIDOR ON SELLER PRODUCTS**

# Broken Access Control (IDOR)

## • BOLA/IDOR On Seller Products

- Executive Summary – Ditemukan bahwa role Seller dapat melihat, mengedit, dan menghapus produk milik seller lain tanpa validasi kepemilikan. Aplikasi gagal melakukan Object Ownership Validation sehingga menyebabkan Broken Object Level Authorization. Kerentanan ini memungkinkan sabotase bisnis antar seller karena seluruh aksi CRUD dapat dilakukan tanpa batasan user.
- CVSS Score – 8.7
- Severity – High
- CVSS String – CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:L/SC:N/SI:N/SA:N

The screenshot shows the CVSS Version 4.0 Calculator page on the FIRST website. The top navigation bar includes links for About FIRST, Membership, Initiatives, Standards & Publications, Events, Education, and Blog. A 'Member' button is also present. The main content area features the CVSS logo and the title 'Common Vulnerability Scoring System Version 4.0 Calculator'. Below this is a search bar containing the string 'CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:H/VI:H/VA:L/SC:N/SI:N/SA:N' with a 'Reset' button. The calculated 'CVSS v4.0 Score' is displayed as '8.7 / High'. A detailed description of the CVSS version 4.0 specification document is provided, mentioning its availability in various formats like JSON and XML. The right side of the page contains sections for 'Base Metrics' and 'Exploitability Metrics', each with dropdown menus for selecting values like 'Network (N)', 'Adjacent (A)', etc.

The screenshot shows a web browser window with the title "Laravel E-commerce Final" and the URL "http://192.168.100.34:8000/seller/dashboard". The page is titled "Beliin" and features a header with a user profile icon and a "Sign in" button.

The main content area is titled "galenium" and includes the sub-instruction "Kelola produk penjualan Anda". It displays two summary boxes: one showing "4 Total Produk" with a cube icon and another showing "142 Total Stok" with a stack of boxes icon.

A "Daftar Produk" (Product List) table is present, with a "Tambah Produk" (Add Product) button above it. The table has columns: PRODUK, KATEGORI, HARGA, STOK, and AKSI. The data is as follows:

PRODUK	KATEGORI	HARGA	STOK	AKSI
komputer ID: #5	elektronik	Rp 1.000.000	123 Unit	
Gelang ID: #4	Aksesoris	Rp 1.000.000	15 Unit	
kalung ID: #3	Aksesoris	Rp 10	2 Unit	
tv ID: #1	elektronik	Rp 1.000.000	2 Unit	

## BROKEN ACCESS CONTROL (IDOR) – BOLAVIDOR ON SELLER PRODUCTS

*Remediation Strategy:*

- *Tambahkan kolom user\_id pada tabel products.*
- *Filter produk di index:*  
`where('user_id', Auth::id())`
- *Validasi ownership pada update/delete:*  
`where('user_id', Auth::id())firstOrFail()`
- *Sembunyikan tombol edit/delete untuk produk non-owner.*

# **VULN-003**

## **WEAK PASSWORD CHANGE POLICY (PASSWORD REUSE ALLOWED)**

# WEAK PASSWORD CHANGE POLICY

## • (PASSWORD REUSE ALLOWED)

- Executive Summary – Sistem mengizinkan pengguna mengubah password namun tetap menggunakan nilai password lama. Hal ini melemahkan keamanan credential lifecycle dan membuat rotasi password tidak efektif.
- CVSS Score – 5.3
- Severity – Medium
- CVSS String – CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N

The screenshot shows the Common Vulnerability Scoring System (CVSS) Version 4.0 Calculator on the FIRST website. The calculator interface includes a sidebar with links to various CVSS resources like the specification document and user guide, and a main panel showing the calculated score and input fields for attack metrics.

**Common Vulnerability Scoring System (CVSS-SIG)**

- Calculator
- Specification Document
- User Guide
- Examples
- Frequently Asked Questions
- CVSS v4.0 Documentation & Resources
- CVSS v3.1 Archive
- CVSS v3.0 Archive
- CVSS v2 Archive
- CVSS v1 Archive
- JSON & XML Data Representations
- CVSS On-Line Training Course
- Identity & logo usage

**CVSS**

**Common Vulnerability Scoring System Version 4.0 Calculator**

CVSS:4.0/AV:N/AC:L/AT:N/PR:L/UI:N/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N

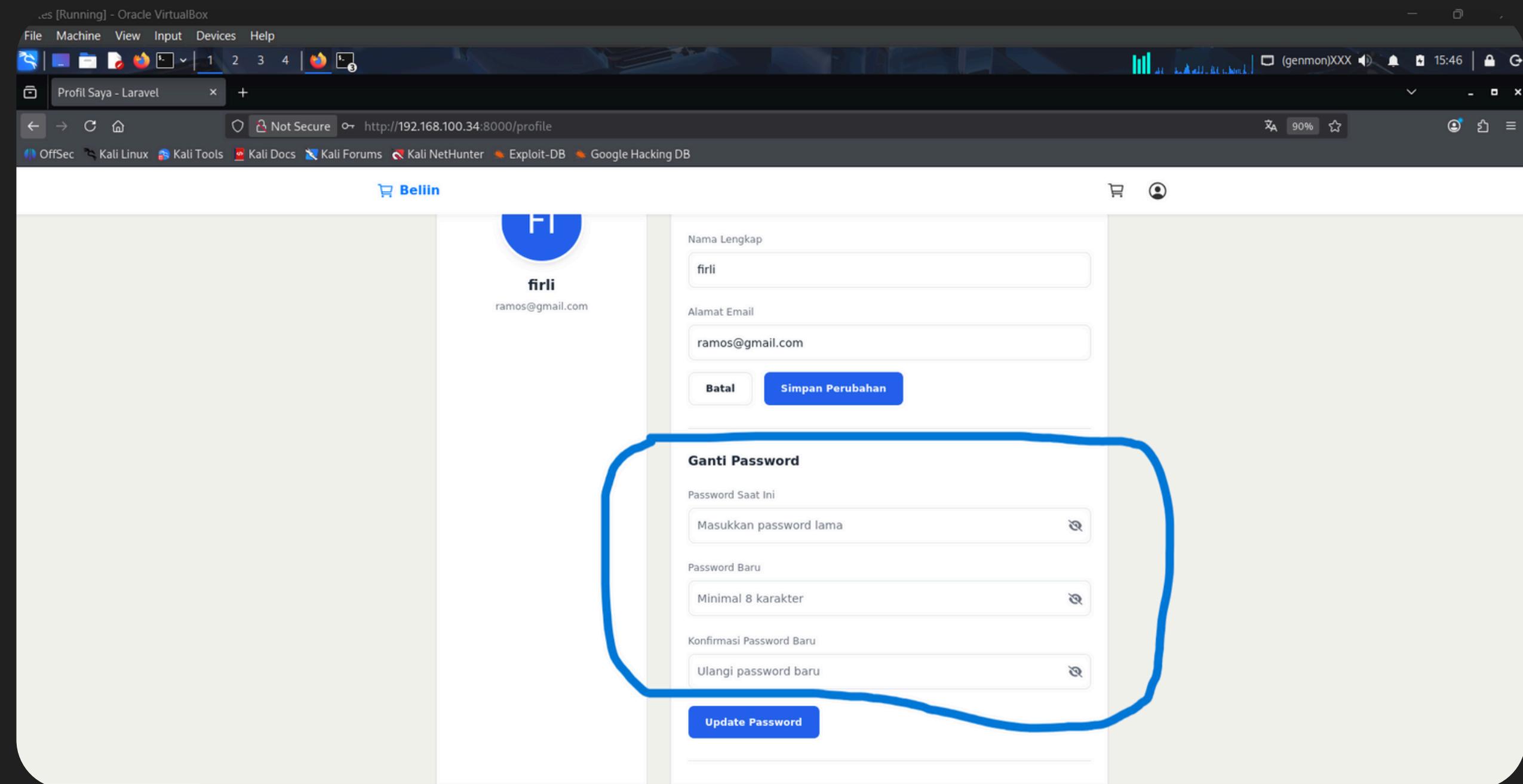
CVSS v4.0 Score: **5.3 / Medium**

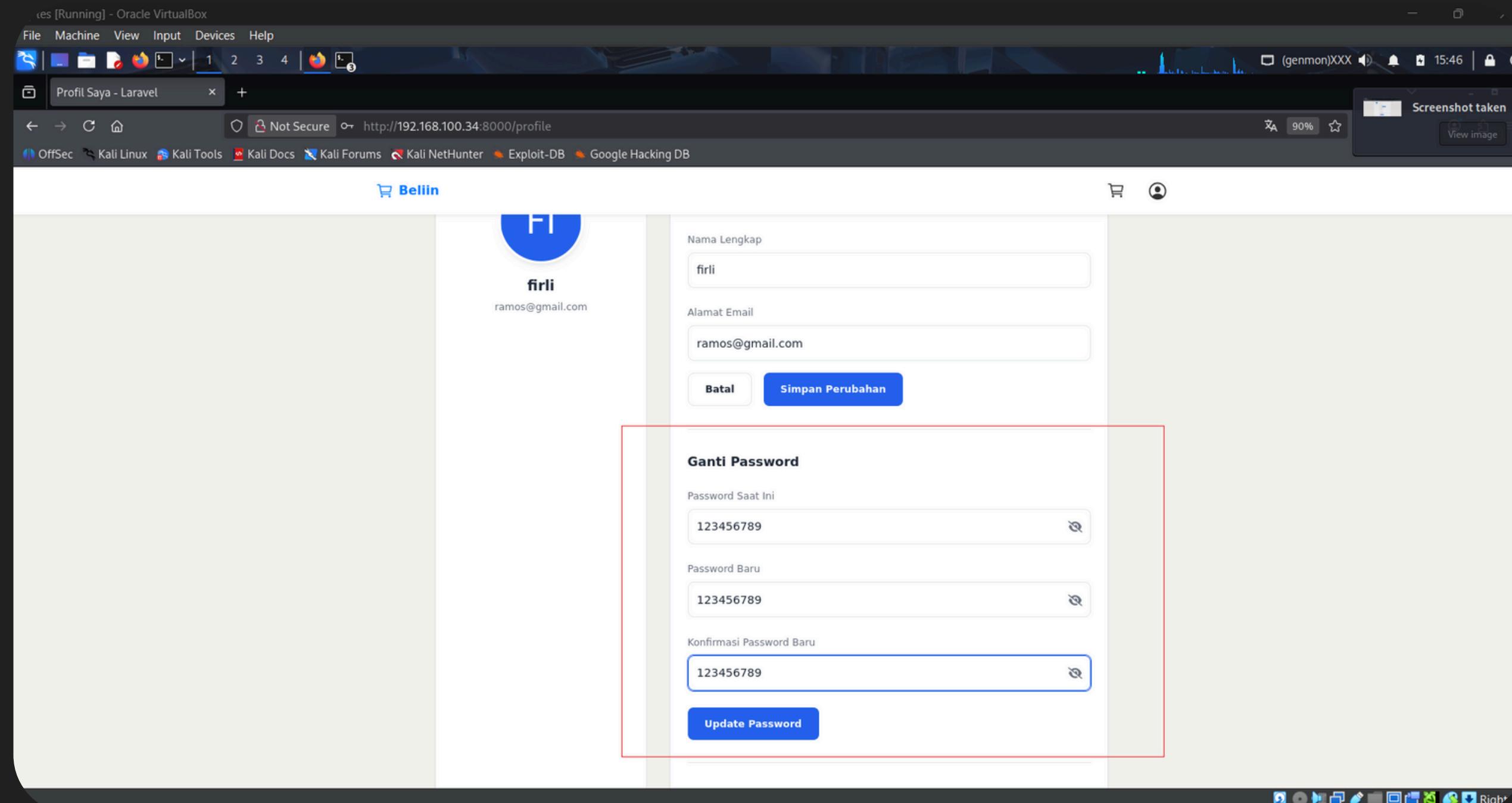
Hover over metric names and metric values for a summary of the information in the official CVSS v4.0 Specification Document. The Specification is available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerabilities, a set of Frequently Asked Questions (FAQ), and both JSON and XML Data Representations for all versions of CVSS.

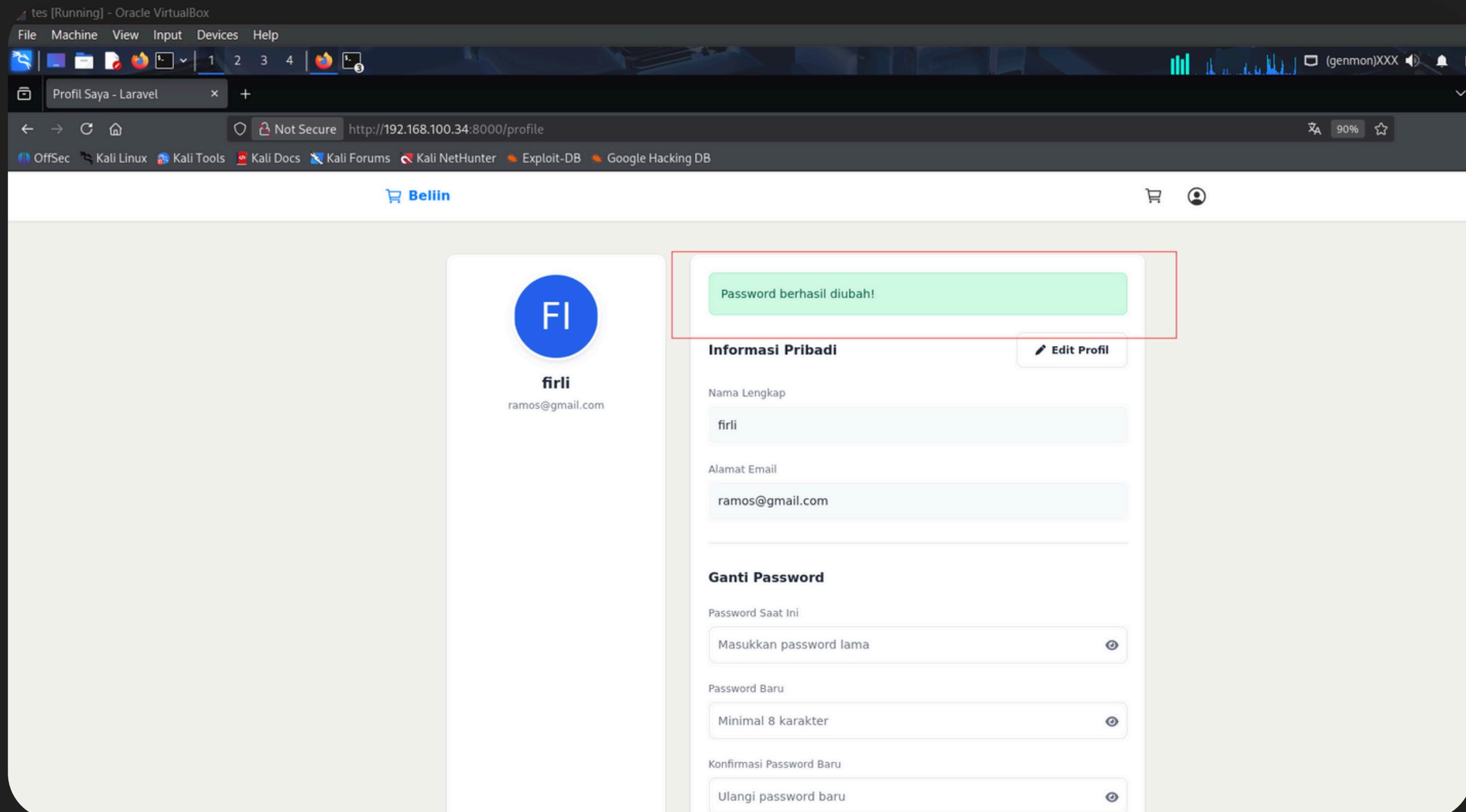
**Base Metrics ?**

**Exploitability Metrics**

Attack Vector (AV):	Network (N)	Adjacent (A)	Local (L)	Physical (P)
Attack Complexity (AC):	Low (L)	High (H)		
Attack Requirements (AT):	None (N)	Present (P)		
Privileges Required (PR):	None (N)	Low (L)	High (H)	







The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists various files under the 'FINALBELI.IN' project, including 'AdminController.php', 'AuthController.php', 'CartController.php', 'Controller.php', 'HomeController.php', and 'ProfileController.php'. The main editor area displays the 'PasswordController.php' file. The code is as follows:

```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Validation\Rules\Password;

class PasswordController extends Controller
{
    /**
     * Update the user's password.
     */
    public function update(Request $request): RedirectResponse
    {
        $validated = $request->validateWithBag('updatePassword', [
            'current_password' => ['required', 'current_password'],
            'password' => ['required', Password::defaults(), 'confirmed'],
        ]);

        $request->user()->update([
            'password' => Hash::make($validated['password']),
        ]);

        return back()->with('status', 'password-updated');
    }
}
```

The code uses Laravel's validation rules and facade methods like Hash::make to handle password updates. A red box highlights the entire class definition.

## WEAK PASSWORD CHANGE POLICY (PASSWORD REUSE ALLOWED)

### *Remediation Strategy*

- *Tambahkan validasi untuk mencegah password baru sama dengan password lama, Pastikan password rotation policy diterapkan sesuai security best practice.*

# VULN-004

## WEAK PASSWORD POLICY

# WEAK PASSWORD POLICY

- Executive Summary – Sistem menggunakan password yang hanya mensyaratkan minimal 8 karakter tapi tidak mewajibkan kombinasi Huruf besar, kecil, angka, simbol. hal ini bisa menjadikan penyerang menggunakan password lemah sehingga kredensial pengguna mudah ditebak, meningkatkan resiko brute force.
- CVSS Score – 6.9
- Severity – Medium
- CVSS String – CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:L/VA:N/SC:N/SI:L/SA:N

The screenshot shows the Common Vulnerability Scoring System (CVSS) Version 4.0 Calculator on the FIRST website. The calculator displays a score of 6.9 / Medium. The interface includes a sidebar with links to various CVSS resources and a main panel for inputting base metrics. The base metrics section shows the following values:

Base Metrics ?	
Exploitability Metrics	
Attack Vector (AV):	Network (N)
	Adjacent (A)
	Local (L)
	Physical (P)
Attack Complexity (AC):	Low (L)
	High (H)
Attack Requirements (AT):	None (N)
	Present (P)
Privileges Required (PR):	None (N)
	Low (L)
	High (H)
User Interaction (UI):	None (N)
	Passive (P)
	Active (A)

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists project files and folders. In the center is the main editor area displaying a PHP file named `PasswordController.php`.

The code in the editor is as follows:

```
11  class PasswordController extends Controller
12  {
13      /**
14      * Update the user's password.
15      */
16     public function update(Request $request): RedirectResponse
17     {
18         $validated = $request->validateWithBag('updatePassword', [
19             'current_password' => ['required', 'current_password'],
20             'password' => ['required', Password::defaults(), 'confirmed'],
21         ]);
22
23         // untuk mengecek password lama dan password baru
24         if (Hash::check($validated['password'], $request->user()->password)) {
25             return back()->withErrors([
26                 'password' => 'Password baru tidak boleh sama dengan password lama.',
27             ]);
28         }
29
30         $request->user()->update([
31             'password' => Hash::make($validated['password']),
32         ]);
33
34         return back()->with('status', 'password-updated');
35     }
36 }
37 }
```

A red rectangular box highlights the password comparison logic in lines 23-28. The code uses the `Hash::check` method to verify if the new password matches the old password stored in the database.

## WEAK PASSWORD CHANGE POLICY (PASSWORD REUSE ALLOWED)

*Remediation Strategy:*

1. *Terapkan password policy yang lebih ketat di file  
RegisteredUserController.php*
2. *Tambahkan juga mitigasi bruteforce seperti rate  
limiting/login throttling*

\* Security Programming

**Thank You**