5

BESHINCHI MA’RUZA

Korxona ilovalarini dasturlash · 5-mavzu

Autentifikatsiya, avtorizatsiya va xavfsizlik

Autentifikatsiya · Gate va Policy · RBAC (spatie/laravel-permission) · Audit jurnali · OWASP Top 10

Autentifikatsiya | RBAC | Policy | Audit | OWASP

ISINISH

Omborchi brauzer manzilidagi /invoices/45 ni /invoices/46 ga o‘zgartirdi. Tizim nima qilishi kerak?

Hozircha javob bermang. Avval bu so‘rovni kim va qayerda to‘xtatishi kerakligini o‘ylab ko‘ring.

1

Menyuda “Hisob-fakturalar” tugmasini yashirish yetarlimi?

2

Tizim bu omborchi aynan kimligini qayerdan biladi?

3

Ertaga rahbar “46-hisob-fakturani kim ko‘rgan?” deb so‘rasa, qanday javob beramiz?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz…

1. **Farqlash** — autentifikatsiya, avtorizatsiya va audit vazifalarini hamda 401 va 403 javoblarini
2. **Tushuntirish** — parollarni bcrypt/argon2id bilan xeshlash, sessiyani yangilash va login urinishlarini cheklash zaruratini
3. **Qo‘llash** — Laravel Gate va Policy yordamida obyekt darajasidagi ruxsatlarni (OrderPolicy)
4. **Loyihalash** — OmborPro uchun rol → ruxsat matritsasini (admin, menejer, omborchi, buxgalter)
5. **Sozlash** — spatie/laravel-permission paketini: HasRoles, seeder, middleware, kesh
6. **Yozish** — Eloquent hodisalari asosida o‘zgartirib bo‘lmaydigan audit jurnalini
7. **Baholash** — Laravel ilovasini OWASP Top 10 (2021) xavflari bo‘yicha

01

Autentifikatsiya: foydalanuvchini aniqlash

AuthN, AuthZ, audit · parol xeshlash · Auth::attempt · login cheklovi

Autentifikatsiya | Hash | Auth::attempt | RateLimiter

UCH SAVOL

Kim? Nima mumkin? Nima qildi?

Xavfsizlik bir-birini to‘ldiradigan uch mexanizmdan iborat. Bittasi yo‘q bo‘lsa, qolganlari ham yarim ishlaydi.

Autentifikatsiya (AuthN)

Foydalanuvchi kimligini isbotlash: e-mail + parol, 2FA kodi. Muvaffaqiyatsiz — `401` yoki login sahifasi.

Avtorizatsiya (AuthZ)

Aniqlangan foydalanuvchiga shu amal ruxsat etilganmi: `orders.confirm`, `OrderPolicy`. Ruxsat yo‘q — `403`.

Audit

Kim, qachon, qaysi IP dan, qaysi obyektni qanday o‘zgartirgani — `audit_logs` jadvalida.

Menyu tugmasini yashirish — avtorizatsiya emas. Tekshiruv har doim serverda, har bir so‘rovda bo‘ladi.

password_hash va password_verify

```php
$hash = password_hash('Ombor#2026', PASSWORD_BCRYPT, ['cost' => 10]);

echo substr($hash, 0, 7), PHP_EOL;   // algoritm va cost
echo strlen($hash), PHP_EOL;         // bcrypt xeshi uzunligi

var_dump(password_verify('Ombor#2026', $hash));
var_dump(password_verify('ombor#2026', $hash));

// cost oshirilgan bo'lsa, keyingi loginda qayta xeshlash kerak
echo password_needs_rehash($hash, PASSWORD_BCRYPT, ['cost' => 12])
    ? 'qayta xeshlash' : 'yaxshi';
```

▶ NATIJA

```
$2y$10$
60
bool(true)
bool(false)
qayta xeshlash
```

`$2y$10$`

xesh ichida algoritm (`2y` = bcrypt), cost va tuz saqlanadi — alohida ustun kerak emas

`password_verify`

tuzni xeshning o‘zidan olib, qayta hisoblaydi; katta-kichik harf farqlanadi

`password_needs_rehash`

cost yoki algoritm o‘zgarsa, login paytida xeshni yangilash imkonini beradi

Laravel

`Hash::make`, `Hash::check`, `Hash::needsRehash` — xuddi shu funksiyalar ustidagi qobiq

LOGIN

LoginController: xavfsiz kirish

```php
public function store(Request $request): RedirectResponse
{
    $credentials = $request->validate([
        'email' => ['required', 'email'],
        'password' => ['required', 'string'],
    ]);
    $credentials['is_active'] = true;     // bloklangan xodim kirmaydi

    if (! Auth::attempt($credentials, $request->boolean('remember'))) {
        throw ValidationException::withMessages([
            'email' => 'Email yoki parol noto‘g‘ri.',
        ]);
    }
    $request->session()->regenerate();    // session fixation

    return redirect()->intended(route('dashboard'));
}
```

▶ NATIJA

```
POST /login (to‘g‘ri)   → 302 Location: /dashboard
POST /login (xato)      → 302 back, errors.email
is_active = false       → xuddi “parol noto‘g‘ri” kabi
```

`Auth::attempt`

foydalanuvchini topadi, `Hash::check` qiladi va sessiyaga yozadi

`'is_active' => true`

qo‘shimcha shart — so‘rovga `where is_active = true` bo‘lib qo‘shiladi

`withMessages`

umumiy xabar: qaysi maydon xato ekani aytilmaydi

`regenerate()`

kirishdan keyin sessiya ID si yangilanadi

`intended()`

login sahifasiga yo‘naltirilgan asl manzilga qaytaradi

LOGIN OQIMI

Kirish so‘rovining xavfsiz yo‘li

1

Cheklov

`RateLimiter`: `email|ip` kaliti bo‘yicha 5 ta urinishdan keyin 429 va kutish vaqti

2

Validatsiya

`email` formati, `password` mavjudligi — `$request->validate()`

3

Tekshirish

`Auth::attempt`: e-mail + parol xeshi + `is_active`

4

Sessiya

`session()->regenerate()`, cookie: `HttpOnly`, `Secure`, `SameSite=Lax`

5

Audit

`Login` va `Failed` hodisalari `audit_logs` ga yoziladi (IP bilan)

Muvaffaqiyatli kirishda `RateLimiter::clear($key)` bilan hisoblagich nolga tushiriladi.

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

1-bo‘lim yakuni: Autentifikatsiya: foydalanuvchini aniqlash

1

401 va 403 javoblarining farqi nimada? OmborPro dan misol keltiring.

Savol 1

2

Nima uchun parolni `sha256` bilan saqlash ham xavfli?

Savol 2

3

Login dan keyin `session()->regenerate()` chaqirilmasa, qanday hujum mumkin?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: Parol mustahkamligi

Ro‘yxatdan o‘tishda parol korxona siyosatiga mos kelishini tekshiring. Qoidalar (shu tartibda tekshiriladi): uzunligi kamida 10 belgi (`uzunlik`), kamida bitta katta lotin harfi (`katta_harf`), kamida bitta kichik harf (`kichik_harf`), kamida bitta raqam (`raqam`), kamida bitta harf-raqam bo‘lmagan belgi (`belgi`). Agar e-mailning `@` gacha bo‘lgan qismi (login) kamida 3 belgidan iborat bo‘lsa va parol ichida u katta-kichik harfdan qat’i nazar uchrasa — `email` xatosi. `@` bo‘lmasa, login — butun satr. Parol faqat ASCII belgilardan iborat deb hisoblang.

Imzo: `function parol_tekshir(string $parol, string $email): array`

Qaytaradi: Buzilgan qoidalar kodlari ro‘yxati, yuqoridagi tartibda: masalan `['uzunlik', 'raqam']`. Parol mustahkam bo‘lsa — `[]`.

MISOL · CHAQIRUV	MISOL · NATIJA
`parol_tekshir('Ombor#2026x', 'ali@omborpro.test')`	`[]`

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

02

Avtorizatsiya: Gate va Policy

Gate · Policy · Gate::before · Gate::authorize · can middleware · @can

Gate | Policy | 403 | IDOR

QAYERDA TEKSHIRAMIZ

Avtorizatsiya vositalari

Vosita	Qachon	OmborPro misoli
`Gate::define`	modelga bog‘lanmagan amal	`view-reports` — hisobotlar sahifasi
Policy	aniq model obyekti ustidagi amal	`OrderPolicy::confirm($user, $order)`
`Gate::before`	hamma tekshiruvdan oldin	`admin` — har doim `true`
`permission:` / `can:` **middleware**	marshrut yoki guruh darajasida	`/invoices/*` → `permission:invoices.view`
`Gate::authorize()`	controller yoki servis ichida	ruxsat yo‘q → `AuthorizationException` → 403
`@can` **(Blade)**	faqat interfeys	“Tasdiqlash” tugmasini ko‘rsatish/yashirish

Qoida: middleware — “modulga kira oladimi?”, Policy — “aynan shu hujjat bilan shu amalni bajara oladimi?”.

POLICY

OrderPolicy: ruxsat + biznes holat

```php
class OrderPolicy
{
    public function update(User $user, Order $order): bool
    {
        return $user->can('orders.create')
            && $order->status === OrderStatus::Draft
            && (int) $order->created_by === $user->id;
    }
    public function confirm(User $user, Order $order): Response
    {
        if (! $user->can('orders.confirm')) {
            return Response::deny('Tasdiqlash huquqingiz yo‘q.');
        }
        return $order->status === OrderStatus::Draft
            ? Response::allow()
            : Response::deny('Faqat qoralama tasdiqlanadi.');
    }
}
```

▶ NATIJA

```
omborchi → POST /orders/124/confirm   → 403 Tasdiqlash huquqingiz
yo‘q.
menejer, status=confirmed             → 403 Faqat qoralama
tasdiqlanadi.
menejer, status=draft                 → 302 (tasdiqlandi)
```

`$user->can('orders.confirm')`

rol orqali berilgan ruxsat (spatie uni Gate ga ro‘yxatdan o‘tkazadi)

`OrderStatus::Draft`

4-mavzudagi holat enumi — biznes holat ham ruxsatning bir qismi

`created_by`

faqat muallif tahrirlaydi — boshqa menejerning qoralamasi emas

`Response::deny()`

403 bilan birga foydalanuvchiga sabab ko‘rsatiladi

ISHLATISH

Gate::before, Gate::authorize va marshrut

```php
// AppServiceProvider::boot() — admin uchun hamma narsa
Gate::before(fn (User $user, string $ability)
    => $user->hasRole('admin') ? true : null);

// OrderController::confirm(Order $order, OrderService $orders)
Gate::authorize('confirm', $order);      // yo'q bo'lsa — 403
$orders->confirm($order);

// istalgan joyda: true / false
$canEdit = $request->user()->can('update', $order);

// routes/web.php — marshrut darajasida
Route::post('/orders/{order}/confirm', [OrderController::class,
    'confirm'])->middleware('auth')->can('confirm', 'order');
```

▶ NATIJA

```
Gate::before → true   : admin, tekshiruv shu yerda tugaydi
Gate::before → null   : Policy::confirm() chaqiriladi
AuthorizationException → HTTP 403 Forbidden
```

`? true : null`

`null` — “men qaror qilmayman, Policy hal qilsin”; `false` hammani bloklaydi

`Gate::authorize`

Laravel 12 bazaviy Controller da `$this->authorize()` yo‘q — shu usul ishonchli

`->can('confirm', 'order')`

ikkinchi argument — marshrut parametri nomi (model binding)

`user()->can()`

shartli mantiq uchun: istisno emas, `bool` qaytaradi

▶ INTERAKTIV

Gate va Policy

Har bir holatda nima bo‘lishini tanlang. (Interaktiv slaydlarda bosib bajariladi.)

1 **Login qilmagan foydalanuvchi API da `/api/v1/orders` ga so‘rov yubordi. Qanday javob kutiladi?**

A) 200 OK
B) 401 Unauthorized
C) 403 Forbidden
D) 404 Not Found

2 **`Gate::before` admin bo‘lmaganlar uchun `false` qaytarsa nima bo‘ladi?**

A) Hech narsa o‘zgarmaydi
B) Faqat admin tekshiriladi
C) Admin bo‘lmagan hammaga barcha amallar taqiqlanadi
D) Policy ikki marta chaqiriladi

3 **“Tasdiqlash” tugmasi `@can` bilan yashirilgan, lekin controller da tekshiruv yo‘q. Bu xavfsizmi?**

A) Ha, tugma ko‘rinmaydi
B) Yo‘q — POST so‘rovni qo‘lda yuborish mumkin
C) Ha, CSRF himoya qiladi
D) Faqat HTTPS bo‘lsa xavfsiz

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

2-bo‘lim yakuni: Avtorizatsiya: Gate va Policy

1

Gate va Policy dan qaysi birini qachon ishlatasiz?

Savol 1

2

Laravel 12 controllerida `$this->authorize()` nima uchun ishlamasligi mumkin va o‘rniga nima yoziladi?

Savol 2

3

IDOR nima va Policy unga qarshi qanday himoya qiladi?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

2-topshiriq: OrderPolicy qarori

`OrderPolicy` mantiqini sof PHP da yozing. `$user = ['id' => int, 'roles' => [...]]`, `$order = ['status' => string, 'created_by' => int]`. Amallar: `view` — `menejer`, `omborchi` yoki `buxgalter` roli bo‘lsa; `update` — `menejer`, buyurtma `draft` va uni shu foydalanuvchi yaratgan bo‘lsa; `confirm` — `menejer` va holat `draft`; `cancel` — `menejer` va holat `draft`, `confirmed` yoki `reserved`. `admin` roli har qanday ma’lum amalga ruxsat oladi (`Gate::before` kabi). Noma’lum amal uchun (hatto admin uchun ham) `InvalidArgumentException` tashlang.

Imzo: `function buyurtma_ruxsat(array $user, string $amal, array $order): bool`

Qaytaradi: `true` — ruxsat berilgan, `false` — rad etilgan (403).

MISOL · CHAQIRUV

```
buyurtma_ruxsat(['id' => 5, 'roles' => ['menejer']], 'update',
['status' => 'draft', 'created_by' => 5])
```

MISOL · NATIJA

```
true
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

03

RBAC: rollar va ruxsatlar

Rol → ruxsat matritsasi · spatie/laravel-permission · HasRoles · kesh

RBAC | spatie/laravel-permission | HasRoles | Seeder

RUXSATLAR MATRITSASI

OmborPro: modul × rol

Modul	menejer	omborchi	buxgalter
Mahsulotlar	`products.view`	`products.view`, `products.manage`	—
Buyurtmalar	`orders.view`, `create`, `confirm`, `cancel`	`orders.view`	`orders.view`
Ombor	`stock.view`	`stock.view`, `stock.move`	—
Hisob-fakturalar	—	—	`invoices.view`, `invoices.create`
To‘lovlar	—	—	`payments.create`
Hisobotlar	`reports.view`	—	`reports.view`
Foydalanuvchilar, audit	—	—	—

`admin` — barcha 14 ta ruxsat (`users.manage`, `audit.view` faqat unda) va `Gate::before`.

SEEDER

RolesAndPermissionsSeeder::run()

```php
app()[PermissionRegistrar::class]->forgetCachedPermissions();
$matrix = [
    'admin' => ['users.manage', 'audit.view'],    // + hammasi pastda
    'menejer' => ['orders.view', 'orders.create', 'orders.confirm',
        'orders.cancel', 'products.view', 'stock.view', 'reports.view'],
    'omborchi' => ['stock.view', 'stock.move', 'products.view',
        'products.manage', 'orders.view'],
    'buxgalter' => ['invoices.view', 'invoices.create',
        'payments.create', 'reports.view', 'orders.view'],
];
foreach ($matrix as $role => $permissions) {
    foreach ($permissions as $name) {
        Permission::findOrCreate($name, 'web');
    }
    Role::findOrCreate($role, 'web')->syncPermissions($permissions);
}
Role::findByName('admin', 'web')->syncPermissions(Permission::all());
$user->assignRole('menejer');          // foydalanuvchiga rol
```

▶ NATIJA

```
php artisan db:seed --class=RolesAndPermissionsSeeder
roles: 4 · permissions: 14
admin: 14 · menejer: 7 · omborchi: 5 · buxgalter: 5
```

`forgetCachedPermissions()`

eski kesh yangi matritsani “yashirmasligi” uchun

`findOrCreate`

seeder qayta ishga tushsa ham takror yozuv yaratmaydi (idempotent)

`syncPermissions`

rolning ruxsatlarini aynan shu ro‘yxatga tenglashtiradi — ortiqchasi olib tashlanadi

`assignRole`

`HasRoles` trait i bergan metod; `syncRoles()` — rollarni almashtiradi

RBAC tekshiruvi qanday ishlaydi

```php
$roles = [
    'menejer' => ['orders.view', 'orders.create', 'orders.confirm'],
    'omborchi' => ['stock.view', 'stock.move', 'orders.view'],
    'buxgalter' => ['invoices.create', 'payments.create'],
];
function can(array $userRoles, string $perm, array $roles): bool
{
    if (in_array('admin', $userRoles, true)) {
        return true;                        // Gate::before analogi
    }
    $lists = array_map(fn ($r) => $roles[$r] ?? [], $userRoles);
    return in_array($perm, array_merge([], ...$lists), true);  // rad
}
foreach ([[['omborchi'], 'stock.move'], [['omborchi'], 'orders.create'],
    [['menejer', 'buxgalter'], 'payments.create'], [[], 'orders.view'],
    [['admin'], 'users.manage']] as [$userRoles, $perm]) {
    echo can($userRoles, $perm, $roles) ? 'ha' : 'yo\'q', PHP_EOL;
}
```

▶ NATIJA

```
ha
yo'q
ha
yo'q
ha
```

`array_merge(...)`

foydalanuvchining barcha rollari ruxsatlari birlashtiriladi

`?? []`

noma’lum rol ruxsat bermaydi — sukut bo‘yicha rad

`can([], ...)`

rolsiz foydalanuvchi hech narsa qila olmaydi

`admin`

`Gate::before` kabi: tekshiruv boshidayoq `true`

▶ INTERAKTIV

Kimga qaysi ruxsat?

Har bir amalni OmborPro matritsasi bo‘yicha to‘g‘ri rolga joylashtiring. (Interaktiv slaydlarda bosib bajariladi.)

Yangi buyurtma yaratish (orders.create)

Buyurtmani tasdiqlash (orders.confirm)

Omborga kirim qilish (stock.move)

Inventarizatsiya tuzatishi (adjust)

Mahsulot kartochkasini tahrirlash (products.manage)

Hisob-faktura chiqarish (invoices.create)

Payme to‘lovini qayd etish (payments.create)

Buyurtmani bekor qilish (orders.cancel)

menejer

omborchi

buxgalter

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

3-bo‘lim yakuni: RBAC: rollar va ruxsatlar

1

Nima uchun kodda `hasRole()` emas, `can()` ni tekshirish afzal?

Savol 1

2

Vazifalarni ajratish (segregation of duties) OmborPro da qanday namoyon bo‘ladi?

Savol 2

3

Ruxsat berildi, lekin foydalanuvchida hali ham 403. Birinchi nimani tekshirasiz?

Savol 3

MA’RUZADAGI MASHQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

3-topshiriq: Ruxsat: aniq nom va wildcard

RBAC tekshiruvini yozing. `$rollar` — rol nomi → ruxsatlar ro‘yxati. Ruxsat namunasi aniq nom (`orders.view`), modul bo‘yicha wildcard (`orders.*` — `orders.` bilan boshlanadigan har qanday ruxsat) yoki `*` (hammasi) bo‘lishi mumkin. Foydalanuvchi rollaridan kamida bittasida mos namuna bo‘lsa — ruxsat bor. `admin` roli xaritada bo‘lmasa ham har doim ruxsat oladi. Foydalanuvchida xaritada yo‘q (va `admin` bo‘lmagan) rol bo‘lsa — `DomainException`.

Imzo: `function ruxsat_bormi(array $foydalanuvchiRollari, string $ruxsat, array $rollar): bool`

Qaytaradi: `true` yoki `false`.

MISOL · CHAQIRUV

```
ruxsat_bormi(['omborchi'], 'stock.move', ['omborchi' => ['stock.*', 'orders.view']])
```

MISOL · NATIJA

```
true
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. PHP kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

Audit jurnali: kim, qachon, nimani o‘zgartirdi

Audit maydonlari · Eloquent hodisalari · o‘zgartirib bo‘lmaslik · maxfiy maydonlar

audit_logs

Eloquent events

Append-only

A09

AUDIT YOZUVI

Har bir yozuv oltita savolga javob beradi

1 Kim?

`user_id` — tizim jarayoni (navbat, scheduler) bo‘lsa `null`

2 Nima?

`action`: `created`, `updated`, `deleted`, `login`, `login_failed`, `role_changed`

3 Qaysi obyekt?

`auditable_type` + `auditable_id`: `App\Models\Product #12`

4 Oldin → keyin

`old_values`, `new_values` — faqat o‘zgargan maydonlar, JSON

5 Qayerdan?

`ip` — `request()->ip()`; kerak bo‘lsa user agent

6 Qachon?

`created_at` UTC da; `updated_at` yo‘q — yozuv o‘zgarmaydi

TRAIT

Auditable: updated hodisasi

```php
trait Auditable
{
    public static function bootAuditable(): void
    {
        static::updated(function (Model $model) {
            $hide = ['password', 'remember_token', 'updated_at'];
            $new = Arr::except($model->getChanges(), $hide);
            $old = Arr::only($model->getOriginal(), array_keys($new));
            AuditLog::create([
                'user_id' => auth()->id(), 'action' => 'updated',
                'auditable_type' => $model->getMorphClass(),
                'auditable_id' => $model->getKey(),
                'old_values' => $old, 'new_values' => $new,
                'ip' => request()->ip(),
            ]);
        });
    }
}
```

▶ NATIJA

```
audit_logs #981: user_id=7, action=updated
auditable = App\Models\Product #12
old_values = {"price":"125000.00"}
new_values = {"price":"119000.00"}
ip = 10.0.5.23
```

`bootAuditable()`

Laravel `boot` + trait nomi ko‘rinishidagi statik metodni avtomatik chaqiradi

`getChanges()`

`save()` da haqiqatan o‘zgargan maydonlar

`getOriginal()`

`updated` hodisasida hali eski qiymatlar

`Arr::except`

parol va token jurnalga tushmaydi

`getMorphClass()`

polimorf tur nomi — `auditable_type` ustuni uchun

O‘zgarishlar farqi va maxfiy maydonlar

```php
function auditDiff(array $old, array $new, array $hidden): array
{
    $diff = ['old' => [], 'new' => []];
    foreach ($new as $key => $value) {
        if (array_key_exists($key, $old) && $old[$key] === $value) {
            continue;                       // o'zgarmagan
        }
        $mask = in_array($key, $hidden, true);
        $diff['old'][$key] = $mask ? '***' : ($old[$key] ?? null);
        $diff['new'][$key] = $mask ? '***' : $value;
    }
    return $diff;
}
$old = ['price' => '125000.00', 'name' => 'Un', 'password' => 'h1'];
$new = ['price' => '119000.00', 'name' => 'Un', 'password' => 'h2'];
echo json_encode(auditDiff($old, $new, ['password']));
```

▶ NATIJA

```
{"old":{"price":"125000.00","password":"*"},"new":{"price":"119000.00","password":"*"}}
```

`===`

qat’iy taqqoslash: `'10'` va `10` farq qiladi — Eloquent cast larini hisobga oling

`array_key_exists`

`null` qiymatli maydon ham “bor” hisoblanadi (`isset` dan farqi)

`'***'`

parol o‘zgargani ko‘rinadi, lekin qiymati emas

`name`

o‘zgarmagan maydon jurnalni to‘ldirmaydi

AMALIYOT

Audit jurnali: xato va to‘g‘ri yondashuv

✗ Noto‘g‘ri

- Auditni faqat `laravel.log` ga yozish — 14 kunda aylanib o‘chadi
- `old_values` ga parol xeshi va tokenlarni ham yozish
- Admin uchun “jurnalni tozalash” tugmasi
- `Product::where(...)->update()` bilan narxlarni ommaviy o‘zgartirish — iz qolmaydi
- Faqat muvaffaqiyatli amallarni yozish

✓ To‘g‘ri

- Alohida `audit_logs` jadvali, indekslar va saqlash muddati siyosati
- `password`, `remember_token` — `***` yoki umuman yozilmaydi
- Faqat qo‘shish (append-only): DB da `REVOKE UPDATE, DELETE`
- Ommaviy o‘zgarish servis orqali — har bir model `save()` yoki aniq audit yozuvi
- `login_failed` va 403 urinishlari ham yoziladi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: Audit jurnali: kim, qachon, nimani o‘zgartirdi

1

Audit jurnali texnik logdan nimasi bilan farq qiladi?

Savol 1

2

Nima uchun `Product::where('category_id', 3)->update(['price' => ...])` audit jurnaliga tushmaydi?

Savol 2

3

Audit yozuvlarini o‘zgartirib bo‘lmasligini qaysi ikki darajada ta’minlaymiz?

Savol 3

05

OWASP Top 10 va Laravel himoya vositalari

A01–A10 (2021) · injection va XSS · mass assignment · CSRF · sozlama

OWASP Top 10 | SQL injection | XSS | CSRF | Mass assign

A01–A05: OmborPro da qanday ko‘rinadi

Kod	Xavf	OmborPro misoli	Laravel himoyasi
A01	Broken Access Control	omborchi /invoices/46 ni ochadi (IDOR)	Policy, `permission:` middleware, `Gate::authorize`
A02	Cryptographic Failures	parol `md5`, STIR ochiq matnda	bcrypt/argon2id, `encrypted` cast, HTTPS
A03	Injection (SQL, XSS)	`whereRaw("sku = '$sku'")`, `{!! $name !!}`	binding, `Rule::in` ustunlar, `{{ }}`
A04	Insecure Design	manfiy qoldiq, tasdiqsiz bekor qilish	invariantlar, holat mashinasi (4-mavzu)
A05	Security Misconfiguration	production da `APP_DEBUG=true`	`.env` nazorati, Telescope o‘chiq

A06–A10: davomi

Kod	Xavf	OmborPro misoli	Laravel himoyasi
A06	Vulnerable and Outdated Components	eski paket versiyasidagi ma’lum zaiflik	`composer audit`, muntazam yangilash
A07	Identification and Authentication Failures	cheksiz login urinishlari, zaif parol	`RateLimiter`, parol qoidalari, 2FA
A08	Software and Data Integrity Failures	imzosiz webhook, soxtalashtirilgan havola	`hash_hmac` imzo, `signed` middleware
A09	Security Logging and Monitoring Failures	kim narxni o‘zgartirgani noma’lum	`audit_logs`, `login_failed`, monitoring
A10	Server-Side Request Forgery	URL bo‘yicha rasm yuklash ichki serverga boradi	host ro‘yxati, ichki IP larni taqiqlash

OWASP Top 10 — tekshiruv ro‘yxati emas, xabardorlik hujjati. Chuqur tekshiruv uchun OWASP ASVS ishlatiladi.

Xavfli va xavfsiz yozuv

✗ Noto‘g‘ri

- `DB::select("select * from products where sku = '$sku'")`
- `->orderByRaw($request->input('sort'))`
- `{!! $product->name !!}` — mahsulot nomida `<script>` bo‘lsa ishlaydi
- `User::create($request->all())`
- `if ($signature == $expected)`

✓ To‘g‘ri

- `Product::where('sku', $sku)->first()` — binding
- `'sort' => Rule::in(['name', 'price', 'sku'])` + `orderBy($sort)`
- `{{ $product->name }}` — avtomatik ekranlash
- `$fillable` + `User::create($request->validated())`
- `hash_equals($expected, $signature)` — vaqtga chidamli

XAVFSIZ QIDIRUV

ProductController::index — validatsiya va binding

```php
public function index(Request $request)
{
    $data = $request->validate([
        'q' => ['nullable', 'string', 'max:100'],
        'sort' => ['nullable', Rule::in(['name', 'price', 'sku'])],
    ]);

    return Product::query()
        ->when($data['q'] ?? null, fn ($query, $term) =>
            $query->where('name', 'ilike', "%{$term}%"))
        ->orderBy($data['sort'] ?? 'name')
        ->paginate(20);
}
```

▶ NATIJA

```
?q=un&sort=price →
select * from "products" where "name"::text ilike ?
  and "products"."deleted_at" is null
  order by "price" asc limit 20 offset 0   bindings: ["%un%"]
?sort=id;drop table users → validatsiya xatosi, SQL bajarilmaydi
```

`Rule::in([...])`

ustun nomi faqat ruxsat etilgan ro‘yxatdan — identifikatorlar binding bilan himoyalanmaydi

`where(..., 'ilike', ...)`

qidiruv matni SQL ga emas, `?` binding ga tushadi

`max:100`

juda uzun so‘rovlar bilan bazani yuklashning oldi olinadi

`deleted_at is null`

`SoftDeletes` (3-mavzu) o‘chirilganlarni avtomatik chiqarib tashlaydi

▶ INTERAKTIV

Qaysi OWASP xavfi?

Har bir holatni tegishli toifaga joylashtiring. (Interaktiv slaydlarda bosib bajariladi.)

Buxgalter `/admin/users` sahifasini ochib, rollarni o‘zgartira oladi

Menejer manzildagi ID ni o‘zgartirib, boshqa menejerning qoralamasini tahrirlaydi

`orderByRaw($request->sort)`

Kontragent nomi `{!! $c->name !!}` bilan chiqariladi

`whereRaw("inn = '$inn'")`

Production serverda `APP_DEBUG=true`

Telescope production da hammaga ochiq

`.env` fayli git repozitoriysiga tushib qolgan

A01 Kirish nazorati

A03 Injection / XSS

A05 Noto‘g‘ri sozlama

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: OWASP Top 10 va Laravel himoya vositalari

1

OWASP Top 10 (2021) da birinchi o‘rinda qaysi xavf turadi va Laravel da unga qarshi qaysi vositalar bor?

Savol 1

2

Nima uchun `orderBy($request->sort)` binding bo‘lsa ham xavfli va qanday tuzatiladi?

Savol 2

3

`{{ }}` va `{!! !!}` ning farqi nimada?

Savol 3

XULOSA

Asosiy xulosalar

1 Uch savol: kim, nima mumkin, nima qildi

Autentifikatsiya (401), avtorizatsiya (403) va audit jurnali bir-birini to‘ldiradi.

2 Parol — faqat sekin, tuzli xesh

bcrypt/argon2id, `Hash::make`, `password_verify`; login dan keyin `session()->regenerate()`.

3 Policy — obyekt darajasidagi qoida

Rol ruxsati + biznes holat; `Gate::authorize`, `->can()`, `Gate::before` da `null`.

4 RBAC — ruxsatni tekshiring, rolni emas

spatie/laravel-permission: `HasRoles`, seeder, `permission:` middleware, keshni tozalash.

5 Audit — faqat qo‘shish mumkin

Eloquent hodisalari, eski/yangi qiymatlar, maxfiy maydonlar `***`, ommaviy `update()` ga e’tibor.

6 OWASP Top 10 — tekshiruv yo‘nalishi

Binding, `{{ }}`, `$fillable`, `@csrf`, `APP_DEBUG=false`, `composer audit`, imzolangan havolalar.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. Login qilmagan foydalanuvchi va huquqsiz foydalanuvchi qanday HTTP kod oladi?
2. Nima uchun bir xil parol ikki xil bcrypt xeshi beradi?
3. `Auth::attempt` ga `'is_active' => true` qo‘shish nima beradi?
4. `Gate::before` admin bo‘lmaganlar uchun nima qaytarishi kerak?
5. Laravel 12 da Policy qanday ulanadi?
6. spatie paketida ruxsatlar keshini qanday tozalaymiz?
7. Qaysi Eloquent amali audit hodisasini chaqirmaydi?
8. SQL injection dan Eloquent qanday himoya qiladi va qayerda himoya qilmaydi?

AMALIY MASHG‘ULOT

5-amaliy mashg‘ulot: 5 ta PHP topshiriq va loyiha bosqichi

4 **Ruxsatlarni modullar bo‘yicha guruhlash**
Oson · 10 ball

5 **Audit: o‘zgargan maydonlar**
O‘rta · 20 ball

6 **Login urinishlarini cheklash**
O‘rta · 20 ball

7 **Imzolangan havola (signed URL)**
Qiyin · 30 ball

8 **O‘zgartirib bo‘lmaydigan audit zanjiri**
Qiyin · 30 ball

Telefon kamerasi bilan skanerlang

PHP kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

Loyiha havolasini ham shu sahifada topshirasiz; o‘qituvchi baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

OmborPro da buxgalter buyurtmani tasdiqlay olmasligini qaysi uchta joyda ta’minlaysiz?

2

Audit jurnalida mahsulot narxi o‘zgarganda qaysi maydonlar to‘ldiriladi? Misol yozing.

3

OWASP Top 10 dan qaysi xavf sizning jamoa loyihangiz uchun eng dolzarb va nima uchun?