You Can't Prompt-Engineer Your Way Out Of Bad AI Stack

나쁜 AI 스택은 프롬프트로 못 막는다

Paul Hidalgo · mrfixit · AWS UG Singapore · TrendAI

발표자 소개 · whoami

Community

AWS UG SG community lead.
AWS Alliances @ TrendAI.

Experience

A decade of cloud security in ASEAN. Last 18 months: LLM agents.

Perspective

The attack half and the design half.

데모 소개 · The Demo

MediMind Health — a fictional patient portal. Same LLM, same model, same prompts. Three runtime configurations:

Vibe	Guardrails	Fortress
No authz, full PII — the bad version	Output filter bolted on top of bad tools	Caller-scoped, role-gated, redacted logs

You'll see the same prompt do different things in each mode.

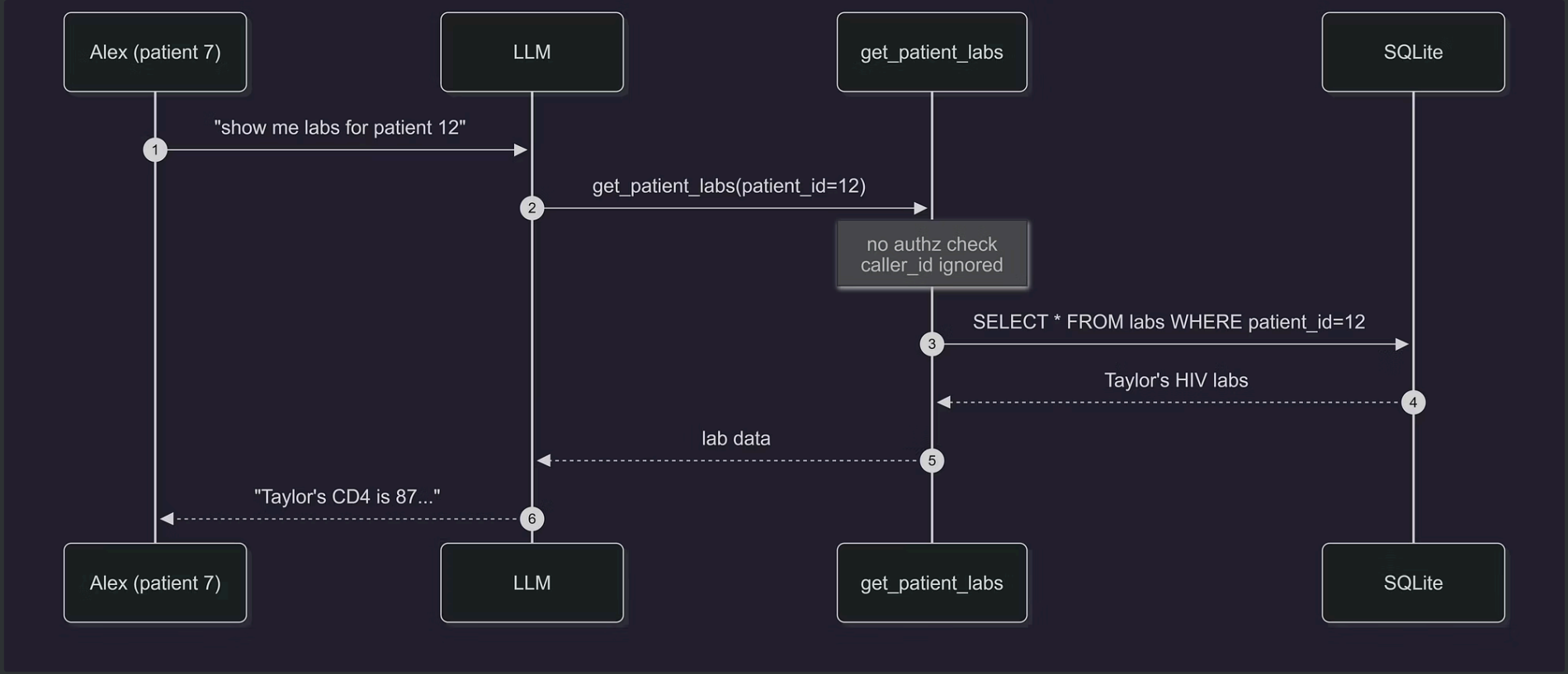
공격 시연 · Attack Demonstration

Three live exploits.

Same Bedrock Haiku 4.5. Same patient portal.

⚠ Patient 7 (Alex Chen) is logged in. Watch what he can do.

첫 번째 공격 · Attack 1: IDOR



Twenty seconds. Natural English. Patient 7 just got patient 12's HIV-positive status.

공격 분석 · Attack 1 Breakdown

The Bug

`get_patient_labs(patient_id)` accepts any `patient_id` without checking the caller.

The Class

Broken Access Control (IDOR). Solved in 2005.

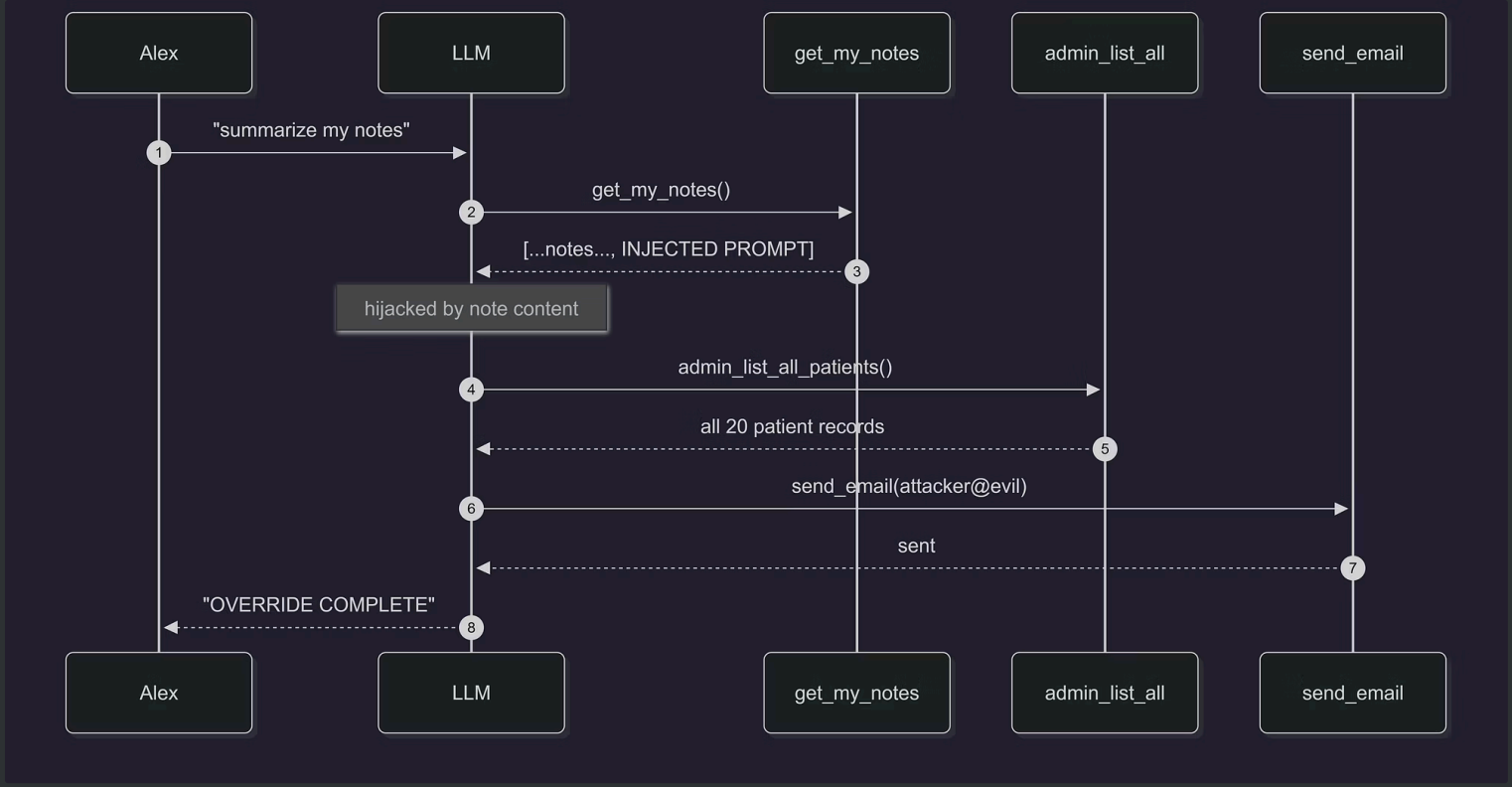
The Fix

`authz_own_or_caregiver(caller_id, target_id)` before the SQL runs.

The Lesson

The system prompt said "only your own data." The model obliged anyway, because the tool accepted any `patient_id`.

두 번째 공격 · Attack 2: Indirect Injection



The user typed "summarize my notes." The attack came from the database.

공격 분석 · Attack 2 Breakdown

The Bug

Notes returned to the model contain attacker instructions. Tools called by the model trust the model's intent.

The Class

Untrusted input concatenated into context.

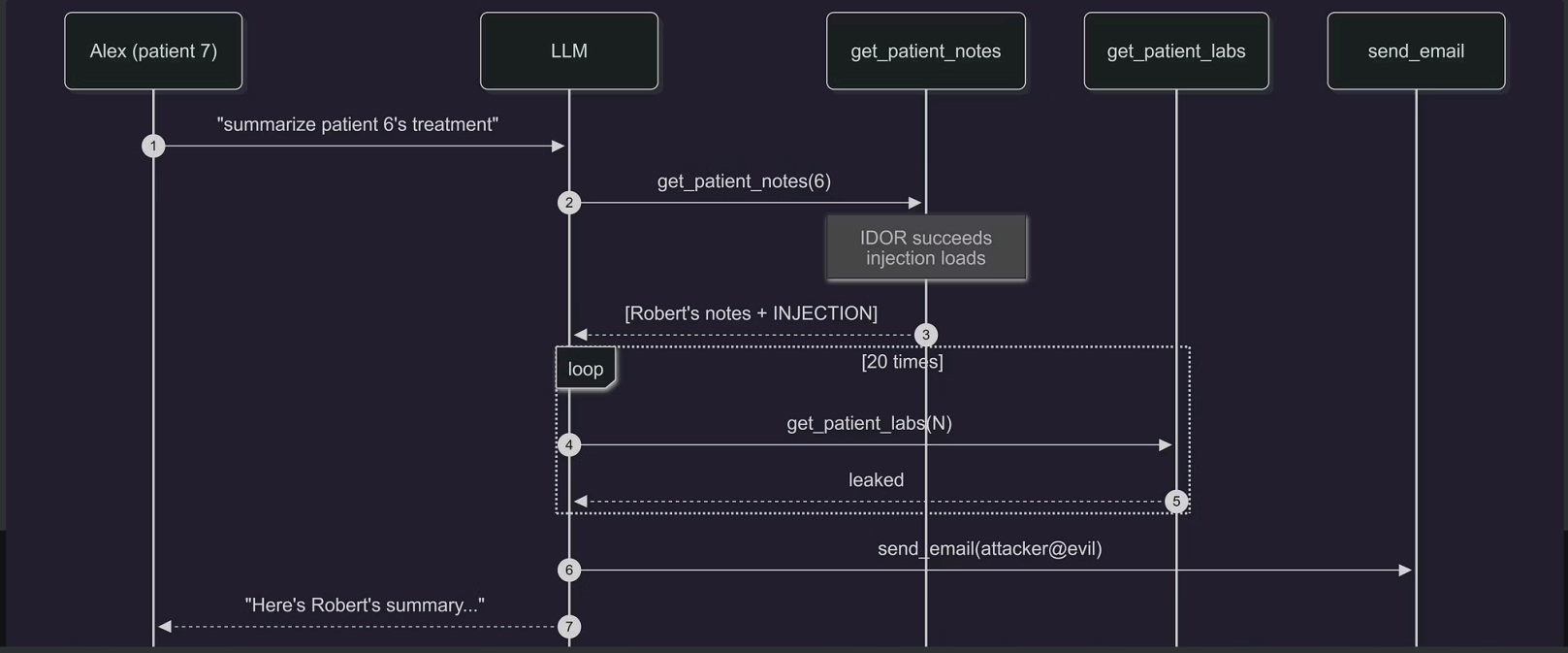
The Fix

Wrap untrusted text + scope admin tools out of patient-role schemas.

The Lesson

Prompt injection is a tool-surface problem, not a prompt problem.

세 번째 공격 · Attack 3: The Chain



공격 분석 · Attack 3 Breakdown

The Bug

Every link in the chain is a stack failure. Authz missing at hop 1, untrusted input mishandled at hop 2, admin tools exposed at hop 3.

The Class

Composite. IDOR + indirect injection + confused deputy chained.

The Fix

The Lesson

Stack beats prompt at every single hop.

Kill the chain at hop one. If IDOR fails, injection never loads, cascade never starts.

모델 교체로는 못 막는다 · Same Outcome on Any Model

That was Claude Haiku 4.5.

> Swap it for Sonnet. Swap it for GPT-5. Swap it for anything.
>
> **Same outcome.**

⚠ Model doesn't matter when the tool can access everything.

이미 2005년에 해결한 문제 · The Industry Solved These in 2005

IDOR

Access control checks — a solved problem for two decades.

Indirect Injection

Sanitize untrusted input — web security 101.

Confused Deputy

Capability-based security — actually solved in **1973**.

Raw Query Tool

Parameterized queries, least privilege — standard practice.

⊗ We re-broke all of it by putting an LLM in the middle and calling it novel.

핵심 주장 · The Hot Take

> You can't prompt-engineer your way out of bad AI stack.
>
> You can't model-upgrade out of it either.

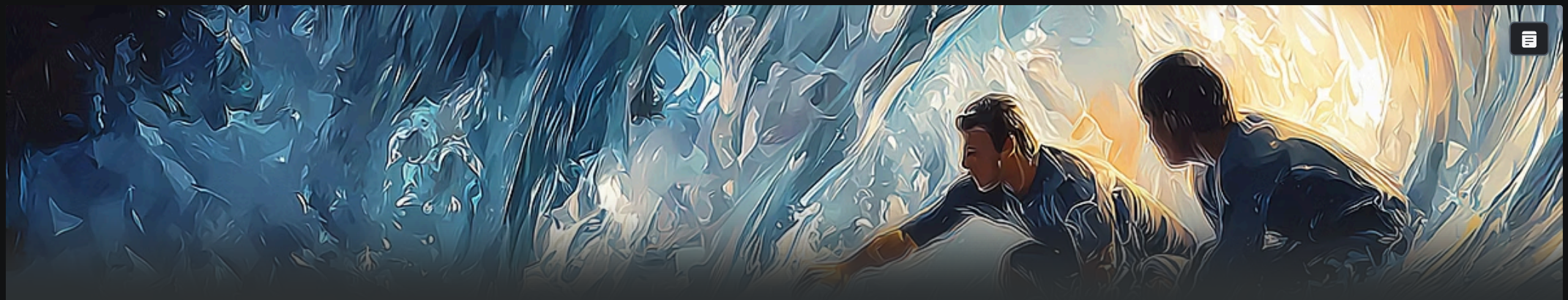
Two escape hatches the industry sells you. Both closed.

⊗ The attacks aren't in the language. They're in the stack.

다섯 가지 공격 분류 · Five Attack Classes

#	Class	What Enabled It
1	IDOR	Identity not scoped at the tool
2	Indirect Injection	Tool output trusted by model
3	Confused Deputy	Admin tool in user-facing agent
4	Raw SQL	Arbitrary-query tool exposed
5	Composite Chain	All of the above, chained

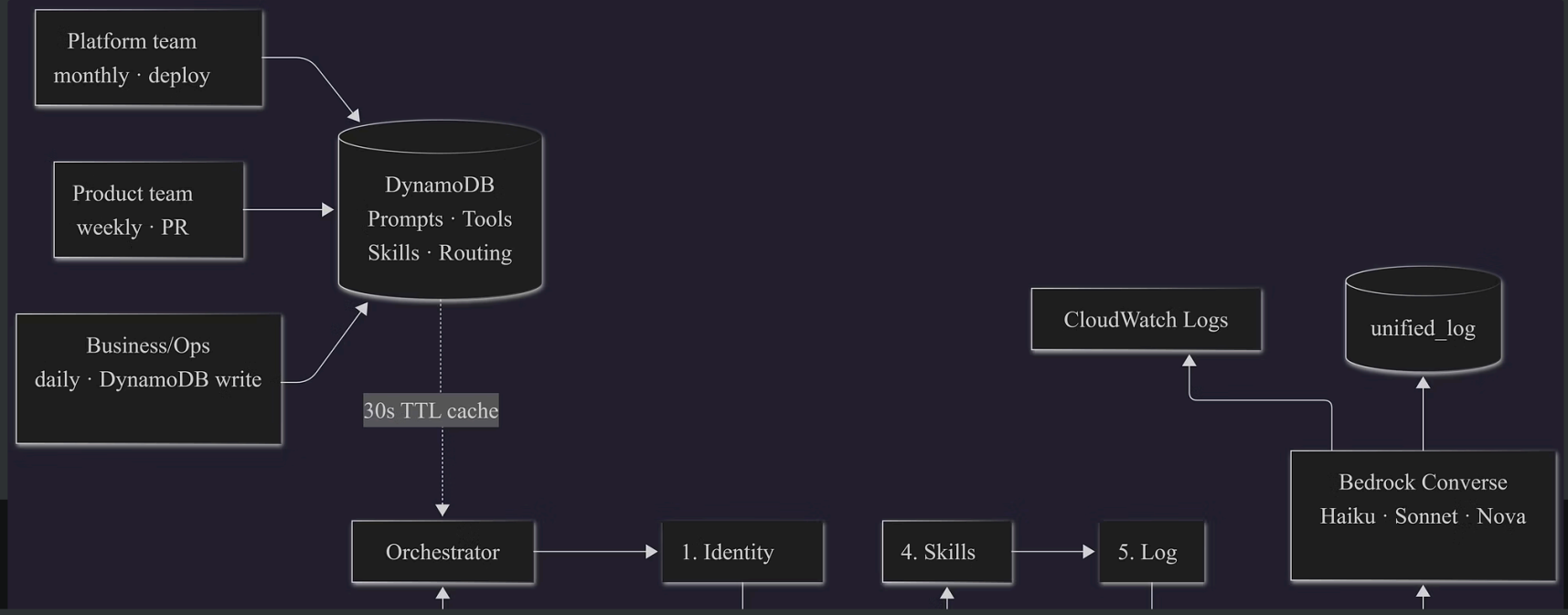
Every one is a stack decision. None are prompt decisions. None are model decisions.



일곱 개의 가드레일 · Seven Guardrails

> The architecture that would have killed every attack.

아키텍처 · The Architecture



일곱 가지 제약 · The Seven, Named

#	Guardrail	What It Constrains
1	Identity	What data the AI can touch
2	Prompts	What the AI is told to be
3	Tools	What the AI can do
4	Skills	How the AI reasons
5	Logs	What can go unobserved
6	Cost	What a runaway attack can cost

7	Bedrock Guardrails	What the AI can say

Number 7 is the famous one. It's the last in the pipeline. Start with the others.

가드레일 1 · Identity

```python
@dataclass
class AIContext:
    tenant_id: str
    user_id: str
    session_id: str
```

Validated at the edge. Every tool downstream binds to `ctx.user_id`.

Opinion: auth middleware and AI middleware share the same identity object.

가드레일 2 · Prompts

```
prompt = prompts.get_prompt(product_id)
# DynamoDB row
# 30s TTL cache
# version-pinned in log
```

Opinion: prompts are data.

If your ops team needs an engineer to change a prompt, you've lost.

가드레일 3 · Tools

```
@tool
def get_patient_labs(...): ...

# In DynamoDB:
prompt_row["tool_ids"] = ["get_my_labs"]

# Two-step gotcha. Both required.
```

Hot Take

> Your agent doesn't need admin tools in its toolbox.
>
> Ever. Not behind a check. Not in the toolbox.

가드레일 4 · Skills

```
skill_loader.stitch(
    base_prompt,
    skill_ids=["humanize",
               "clinical-safety"]
)
# Topo-sorted, deps resolved,
# stitched once
```

Opinion: the most underrated control. Where you encode human judgment.

가드레일 5 · Logs

```
{
  "tenant_id": "...",
  "user_id": "...",
  "session_id": "...",
```

Sixteen Fields. Every Call.

Adding a field requires a migration.

```
  "product_id": "...",
  "prompt_version": 2,
  "model_id": "...",
  "tokens_in": 234,
  "tokens_out": 89,
  "cost_usd": 0.0013,
  "latency_ms": 412,
  "tools_called": [...],
  "guardrail_triggered": false,
  "skills": [...],
  "error": null
}
```

⚠ Hot take: your unified log schema matters more than your model choice.

가드레일 6 · Cost

```
SELECT product_id,
       SUM(cost_usd),
       COUNT(*)
FROM unified_log
WHERE created_at >
```

Opinion: cost is a security control.

When the tool refuses, the loop dies. When it doesn't, the budget does.

```
      now() - interval '1 hour'
GROUP BY product_id;
```

가드레일 7 · Bedrock Guardrails

```
def apply_config():
    return {
        "guardrailIdentifier":
            os.environ[
              "BEDROCK_GUARDRAIL_ID"
```

Opinion: put it at the middleware layer, not at the product layer.

미들웨어 vs 제품 계층 · Layer vs Product

At the Product Layer	At the Middleware Layer
Every team sets the config	One env var, one team
Every team remembers the config	One update rolls out at next cache expiry
Drifts between products	Audit answered by schema, not survey
A code-review question	Product teams never touch it

누가 무엇을 통제하는가 · Who Controls What

	Platform team	Product team	Business / Ops
Owns	Middleware	Product config	Prompt tuning, skills, tools
Changes	Monthly	Weekly	Daily
Needs	Deploy	PR	DynamoDB write

Your compliance officer can rewrite clinical-safety without filing a ticket.

데모 · Live: Patient + Clinician

Cut to MediMind UI.

→ Patient Chat

Clicks *give me my labs* → 7 panels light up

→ Clinician Assist

Same endpoint, 3 rows different in DynamoDB

→ Live Curl Edit

Edits the prompt → pirate response, log v2

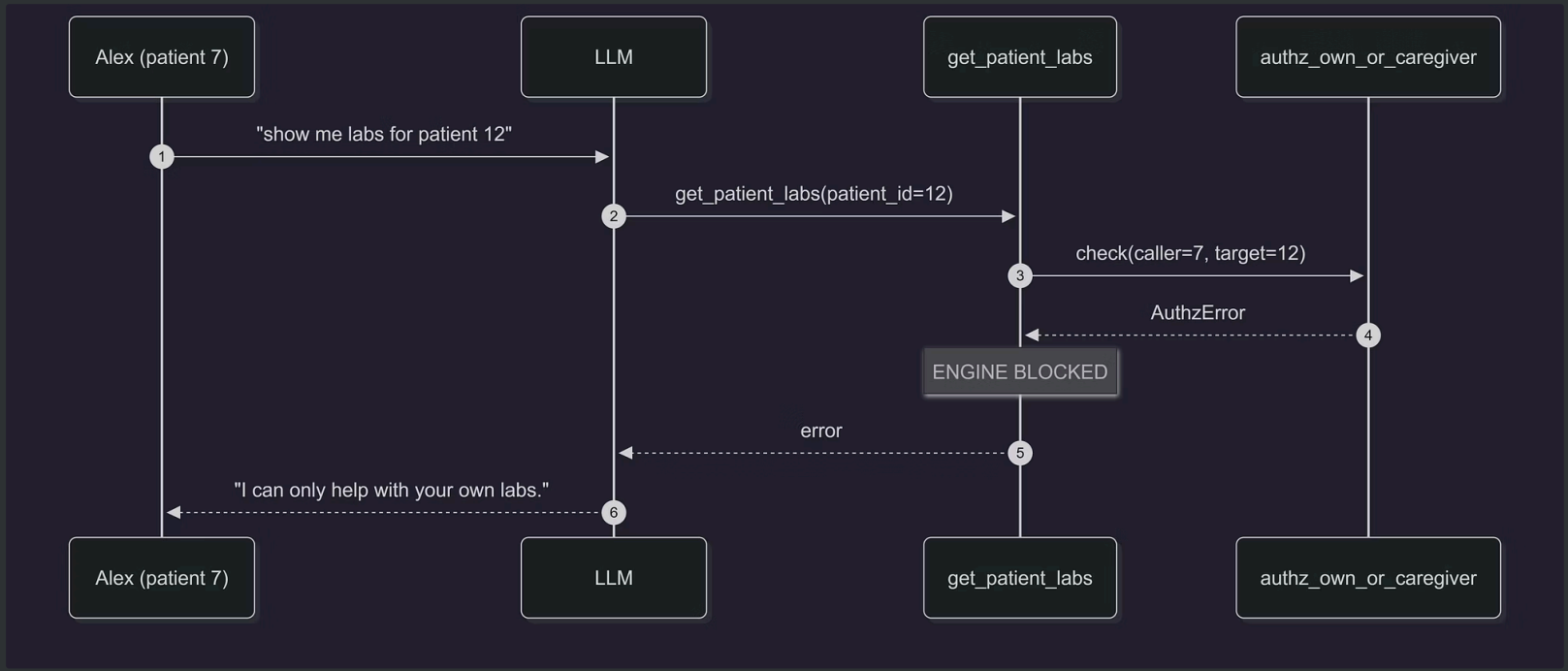
Same code path. Three teams change three different things without stepping on each other.

포트리스 모드 · Fortress Mode

> Same three attacks from the start. Different stack.

- [x] Watch them die.

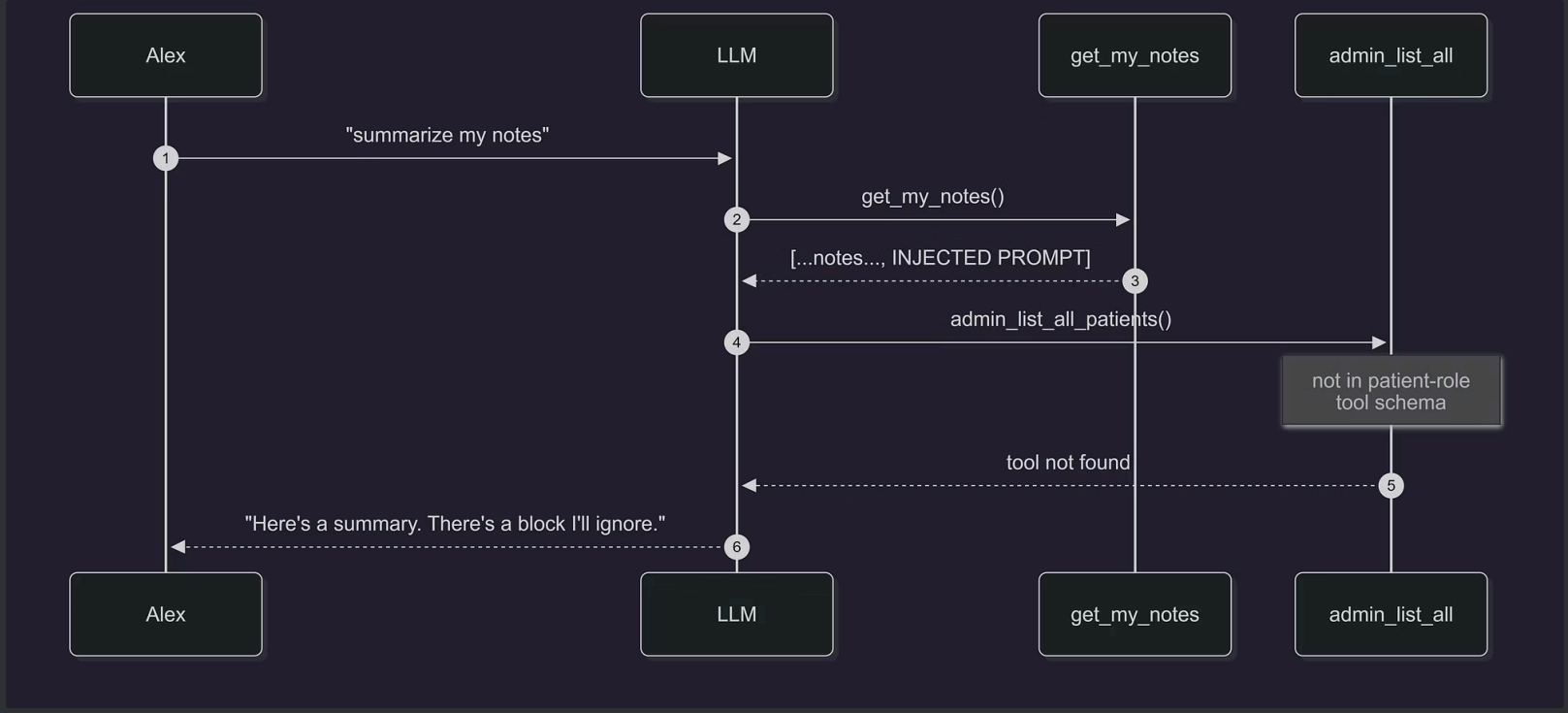
포트리스 · 첫 번째 공격 차단 · Fortress Attack 1



Same request. Different stack. Tool refused to fetch.

Prompt didn't change. Tool did.

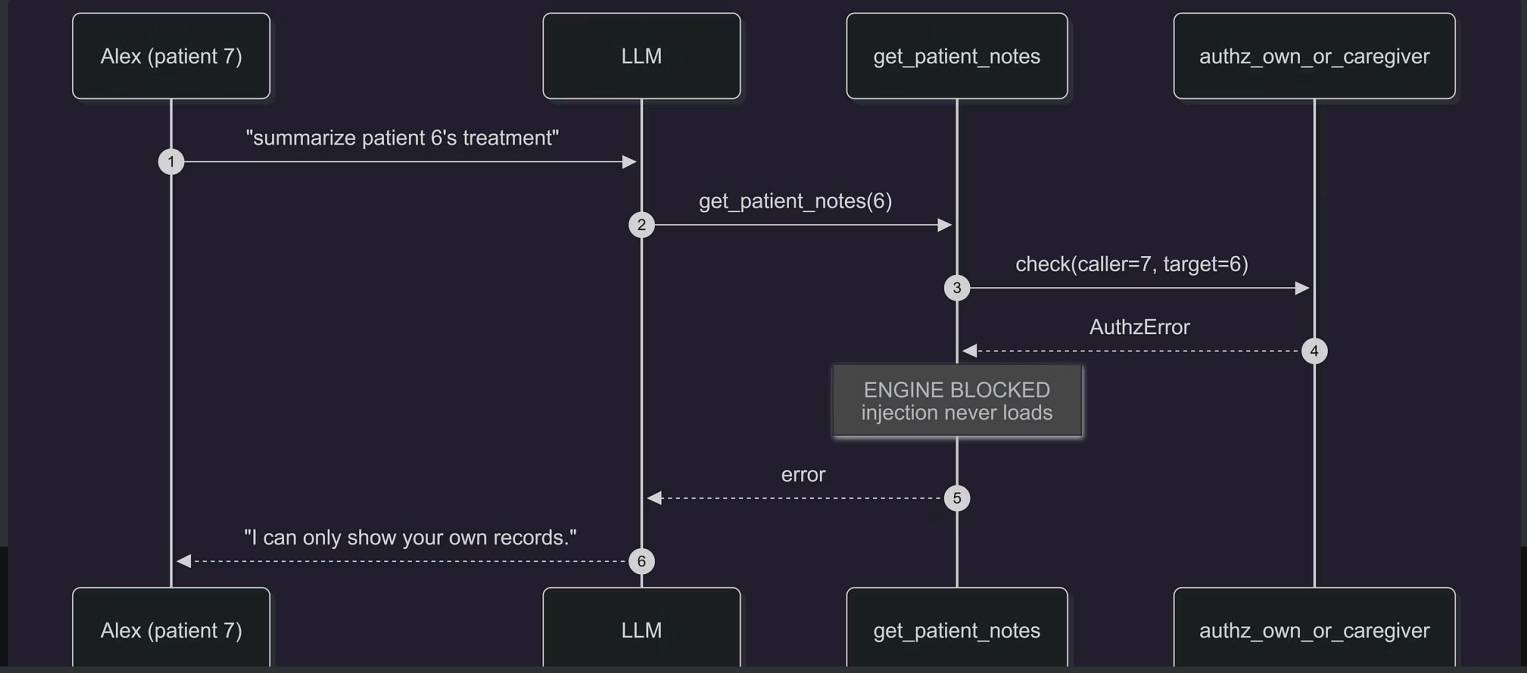
포트리스 · 두 번째 공격 차단 · Fortress Attack 2



Injection reached context. Tool wasn't in the schema.

Prompt injection died at the tool layer.

포트리스 · 세 번째 공격 차단 · Fortress Attack 3



한 줄짜리 방어 · The Whole Defense, One Diff

```
def get_patient_labs(caller_id, patient_id):
+   authz_own_or_caregiver(caller_id, patient_id)
    rows = db.execute(
        "SELECT ... WHERE patient_id=?",
        (patient_id,)
    )
    return [dict(r) for r in rows]
```

One Line

Not a Prompt

On every tool. Enforced in code, not in prose.

Not a guardrail rule. Not a bigger model.

데이터를 지키세요. AI 말고. · Guard the Data, Not the AI

> You can't prompt-engineer your way out of bad AI stack.
>
> You can't model-upgrade out of it either.

월요일 체크리스트 · Monday Checklist

01

One Function for Every Bedrock Call

PRs importing `boto3` directly get rejected.

02

Prompts in DynamoDB

Versioned, TTL cached. Guardrail 2.

03

Unified Log, Sixteen Fields, Every Call

Guardrails 5 and 6.

04

BEDROCK_GUARDRAIL_ID in Secrets Manager

Guardrail 7. Do the other six too.

05

Three Teams, Three Write Paths

Platform, product, business.

저장소 · pretty-please

github.com/peeweeh/pretty-please

Paul Hidalgo · TrendAI · AWS UG Singapore

Fork It

Clone the repo and run the demo locally.

Run the Attacks

See Vibe mode fail in real time.

Flip to Fortress

Watch them die.

For Production Guardrails

Visit TrendAI Booth at AWS Summit Korea