ScottyBites · Break In

Andrew IDs:

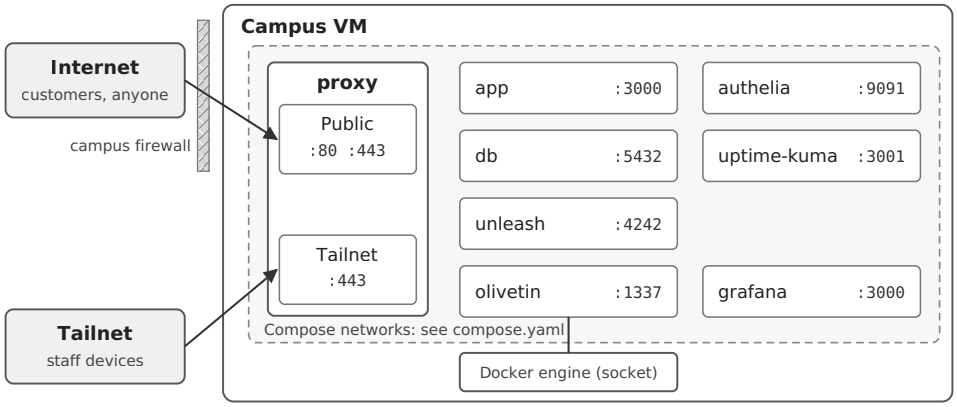
Find **one path** that gives your assigned attacker access to Grafana, OliveTin, or the database that they should not have. Use only this sheet; guessing passwords does not count.

☐ **Outsider**	☐ **Alex's laptop**	☐ **Compromised Uptime Kuma**
On the internet. Not on the tailnet; no account.	Malware on Alex's staff laptop, on the tailnet. It can use Alex's signed-in session: `alex`, in group `staff`, not `oncall`.	You can run any code inside the `uptime-kuma` container. No account.

Who should reach what

Service	Name	Who should reach it
Menu app (`app`)	`scottybites.example`	Anyone
Grafana dashboards	`grafana.scottybites.example`	Staff
OliveTin deploy and roll-back buttons	`ops.scottybites.example`	On-call staff only
Unleash feature flags	`flags.scottybites.example`	Staff; the app calls `unleash:4242` directly
Uptime Kuma status checks	`status.scottybites.example`	Staff; it checks every service is up
Database (`db`)	—	The app only

The deployment



- **Tailnet:** the team's private network; only staff devices can join. Staff devices may reach the VM on TCP 443; on-call devices also on TCP 22.
- **Authelia:** the team's single sign-on. The proxy asks it whether each request is allowed.
- **DNS:** public DNS has one record, `scottybites.example → 203.0.113.10`. Tailnet DNS resolves the internal names to `100.101.102.103`.
- **Firewall:** the campus firewall lets only TCP 80 and 443 reach the VM from the internet.
- **Certificates:** the proxy has a valid certificate for every `*.scottybites.example` name, on both listeners.
- **Login:** on the tailnet listener, the proxy asks Authelia about every request. If Authelia allows it, the proxy sets the `Remote-User` header to the signed-in username, replacing any copy the client sent, then forwards the request. The public listener forwards requests without consulting Authelia or changing client-supplied headers.
- **Grafana** signs in whichever user the `Remote-User` header names. **OliveTin** has no login of its own; its buttons restart and roll back containers through the Docker engine.
- **Uptime Kuma** stores the address and password for each service it checks.

Break In · Your Findings

Your role: ☐ Outsider ☐ Alex's laptop ☐ Compromised Uptime Kuma

`compose.yaml` (excerpt)

```yaml
services:
  proxy:
    ports:
      - "203.0.113.10:80:80"
      - "203.0.113.10:443:443"
      - "100.101.102.103:443:8443"
    networks: [web, auth, grafana, ops, flags, kuma]
  authelia:
    networks: [auth]
  app:
    networks: [web, backend, flags]
  db:
    networks: [backend]
  grafana:
    networks: [grafana]
  olivetin:
    networks: [ops]
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
  unleash:
    networks: [flags]
  uptime-kuma:
    networks: [kuma, web, backend, grafana, ops, flags]
```

Proxy routes

```
Port 80 (both): redirect to HTTPS

Public · 203.0.113.10:443
  scottybites.example         → app:3000
  grafana.scottybites.example → grafana:3000

Tailnet · 100.101.102.103:443
Every request is checked by Authelia
  grafana.scottybites.example → grafana:3000
  ops.scottybites.example     → olivetin:1337
  flags.scottybites.example   → unleash:4242
  status.scottybites.example  → uptime-kuma:3001
```

Login rules (Authelia)

```
Checked top to bottom; anything unmatched is denied
  *.scottybites.example → anyone in group staff
```

1. How do you get access you should not have?

Show where you start, how you reach the service, and what you can do.

2. What would you change to stop that path?

Propose one change that keeps the service available to the people and services that need it.

Optional: If a path you tried failed, what stopped it?

If you finish early, look for another path.