

分散式智慧校園路燈蜜罐系統

Distributed Smart Campus Streetlight Honeypot System

指導教授：陳奇業、李南逸教授 專題成員：周昀軒

Demo 影片



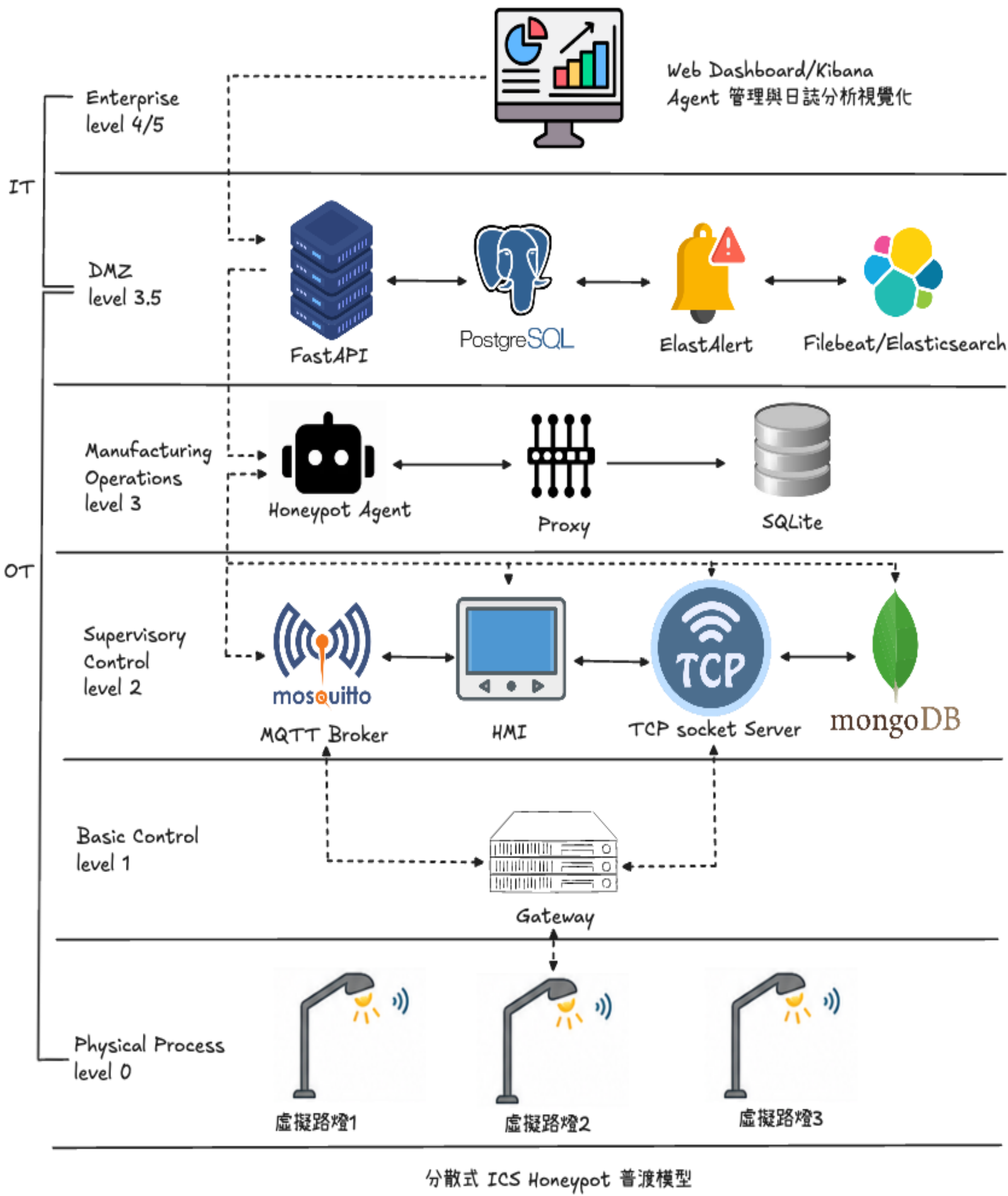
專題動機與簡介：

在 ICS 場域中，PLC、感測設備、通訊協定與後端服務之間通常存在多節點互動。然而現有許多 Honeypot 仍多著重於單一設備或單一通訊服務，用以吸引攻擊者並蒐集攻擊流量。本專題實作一套分散式 ICS Honeypot 平台，可在多個環境中部署 Honeypot Agent，並透過中央 Server 統一管理各 Agent 的 Honeypot 服務部署、狀態監控與攻擊流量蒐集分析，同時 Honeypot 服務之間能相互溝通，使整體行為更加貼近真實場域並比其他 Honeypot 更具部署彈性。本專題以成功大學智慧路燈系統作為模擬情境，建構路燈系統架構以及主要溝通流程，並蒐集實際互動與攻擊流量

系統技術與環境：

本專案主要以 Python 建置分散式 ICS Honeypot 路燈模擬系統環境，後端採用 FastAPI 作為中央控制伺服器，負責蜜罐 Agent 管理、部署設定、攻擊流量儀表板展示，並將攻擊日誌存入 PostgreSQL，同時透過 Filebeat、Elasticsearch、Kibana 與 ElastAlert 建立日誌分析與告警機制。蜜罐 Agent 部署於各節點，負責接收 Server 指令，並透過 Docker 部署 MQTT、HTTP、TCP Socket 等工控服務模擬協議與服務。系統亦利用 Proxy 攔截、轉送與記錄攻擊流量，將資料暫存於本地 SQLite，並定時回傳服務狀態與攻擊日誌至 Server，此外系統可依據模擬服務設計誘導攻擊的弱點情境，例如弱密碼、未驗證登入等，以觀察攻擊者行為與攻擊手法。

Honeypot 專題系統架構：

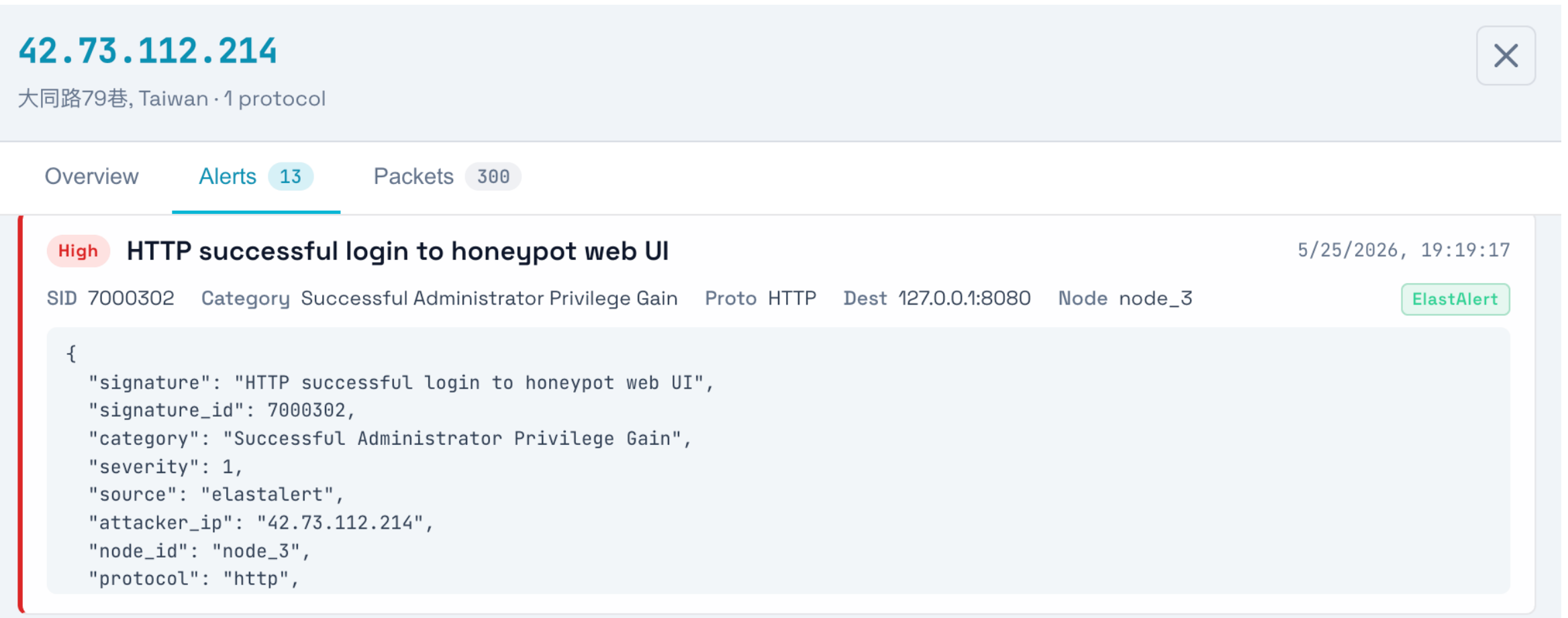


成果：

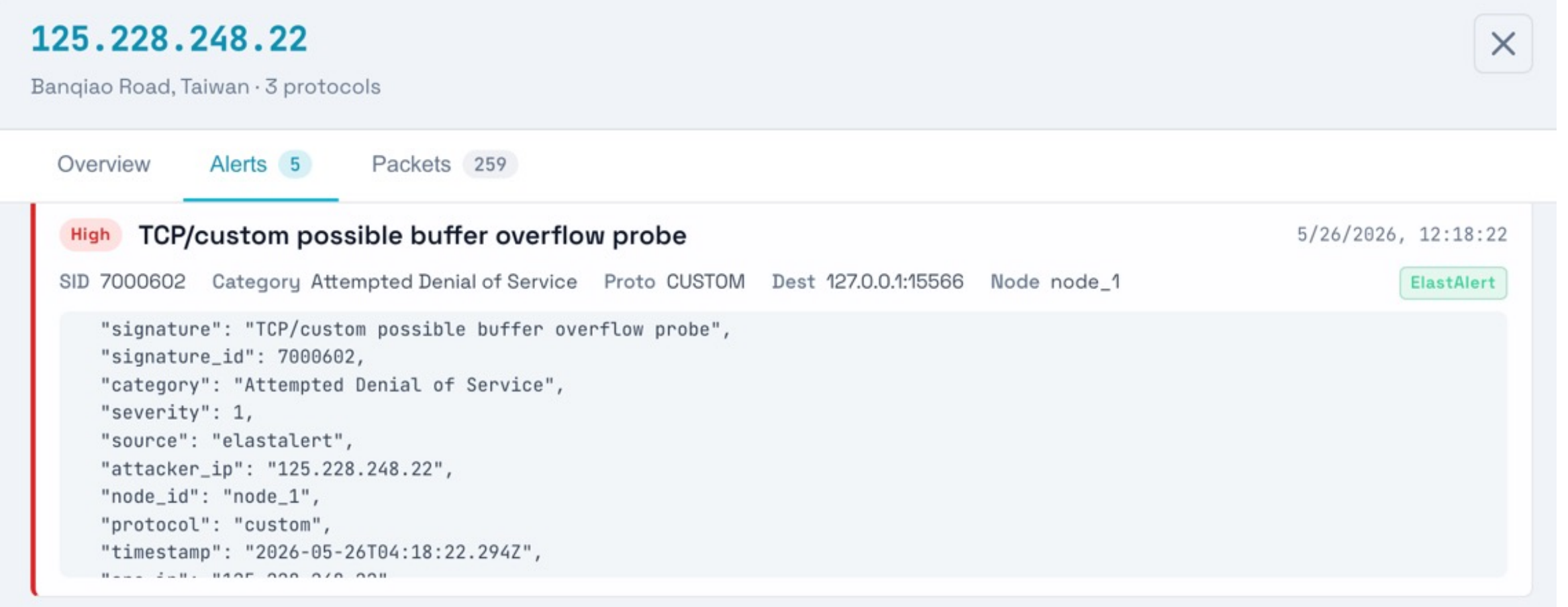
成功大學部署的虛擬路燈展示畫面



案例1：收到 HMI Honeypot 登入請求紀錄



案例2：收到針對路燈進行攻擊的警報



總結與未來展望：

本專案採分散式 ICS Honeypot 架構，擁有以下三個優勢

- 蜜罐間可互動形成蜜網
- Server 與蜜罐可部署於不同主機，方便集中管理遠端節點。
- 系統具高度彈性，可部署自製 HMI、模擬 PLC 等服務。

未來將擴充更多 ICS 協定、強化攻擊分析與告警機制。