

@startuml

sequenceDiagram

autonumber

participant Analyst as Security Analyst

participant UI as MIRS Dashboard
(Boundary)

participant Controller as Incident Controller
(Control)

participant DB as Log Repository & SIEM
(Entity)

participant TI as Threat Intel Feed
(External)

participant FW as Firewall API
(System)

Note over Analyst,FW: Detection Phase

FW->>DB: Send NetFlow/Syslogs

DB->>DB: Analyze Patterns (Signature Match)

DB->>Controller: Trigger Alert (Severity: High)

Controller->>UI: Push Notification (Alert ID: #9921)

UI->>Analyst: Display Alert Popup

Note over Analyst,FW: Investigation Phase

Analyst->>UI: Click "View Details"

UI->>Controller: GetIncidentDetails(ID)

Controller->>DB: Query L3/L4 Headers & Payload

DB-->>Controller: Return Log Data

Controller-->>UI: Show Metadata (SrcIP, DstPort)

Analyst->>UI: Request "Deep Dive" (PCAP + Reputation)

activate UI

UI->>Controller: FetchForensicData(SrcIP)

activate Controller

par Parallel Execution

Controller->>DB: Retrieve PCAP Files

DB-->>Controller: Return .pcap Blob

and

Controller->>TI: CheckReputation(SrcIP)

TI-->>Controller: Return Score (Malicious: 98/100)

end

Controller-->>UI: Display PCAP & Threat Score

deactivate Controller

deactivate UI

Note over Analyst,FW: Mitigation Phase

Analyst->>UI: Select Action: "Block Source IP"

activate UI

UI->>Controller: ExecuteMitigation(SrcIP, "Deny")

activate Controller

Note right of Controller: Logic: Map to MITRE ID
& Log Action

Controller->>FW: POST /acl/rules (Action=Deny, IP=SrcIP)

activate FW

FW->>FW: Apply Rule & Propagate

FW-->>Controller: 200 OK (Rule Created)

deactivate FW

```
Controller->>DB: Update Incident Status ("Resolved")
Controller-->>UI: Show Success Message
deactivate Controller
deactivate UI
@enduml
```