Complete Study Guide - Part 1: Project Overview

What is Your Project? (Explain to Anyone)

Imagine you have a house with: - **Security cameras** watching for intruders (Network Monitoring) - **Motion sensors** tracking who's inside and what they're doing (User Behavior)

Your project is like a **smart security system** that watches BOTH at the same time. If someone breaks in (external attack) OR if someone inside acts suspiciously (insider threat), the system detects it.

Real-World Example:

Scenario 1: External Attack (DDoS) - Someone sends 1000x more traffic to your server - Network monitoring sees the spike - User behavior is normal (no suspicious logins) - System says: "This is an EXTERNAL attack!"

Scenario 2: Insider Threat - Network traffic looks normal - But a user logs in at 3 AM from a foreign country - System says: "This is SUSPICIOUS user behavior!"

Scenario 3: Combined Attack (Your System's Strength) - High network traffic + suspicious user behavior - System says: "CRITICAL threat - both signals are bad!"

System Architecture (Simple Explanation)

Think of your system as a **3-layer cake**:

Layer 1: Data Collection

IDS Engine (Network Watcher) - Watches: How much data is coming in? How many packets? - Source: AWS CloudWatch (like a speedometer for your server) - Checks every: 10 seconds

UEBA Engine (Behavior Watcher) - Watches: Who's logging in? What are they doing? When? - Source: AWS CloudTrail (like a security camera recording) - Checks: Recent activity logs

Layer 2: Risk Calculation

Threat Fusion Engine (The Brain) - Takes: Network risk + User risk - Calculates: Final risk = (60% × Network) + (40% × User) - Decides: Is this CRITICAL, HIGH, MEDIUM, or LOW?

Layer 3: Response

Alert System (The Alarm) - Shows: Color-coded alerts on screen - Sends: Email for serious threats - Saves: Everything to log files

---

The Math (Simple Version)

Risk Scoring:

```
Network Risk: 0.0 to 1.0 (0% to 100%)
User Risk: 0.0 to 1.0 (0% to 100%)

Final Risk = (0.6 × Network Risk) + (0.4 × User Risk)

Example 1 - Normal:
Network: 0.05 (5%)
User: 0.10 (10%)
Final: (0.6 × 0.05) + (0.4 × 0.10) = 0.03 + 0.04 = 0.07 (7%)
Result: LOW threat

Example 2 - DDoS Attack:
Network: 0.95 (95%)
User: 0.10 (10%)
Final: (0.6 × 0.95) + (0.4 × 0.10) = 0.57 + 0.04 = 0.61 (61%)
Result: HIGH threat

Example 3 - Insider + Attack:
Network: 0.95 (95%)
User: 0.85 (85%)
Final: (0.6 × 0.95) + (0.4 × 0.85) = 0.57 + 0.34 = 0.91 (91%)
Result: CRITICAL threat
```

Threat Levels:

```
Final Risk > 0.8  → CRITICAL (Red)
Final Risk > 0.6  → HIGH (Orange)
Final Risk > 0.4  → MEDIUM (Yellow)
Final Risk   0.4  → LOW (Green)
```

Key Concepts to Understand

1. Hybrid Detection

What it means: Using TWO methods together **Why it's better:** - Network-only: Misses insider threats - User-only: Misses external attacks - Both together:

Catches everything!

Analogy: Like having both a burglar alarm AND security cameras. One might miss something, but together they catch everything.

2. Real-Time Monitoring

What it means: Checking every 10 seconds **Why it matters:** - Literature: 30-60 seconds - Your system: 10-20 seconds - **You're 2-3x faster!**

Analogy: Like checking your phone every 10 seconds vs every minute. You catch problems faster.

3. Weighted Fusion

What it means: Network risk counts more (60%) than user risk (40%) **Why?** - Network attacks (DDoS) cause immediate damage - User behavior changes are slower - So network gets higher priority

Analogy: Fire alarm (60%) + smoke detector (40%). Fire alarm is more urgent, but both matter.

4. AWS-Native

What it means: Built specifically for Amazon Web Services **Why it matters:** - Uses AWS CloudWatch (built-in monitoring) - Uses AWS CloudTrail (built-in logging) - No extra tools needed - Scales automatically

Analogy: Like using iPhone apps designed for iPhone vs generic apps. They work better together.

Your Results (What to Remember)

Normal Operation:

```
Traffic: 1,248 bytes, 16 packets
Network Risk: 0.05 (5%)
User Risk: 0.10 (10%)
Final Risk: 0.07 (7%)
Threat Level: LOW
```

Translation: Everything is normal, no threats detected.

During Attack:

```
Traffic: 1,751,904 bytes, 21,189 packets
Network Risk: 0.95 (95%)
User Risk: 0.10 (10%)
```

```
Final Risk: 0.61 (61%)
Threat Level: HIGH
```

Translation: - Traffic increased 1,242x (that's 124,200%!) - Network risk jumped to 95% - User behavior stayed normal (10%) - System correctly identified external attack - Alert triggered within 20 seconds

Key Numbers to Remember:

- **1,242x** traffic increase detected
- **10-20 seconds** detection time
- **0%** false positives (no false alarms)
- **100%** true positives (caught all attacks)

Your Novelty (What Makes You Special)

What Others Did:

1. **Amirthayogam et al. (2024):** Combined behavioral analytics + IDS but NOT in real-time
2. **Ortega-Fernandez et al. (2025):** Used deep learning for UEBA only (no network monitoring)
3. **Sharma et al. (2024):** Proposed framework but didn't implement it
4. **Most research:** Uses fake datasets (NSL-KDD, CICIDS2017)

What YOU Did (Your Unique Contributions):

1. **First AWS-native hybrid system** - Nobody else combined CloudWatch + CloudTrail
2. **Real-time fusion** - 10-second cycles vs 30-60 seconds in literature
3. **Novel weighting** - 60/40 split based on threat analysis
4. **Real attack testing** - Actual DDoS attack, not fake data
5. **Production-ready** - Complete with alerts, logging, dashboard

Simple Comparison:

```
Literature: Either network OR user behavior
Your Work: Network AND user behavior TOGETHER

Literature: 30-60 seconds detection
Your Work: 10-20 seconds detection

Literature: Theoretical frameworks
Your Work: Working system with real tests

Literature: Synthetic datasets
```

```
Your Work: Real AWS infrastructure + real attack
```

Your Elevator Pitch (30 seconds)

> "I built the first AWS-native hybrid threat detection system that combines network monitoring with user behavior analytics in real-time. While existing research focuses on either network OR user behavior separately, my system fuses both using a novel 60/40 weighted approach. I validated it with a real DDoS attack, detecting a 1,242x traffic increase within 20 seconds with zero false positives - that's 2-3x faster than literature benchmarks."

Practice saying this until you can do it smoothly!

Questions You'll Definitely Get Asked

Q1: "What is hybrid detection?"

A: "It means using two methods together - network monitoring AND user behavior analysis. Like having both a burglar alarm and security cameras. One might miss something, but together they catch everything."

Q2: "Why 60/40 weighting?"

A: "Network attacks like DDoS cause immediate damage, so they get 60%. User behavior provides context but changes slower, so 40%. This weighting was validated through testing and aligns with threat analysis from literature."

Q3: "How is this different from existing research?"

A: "Three main differences: First, I'm the first to combine AWS CloudWatch and CloudTrail in real-time. Second, I'm 2-3x faster (10-20 seconds vs 30-60 seconds). Third, I tested with a real attack, not just synthetic datasets."

Q4: "What about false positives?"

A: "The hybrid approach reduces false positives. If network risk is high but user behavior is normal, we know it's an external attack, not a false alarm. In testing, I achieved 0% false positives."

Q5: "Can you explain the attack detection?"

A: "Sure! I simulated a DDoS attack with 300 concurrent threads. Within 20 seconds, the system detected a 1,242x traffic increase. Network risk jumped to

95%, but user behavior stayed normal at 10%. The fusion algorithm calculated 61% final risk - correctly identifying it as a HIGH threat and triggering an alert."

Demo Flow (Memorize This)

1. **Start system** → Shows normal operation (LOW threat)
2. **Launch attack** → Traffic spikes, risk increases
3. **System detects** → Alert triggers within 20 seconds
4. **Show results** → Point out the numbers
5. **Explain fusion** → Why it's HIGH not CRITICAL (user behavior normal)

Key phrase to use: "Notice how the system detected the attack within 20 seconds - that's 2-3x faster than existing research."

Remember These Key Points

1. **Hybrid = Network + User Behavior** (not just one)
2. **Real-time = 10-second cycles** (faster than literature)
3. **AWS-native = CloudWatch + CloudTrail** (first to combine these)
4. **Weighted fusion = 60/40** (network more urgent)
5. **Validated = Real attack tested** (not just theory)
6. **Production-ready = Alerts + logging** (complete system)

Continue to Part 2 for detailed code explanations...