🧱 PROJECT: ELK-Based APT Detection System

🎯 Objective Recap

Build an end-to-end cybersecurity monitoring pipeline using ELK Stack that:

* Ingests and parses logs (from simulated attacks)
* Applies detection rules for APT techniques
* Visualizes alerts and attack stages in Kibana
* Shows a working demo for SIH

🧩 Team Roles Breakdown

Here’s how to split the work for 6 members (assuming all beginners).  
Each person owns a clear module but collaborates through integration checkpoints.

| Member | Role / Module | Key Responsibilities | Difficulty | Approx. Time |
| --- | --- | --- | --- | --- |
| 👩‍💻 Member 1 – Environment & Setup Lead | *ELK Stack Setup & Configuration* | - Install & configure Elasticsearch, Logstash, Kibana via Docker or Elastic Cloud - Verify connections between them - Document setup steps for team - Basic CLI, config editing | ⭐⭐ Moderate | ~10–12 hrs |
| 👨‍💻 Member 2 – Data & Log Integration Engineer | *Log Collection & Ingestion* | - Configure Filebeat / Winlogbeat to forward logs - Prepare sample logs (e.g., CICIDS / Cowrie) - Verify logs reaching Logstash - Test pipelines with sample data | ⭐⭐ Moderate | ~10–12 hrs |
| 👩‍💻 Member 3 – Logstash Pipeline & Parser Developer | *Log Parsing & Normalization* | - Write Logstash .conf pipeline files - Apply filters (grok, geoip, json) - Create index templates for Elasticsearch - Test for correct field mappings | ⭐⭐⭐ Intermediate | ~12–14 hrs |
| 👨‍💻 Member 4 – Detection Rules & APT Mapping Specialist | *Rule Creation & MITRE Mapping* | - Study Sigma / Elastic rule formats - Create JSON-based rules for PowerShell, credential dumping, etc. - Map them to MITRE ATT&CK (T1059, T1003) - Test alerts in Kibana | ⭐⭐⭐⭐ Challenging | ~14–16 hrs |
| 👩‍💻 Member 5 – Dashboard & Visualization Designer | *Kibana Visualization & Dashboard* | - Create visual dashboards (bar charts, maps, timelines) - Design an APT Kill Chain visualization - Configure alert triggers (email/webhook) - Make it presentation-ready | ⭐⭐ Moderate | ~10–12 hrs |
| 👨‍💻 Member 6 – Documentation & Presentation Owner | *Docs, Demo, and Coordination* | - Prepare system architecture diagrams - Write README, report, and SIH PPT - Record demo (screen + explanation) - Coordinate integration testing | ⭐⭐ Easy/Moderate | ~10 hrs |

🧠 Workload Summary Table

| Type of Work | Who | % Workload | Complexity | Time (hrs) |
| --- | --- | --- | --- | --- |
| Environment Setup | M1 | 15% | ⭐⭐ | 10–12 |
| Log Ingestion | M2 | 15% | ⭐⭐ | 10–12 |
| Log Parsing | M3 | 20% | ⭐⭐⭐ | 12–14 |
| Rule Creation | M4 | 25% | ⭐⭐⭐⭐ | 14–16 |
| Visualization | M5 | 15% | ⭐⭐ | 10–12 |
| Documentation | M6 | 10% | ⭐⭐ | 10 |

✅ Total team hours (simultaneous work):  
~10–15 hrs each × 6 members = ~70–90 person-hours over 4–5 days  
👉 Equivalent to 4–5 full working days (8 hrs/day) if everyone contributes in parallel.

That’s enough to reach 70–80% (working prototype + live demo).

🔄 Suggested 5-Day Sprint Plan (for 80% Completion)

| Day | Team Focus | Key Outputs |
| --- | --- | --- |
| Day 1 | 🔧 Setup & Planning | ELK installed via Docker/Cloud + datasets chosen (CICIDS + Cowrie). All configs initialized. |
| Day 2 | 📡 Log Ingestion | Filebeat + Logstash working; verify data visible in Elasticsearch (index created). |
| Day 3 | 🧠 Parsing & Rules | Logstash filters + detection rules (PowerShell, Credential Dumping). Alerts firing in Kibana. |
| Day 4 | 📊 Visualization & Demo Prep | Dashboards built, rules mapped to MITRE ATT&CK, sample attacks simulated via Atomic Red Team. |
| Day 5 | 📝 Integration & Presentation | Demo run, screenshots, PPT finalized, video recorded. |

📦 Collaboration Plan

| Tool | Use |
| --- | --- |
| 🧩 GitHub Repo | Store config files, rules, docs |
| 💬 Google Chat / Discord | Daily coordination |
| 🧾 Google Sheets | Track task progress |
| 🖥️ VS Code / Elastic Cloud | Development environment |
| 📽️ OBS / ScreenPal | Demo recording |

⚙️ Example Deliverables by Each Member

| Member | Deliverable Example |
| --- | --- |
| M1 | docker-compose.yml, Elastic config screenshots |
| M2 | filebeat.yml, sample logs, ingestion proof |
| M3 | logstash.conf, pipeline success screenshots |
| M4 | rules/detection\_rules.json, MITRE mapping table |
| M5 | dashboard.png, alert screenshot, alert\_demo.mp4 |
| M6 | SIH\_Presentation.pptx, README.md, system diagram |

🧭 Integration Checkpoints

| Time | Integration Goal |
| --- | --- |
| Mid-Day 2 | All logs visible in Elasticsearch |
| End of Day 3 | Detection rules firing successfully |
| Day 4 morning | Dashboard ready + alerts mapped |
| Day 5 | Demo + PPT ready |

🔥 Tips to Ensure Smooth Workflow

* Pair each beginner with a slightly faster learner for shared modules.
* Keep ELK running in Docker (so setup is easy & consistent).
* Use sample logs first, then move to APT simulations later.
* Focus on making the pipeline work rather than perfection.
* Each member records a 1–2 min walkthrough of their task for internal review.

✅ Final Outcome After 5 Days

By end of day 5, your team will have:

* ✅ Working ELK setup
* ✅ Real log ingestion (CICIDS or Cowrie)
* ✅ Detection rules triggering in Kibana
* ✅ Dashboards visualizing attack patterns
* ✅ Demo video + PPT ready for SIH presentation

That’s 80% of the project complete — perfect for finals!

🗓️ Day 1 – Setup, Dataset Prep, and Planning

🎯 Goal of the Day:  
Set up the ELK environment, gather APT log datasets, and get everyone’s workstation ready.  
By end of the day, everyone should have a working Elastic Stack and basic understanding of their part.

👩‍💻 Member 1 – Environment & Setup Lead

Role: Set up and configure ELK Stack for the whole team

🧩 Tasks

1. Install ELK using Docker (recommended for beginners):
   * Follow official guide: <https://www.elastic.co/guide/en/elastic-stack-get-started/current/get-started-docker.html>
   * Install Docker Desktop: https://www.docker.com/products/docker-desktop/
   * Run ElasticSearch + Kibana containers
2. Verify access:
   * Test Elasticsearch: <http://localhost:9200>
   * Test Kibana: <http://localhost:5601>
3. Take screenshots once it runs successfully.
4. Document setup commands in a text/markdown file (to share with others).

🕐 Time Required: ~5–6 hours  
📘 Learning Resource:

* [ELK Stack Beginner Setup Video](https://www.youtube.com/watch?v=lBfshkPlMW8)

👨‍💻 Member 2 – Data & Log Integration Engineer

Role: Collect and prepare sample log datasets

🧩 Tasks

1. Download sample cybersecurity log datasets:
   * CICIDS 2017 dataset → [https://www.unb.ca/cic/datasets/ids-2017.html](https://www.unb.ca/cic/datasets/ids-2017.html?utm_source=chatgpt.com)
   * Cowrie honeypot logs → [https://github.com/cowrie/cowrie](https://github.com/cowrie/cowrie?utm_source=chatgpt.com)
   * APT simulation logs (APT1 sample) → https://attack.mitre.org/resources/
2. Clean and prepare a few .log files for testing.
3. Create a folder structure like:
4. /logs/
5. ├── apache/
6. ├── network/
7. ├── APT/
8. Share sample logs with Member 3 for ingestion testing.

🕐 Time Required: 5 hours  
📘 Learning Resource: Kaggle Security Log Datasets

👩‍💻 Member 3 – Logstash Pipeline & Parser Developer

Role: Learn Logstash and prepare to handle incoming logs

🧩 Tasks

1. Learn Logstash basics →
   * [Official Docs](https://www.elastic.co/guide/en/logstash/current/index.html)
   * [Video Tutorial](https://www.youtube.com/watch?v=tuPzI8wKl4A)
2. Install Logstash (if not using Docker):
   * <https://www.elastic.co/downloads/logstash>
3. Write a simple Logstash config to test:
4. input { file { path => "/logs/test.log" start\_position => "beginning" } }
5. output { stdout { codec => rubydebug } }
6. Run and verify logs are being read.
7. Document what worked and what didn’t.

🕐 Time Required: 6 hours  
📘 Learning Resource: [Logstash Configuration Guide (Elastic)](https://www.elastic.co/guide/en/logstash/current/configuration.html)

👨‍💻 Member 4 – Detection Rules & APT Mapping Specialist

Role: Understand APT attacks and how rules are built

🧩 Tasks

1. Learn what APT (Advanced Persistent Threat) means:
   * [MITRE ATT&CK Framework Overview](https://attack.mitre.org/?utm_source=chatgpt.com)
   * [APT Lifecycle Explained (YouTube)](https://www.youtube.com/watch?v=yg1a7xJpS-8)
2. Study how Elastic Detection Rules are written:
   * [Elastic Security Rules](https://www.elastic.co/guide/en/security/current/prebuilt-rules.html)
3. Collect 3–4 example Sigma rules:
   * [Sigma Rule Repository](https://github.com/SigmaHQ/sigma?utm_source=chatgpt.com)
4. Make a list of APT behaviors you want to detect tomorrow (e.g., PowerShell execution, credential dumping).

🕐 Time Required: 6–7 hours  
📘 Learning Resource: [Elastic Detection Rule Format](https://www.elastic.co/guide/en/security/current/rules-ui-management.html)

👩‍💻 Member 5 – Dashboard & Visualization Designer

Role: Explore Kibana dashboards

🧩 Tasks

1. Learn how to use Kibana dashboards:
   * [Kibana Guide](https://www.elastic.co/guide/en/kibana/current/dashboard.html)
   * [Dashboard Creation Tutorial](https://www.youtube.com/watch?v=kX09mJcPBo8)
2. Prepare wireframe sketches for tomorrow:
   * Draw how your dashboard will look:
     + Chart for number of attacks detected
     + Table of active alerts
     + Map of affected hosts
3. Document your ideas for the visualization part.

🕐 Time Required: 5 hours  
📘 Tool for Sketching: Excalidraw or Figma

👨‍💻 Member 6 – Documentation & Coordination

Role: Keep everything organized and ready for SIH presentation

🧩 Tasks

1. Create a shared Google Drive folder:
2. /ELK-APT-Project/
3. ├── Docs/
4. ├── Logs/
5. ├── Configs/
6. ├── Screenshots/
7. Create a Google Doc or Notion page summarizing today’s setup progress.
8. Create SIH PPT outline (you’ll fill details later).
9. Collect screenshots from other members (ELK, logs, configs).
10. Summarize progress by end of the day for team sync.

🕐 Time Required: 4–5 hours  
📘 Template for PPT: SIH Presentation Template (Google Slides)

🧭 Team Checkpoints for Today

✅ 1. ELK stack runs locally (Member 1)  
✅ 2. Sample logs ready (Member 2)  
✅ 3. Logstash tested (Member 3)  
✅ 4. Basic understanding of APT & rules (Member 4)  
✅ 5. Dashboard layout sketched (Member 5)  
✅ 6. Docs organized (Member 6)

📅 End-of-Day Deliverable:  
Each member uploads screenshots + notes + configs to the shared drive.