

Lab 3 - Event-Driven Cybersecurity Pipeline

Yuri Matiyash, Guy Dazanashvili, Bar Sberro, Sagi Pichon

Pipeline Execution

Here are the screenshots from running the pipeline locally with docker

Figure 1: Redpanda console showing the synthetic Windows events being pushed into the raw-events Kafka queue by the producer script.

TIMESTAMP	KEY	VALUE
> 2/20/2026, 7:49:09 PM	∅ Null	{"event_id": "0e21590a-de4e-4e04-92f9-4dead8c99b26", "timestamp": "2026-02-20T17:49:09.525948+00:00", "user": "alice", "host": "... JSON - 309 B"}
> 2/20/2026, 7:49:12 PM	∅ Null	{"event_id": "caf2e143-cf26-4484-b3f4-4b48bb980c39", "timestamp": "2026-02-20T17:49:12.481764+00:00", "user": "alice", "host": "... JSON - 283 B"}
> 2/20/2026, 7:49:14 PM	∅ Null	{"event_id": "1c729fc0-0407-4c76-a40c-6c1165fc3fa5", "timestamp": "2026-02-20T17:49:14.849608+00:00", "user": "charlie", "host": "... JSON - 311 B"}
> 2/20/2026, 7:49:16 PM	∅ Null	{"event_id": "a6cdb613-c0d8-46cb-9ce1-3089b479c607", "timestamp": "2026-02-20T17:49:16.226165+00:00", "user": "charlie", "host": "... JSON - 285 B"}
> 2/20/2026, 7:49:19 PM	∅ Null	{"event_id": "5d46fdae-8c1c-487a-92d5-a4e6502e1a83", "timestamp": "2026-02-20T17:49:19.056984+00:00", "user": "admin", "host": "... JSON - 311 B"}
> 2/20/2026, 7:49:20 PM	∅ Null	{"event_id": "84ef5246-3e9e-4115-bba4-a1178569e0e8", "timestamp": "2026-02-20T17:49:20.336163+00:00", "user": "bob", "host": "... JSON - 334 B"}
> 2/20/2026, 7:49:21 PM	∅ Null	{"event_id": "53255b6e-71a0-4c13-999e-338881125481", "timestamp": "2026-02-20T17:49:21.591101+00:00", "user": "charlie", "host": "... JSON - 285 B"}
> 2/20/2026, 7:49:26 PM	∅ Null	{"event_id": "0f3408b9-1925-4d95-8e02-9cf278b66fc4", "timestamp": "2026-02-20T17:49:26.525680+00:00", "user": "bob", "host": "... JSON - 334 B"}
> 2/20/2026, 7:49:29 PM	∅ Null	{"event_id": "6ba989e9-6d7e-4f3a-9253-ffd765a9a851", "timestamp": "2026-02-20T17:49:29.253935+00:00", "user": "admin", "host": "... JSON - 283 B"}
> 2/20/2026, 7:49:33 PM	∅ Null	{"event_id": "82e80e9d-5f88-4ea4-8fb8-8b5263c90829", "timestamp": "2026-02-20T17:49:33.494756+00:00", "user": "alice", "host": "... JSON - 336 B"}

Figure 2: Jaeger UI showing the distributed tracing. The expanded view shows the full timeline of a single event, from being produced to being consumed, classified, and written to the CSV.

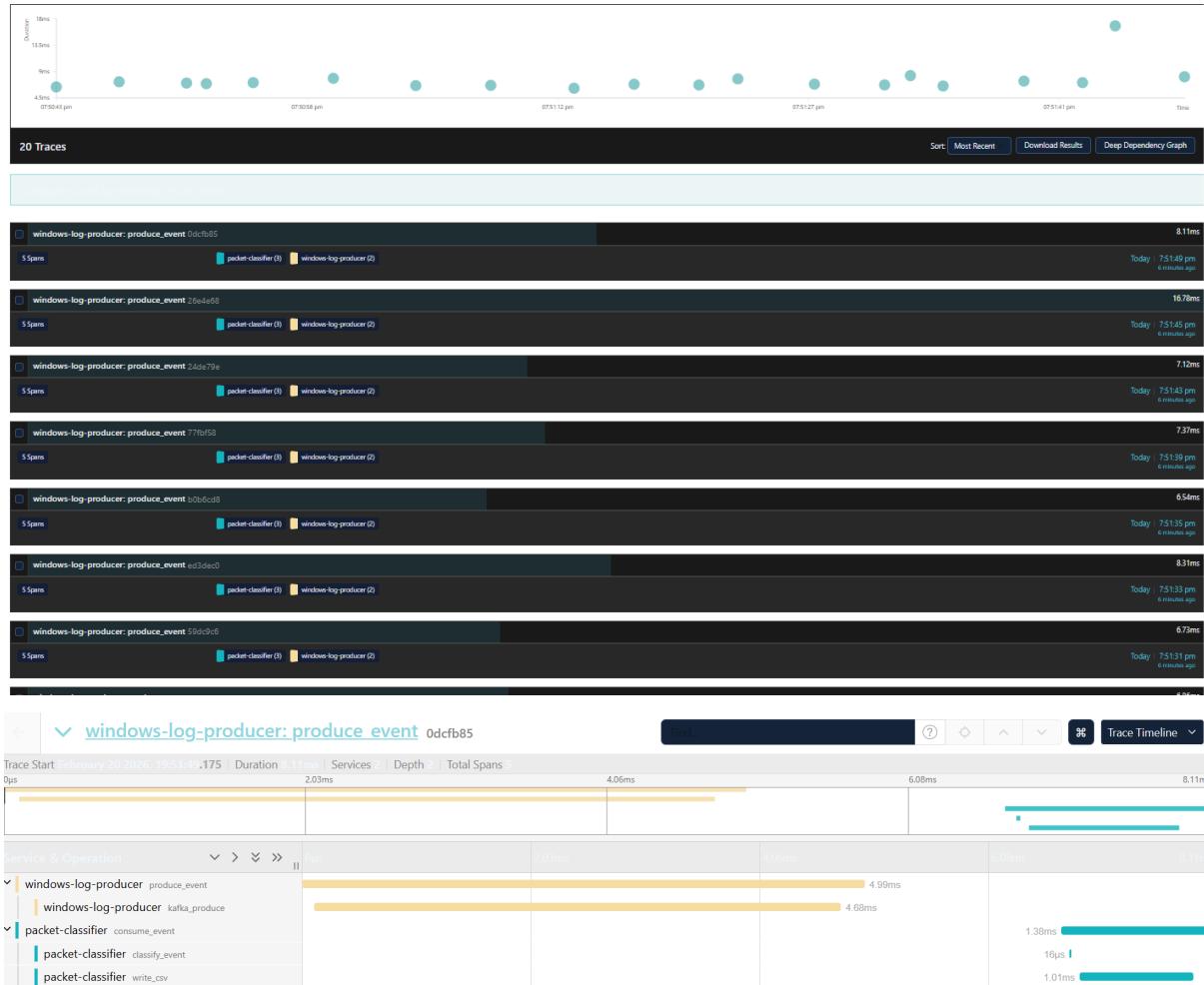
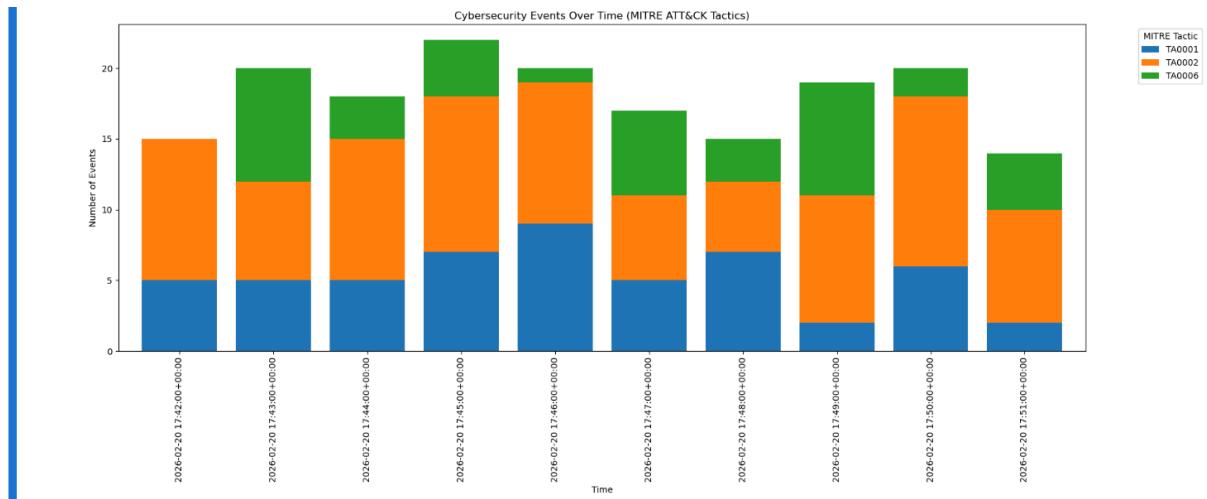


Figure 3: The output from the statistics notebook. It reads the final CSV and groups the assigned MITRE ATT&CK tactics by minute.



Conceptual Questions

- 1. Why is Kafka used instead of direct function calls?** If we just used normal synchronous function calls, a sudden spike in security events would probably crash the classifier or cause us to lose data. Kafka acts like a buffer. The producer just throws the logs into the queue and doesn't have to wait for the consumer to finish, so nothing gets lost if the system gets busy.
- 2. What happens if the consumer is slower than the producer?** The system doesn't crash. The extra events just sit safely in the Kafka queue waiting their turn. The consumer keeps processing them as fast as it can. It means the detection isn't exactly real-time anymore while it catches up, but at least we don't drop any important logs.
- 3. How does tracing help debug pipeline behavior?** It lets us see exactly how long an event takes to get through the whole system. If the pipeline is lagging, we can look at the trace in Jaeger and see exactly which part is slow—like if it's the classification function taking too long or just the disk lagging when writing to the CSV file.
- 4. Which pipeline stages could be scaled independently?** Definitely the consumer/classifier part. Since the events are just sitting in Kafka, if the processing is too slow, we can just spin up more consumers at the same time to pull from the same queue and split the workload.
- 5. How would this pipeline change in a real SOC system?** This lab is a pretty basic simulation. In a real SOC, the producer wouldn't be a Python script making fake data; it would be actual logs coming from firewalls or EDR agents. The classifier would also probably use actual machine learning models instead of basic if/else rules to map the threats. Finally, instead of saving everything to a local CSV, it would dump the data into something like Splunk or Elasticsearch so analysts could actually search it and build live dashboards.