Lab – Mini SIEM

Instructor: Tesfaye W. Lemma

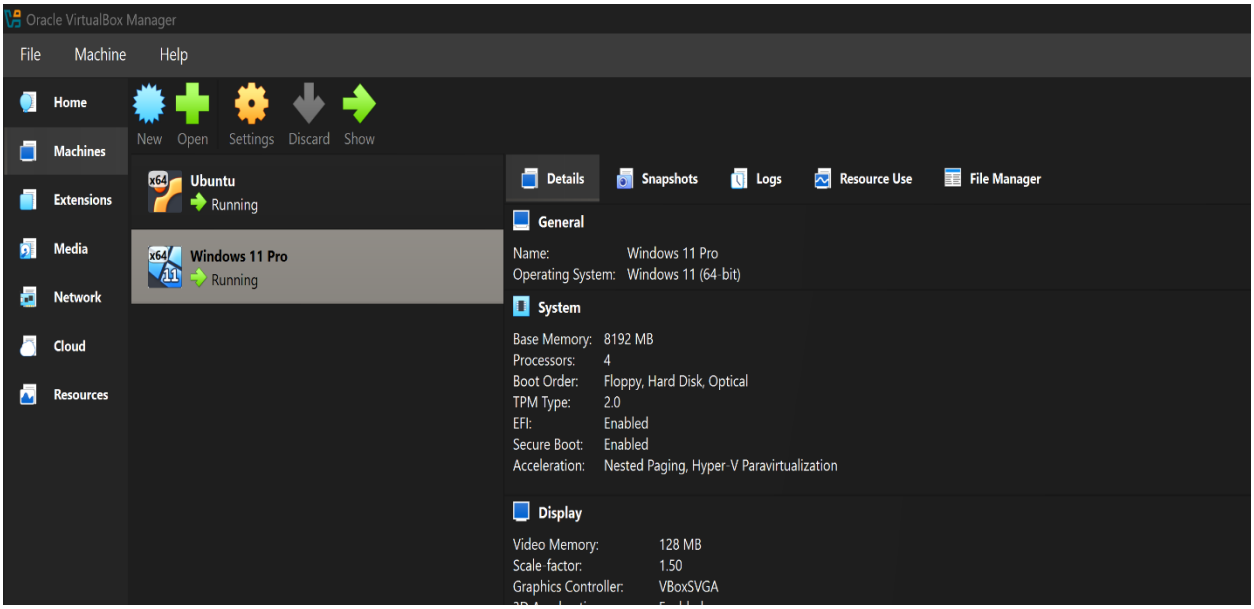
Mini-SIEM Lab: Windows Log Collection and Detection Using the Elastic Stack on VirtualBox

This project consists of implementing a mini-SIEM in a controlled lab environment using two virtual machines in Oracle VM VirtualBox. An Ubuntu virtual machine was configured as the monitoring server running Elasticsearch and Kibana, while a Windows 11 Pro virtual machine acted as the endpoint generating Windows Security logs. The primary objective was to validate the ability to collect Windows logs, centralize them in a search and analytics backend, visualize activity through dashboards, and create basic detections for security-relevant events such as user logons and privilege escalation. Basic instructions were provided in class by the instructor, and the remaining configuration and troubleshooting were completed independently.

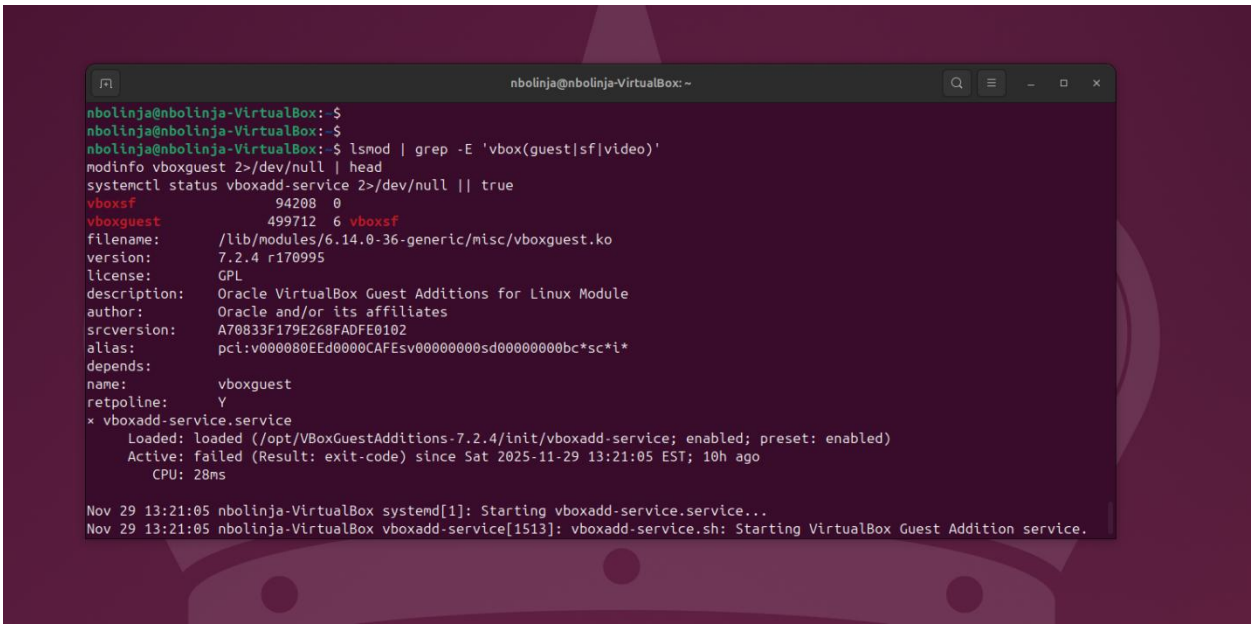
Throughout the project, real-world challenges related to service configuration, virtual networking, and hardware limitations were encountered and resolved, closely reflecting scenarios commonly faced in SOC and security operations environments.

Ubuntu and Windows 11 Pro on Oracle VM Virtual Box

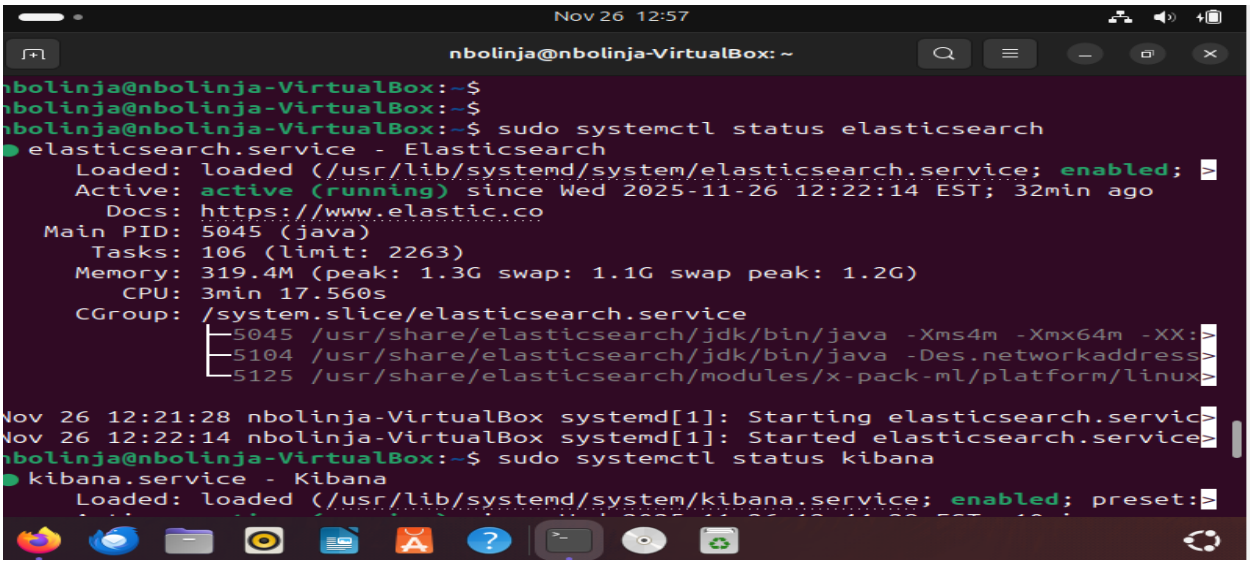
Ubuntu VM and Windows 11 Pro VM in Oracle VM VirtualBox; used to build a mini-SIEM for detection and log monitoring



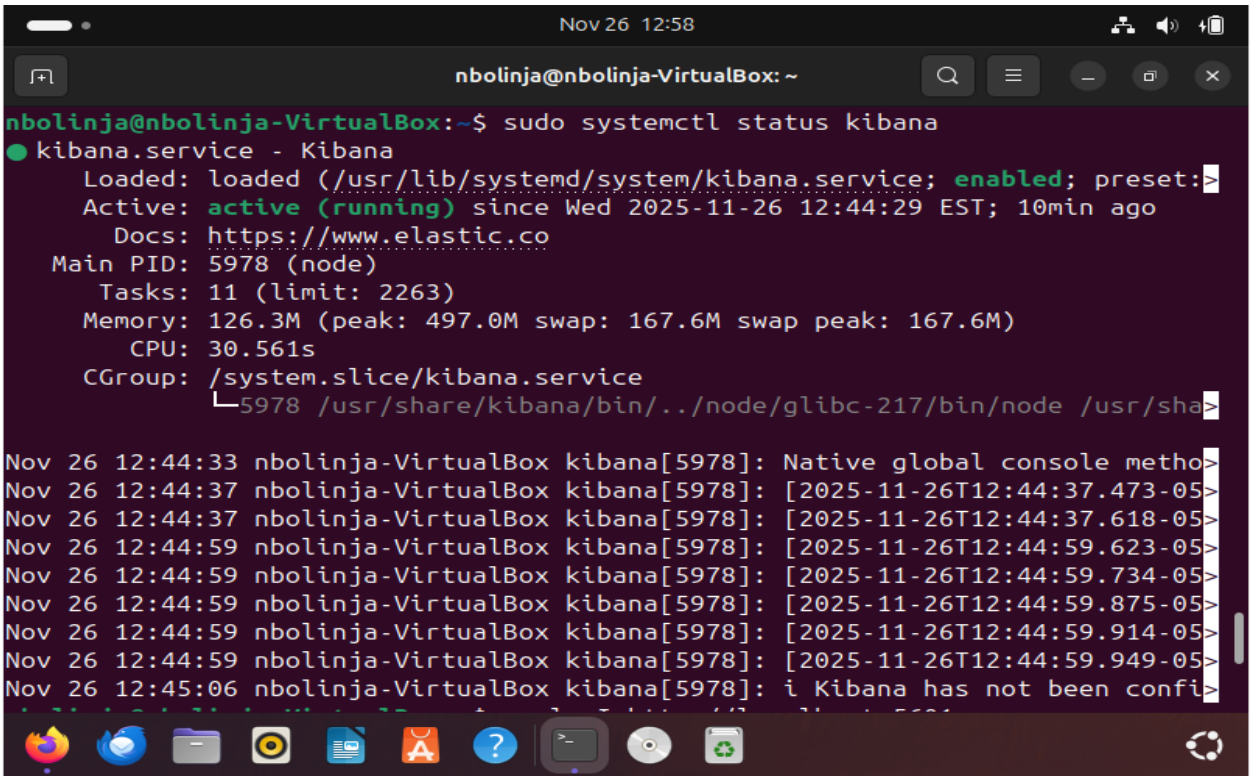
Guest Additions: Installed on ubuntu.



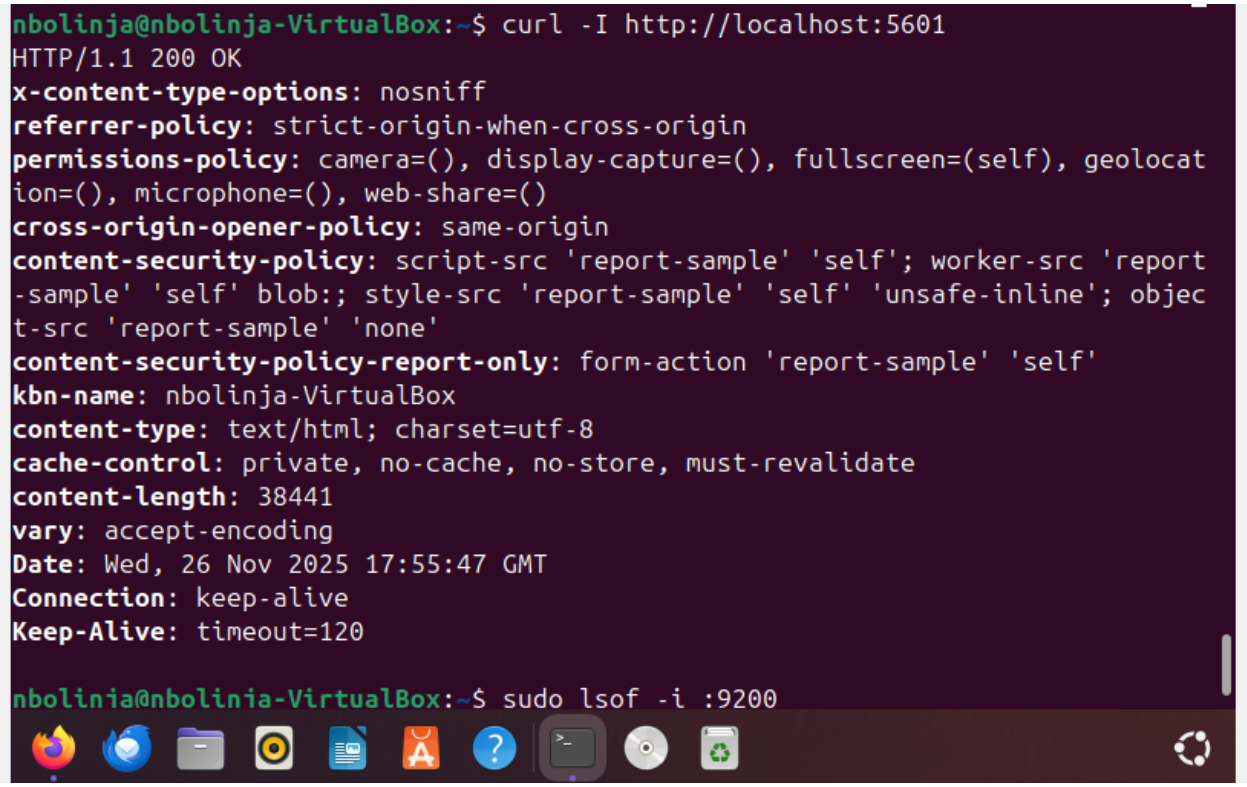
Elastisearch: service is installed and running (active; running) on Ubuntu.



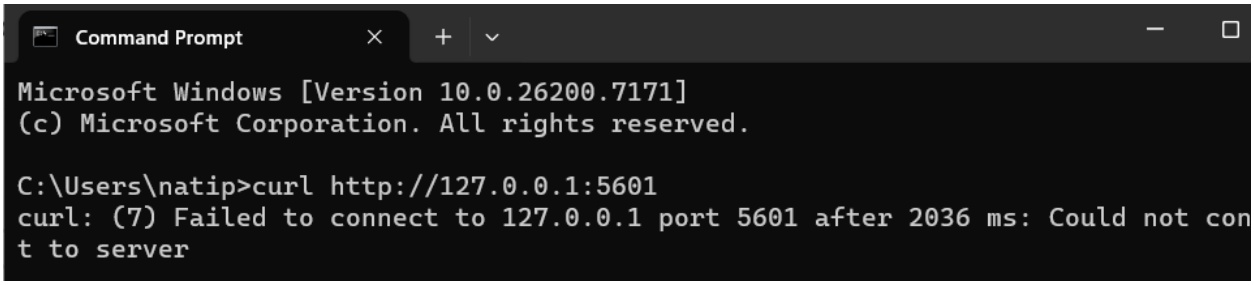
Kibana validation: service is running (active; running) on Ubuntu.



Kibana local reachability: localhost:5601 responds successfully; confirms the UI endpoint is up on the Ubuntu VM.

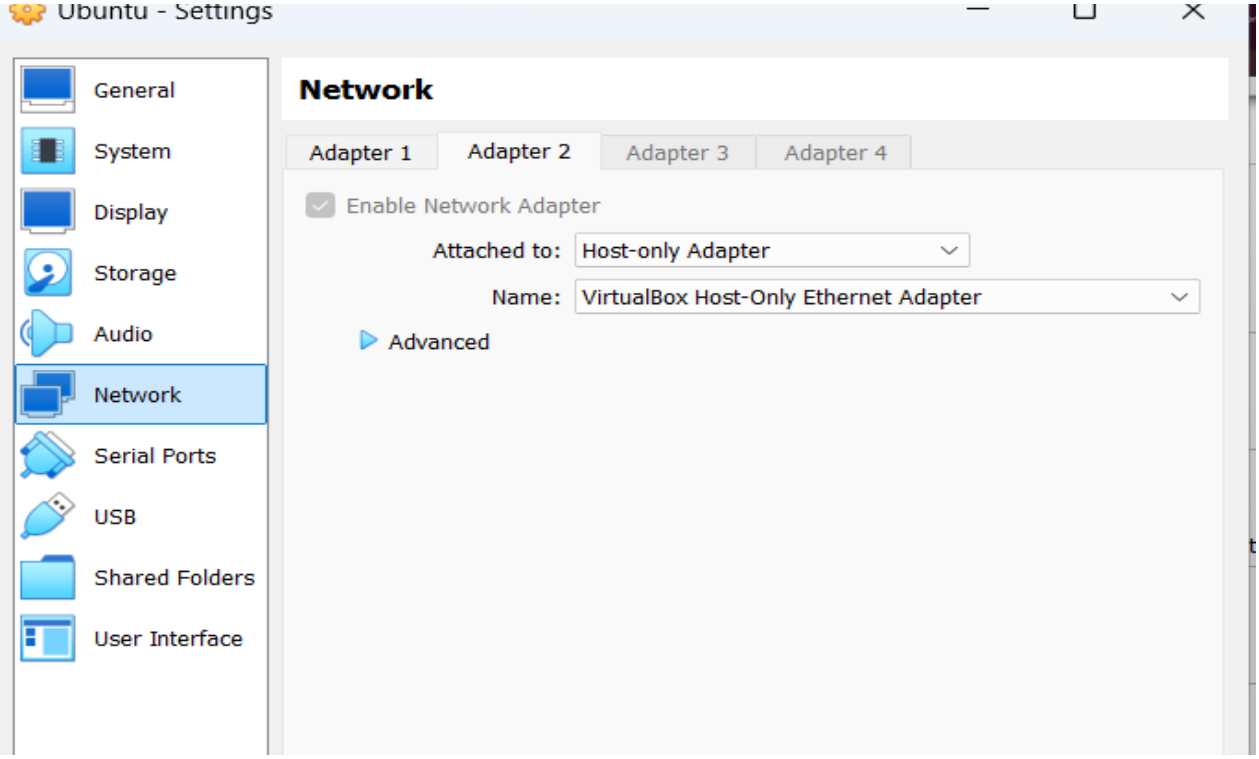


Connectivity issue: NAT did not allow Windows 11 to reach Kibana; access failed from the Windows VM.

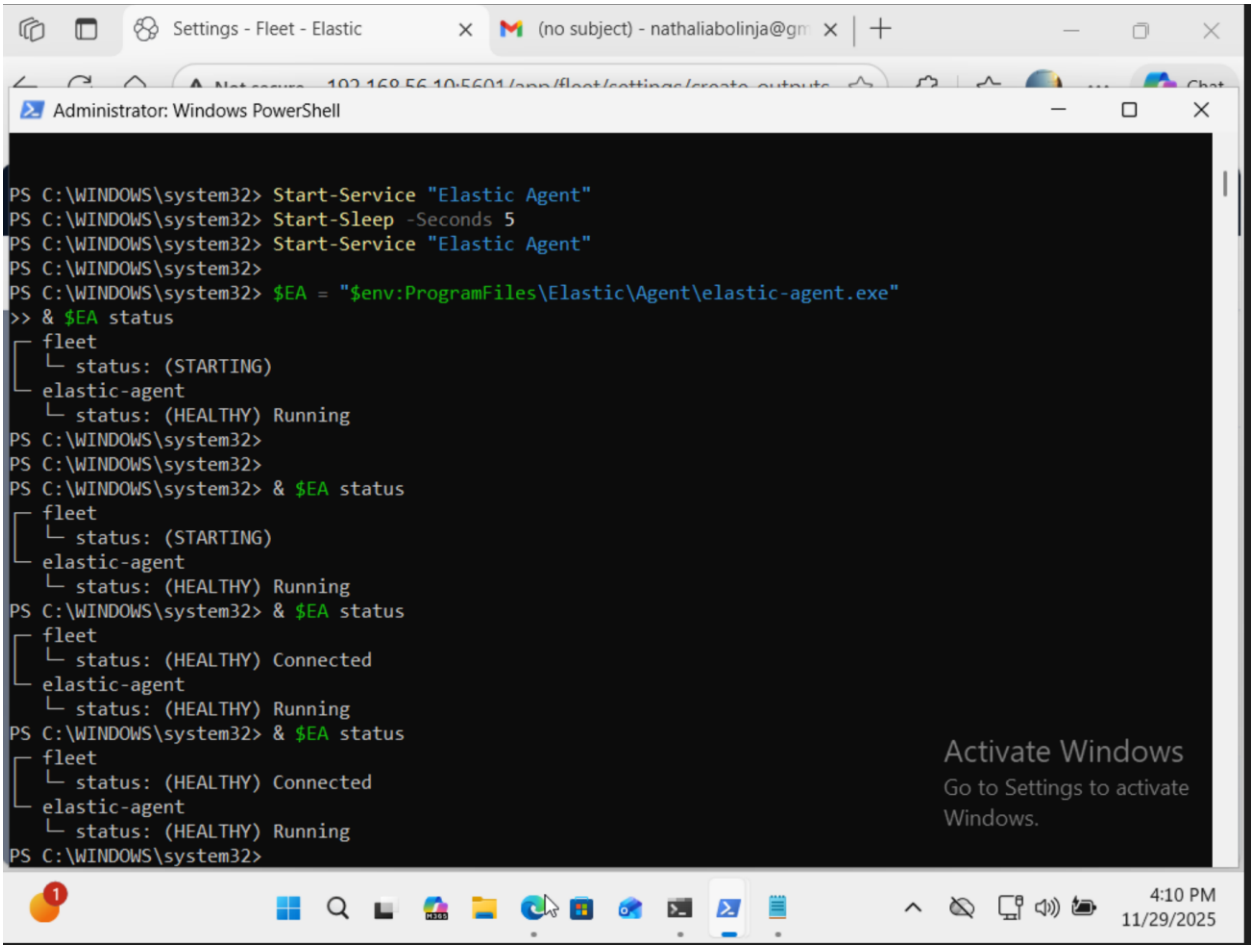


Fix applied: Kibana became accessible from Windows 11 Pro after switching to a Host-only adapter.

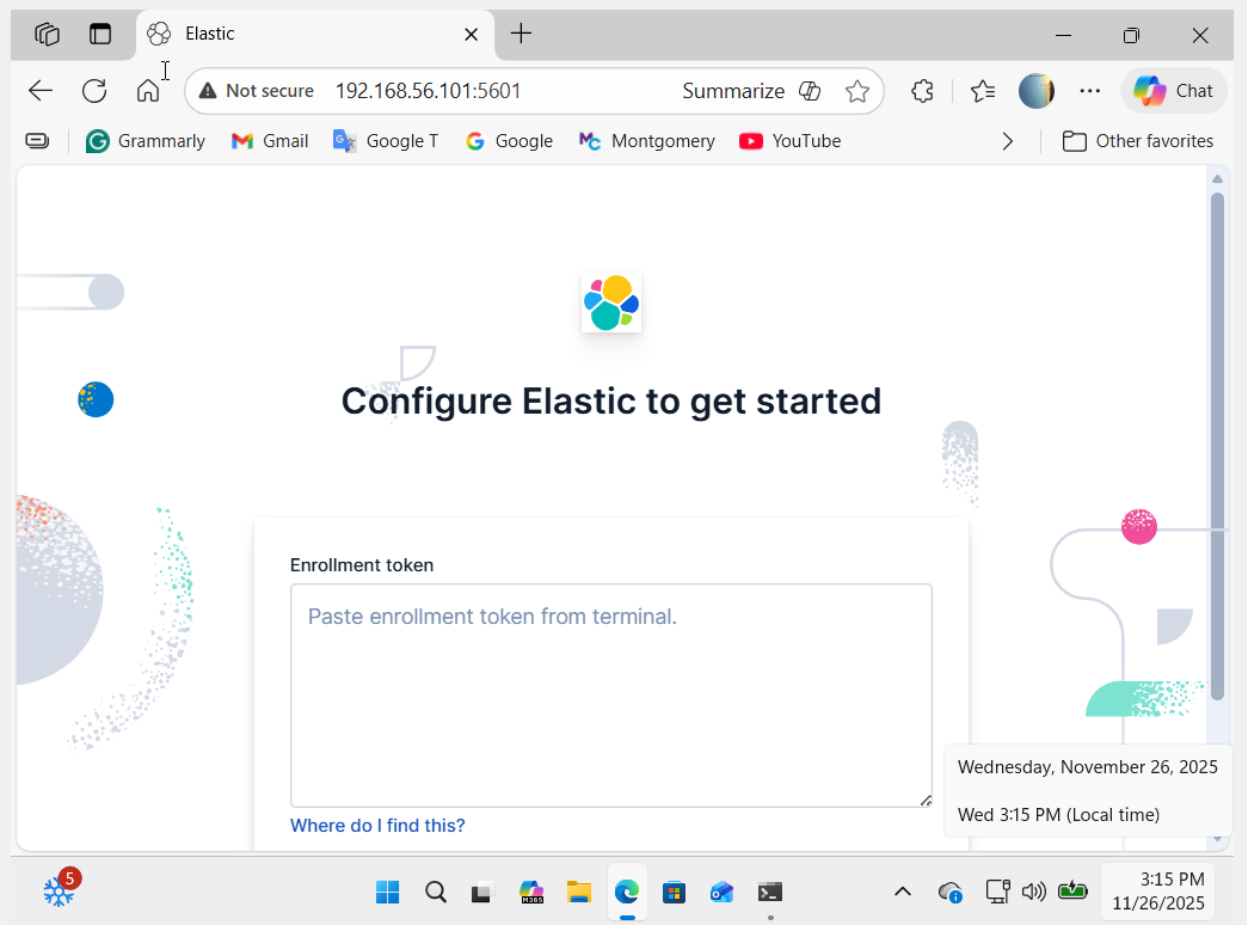
Host-only networking: Adapter 2 set to Host-only for direct Ubuntu - Windows communication; 192.168.56.0/24; Kibana reachable at 192.168.56.101 or 192.168.56.10.



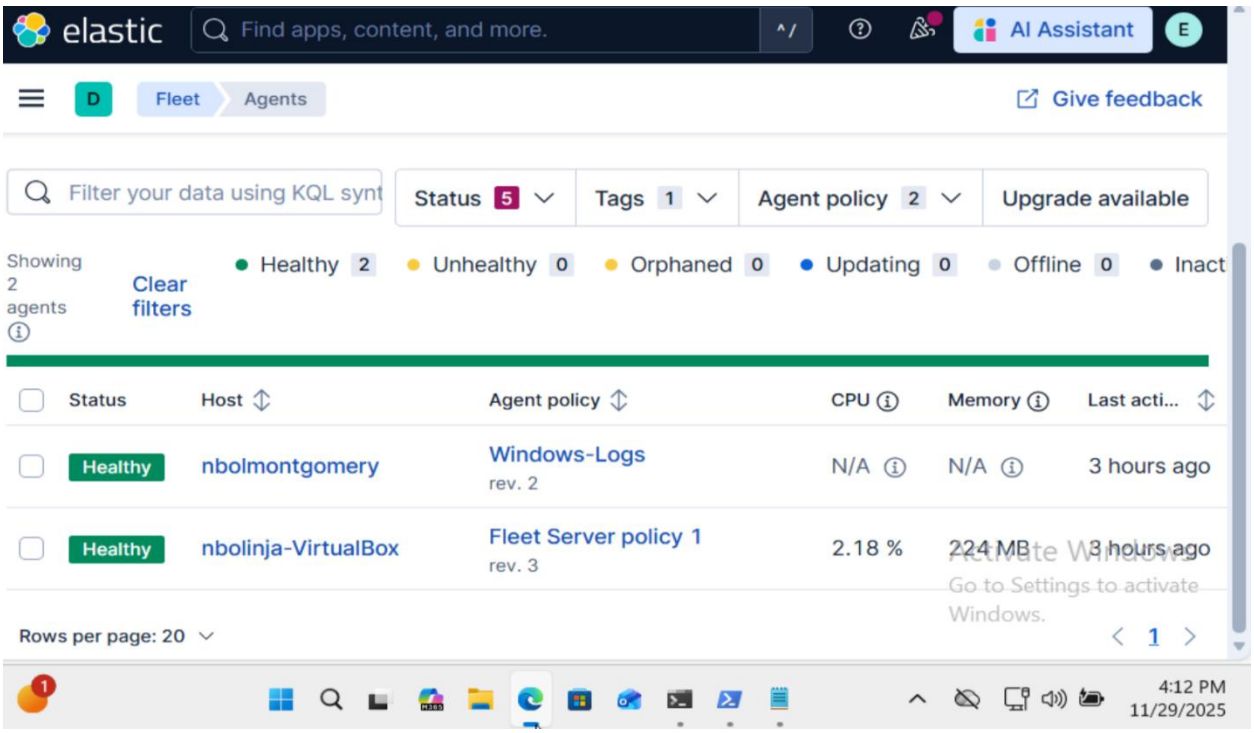
Fleet connectivity: Elastic Agent is running and connected through Fleet.

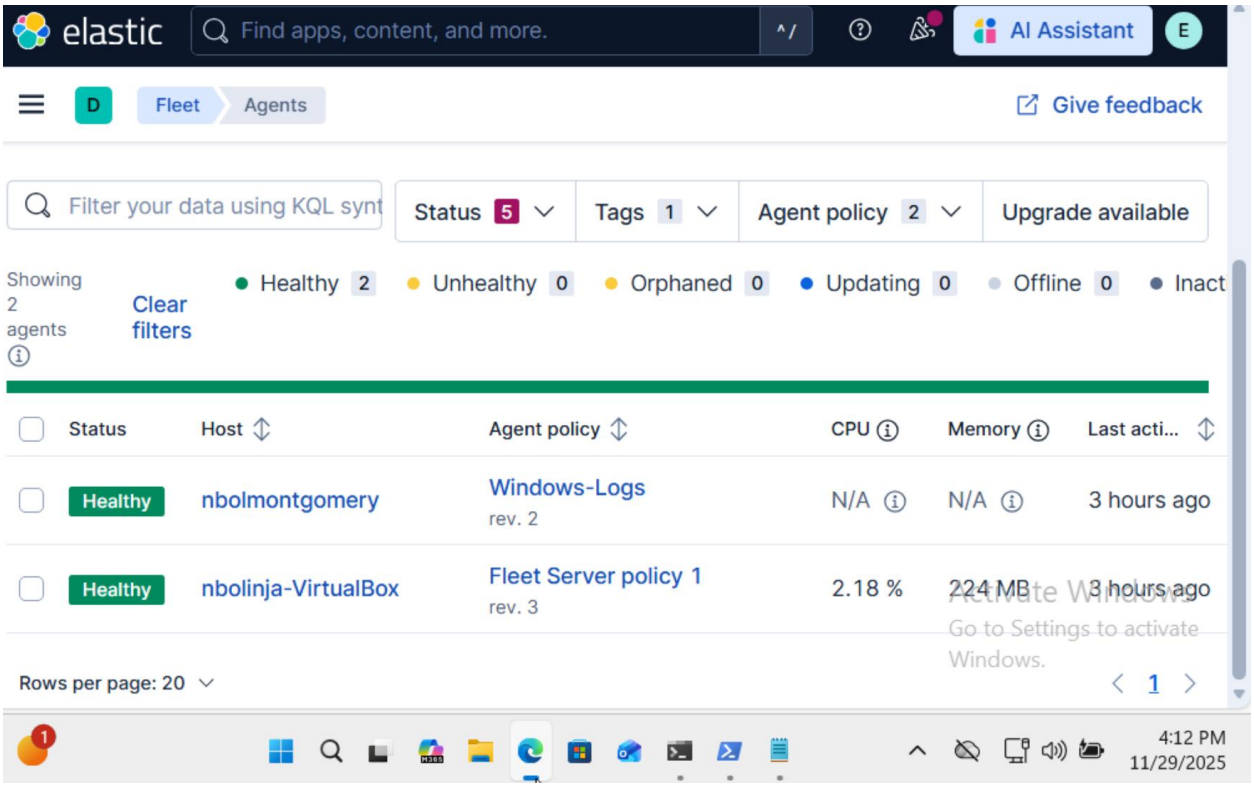


Enrollment token: token used to enroll/register the Elastic Agent in Fleet from the Elastic website workflow.

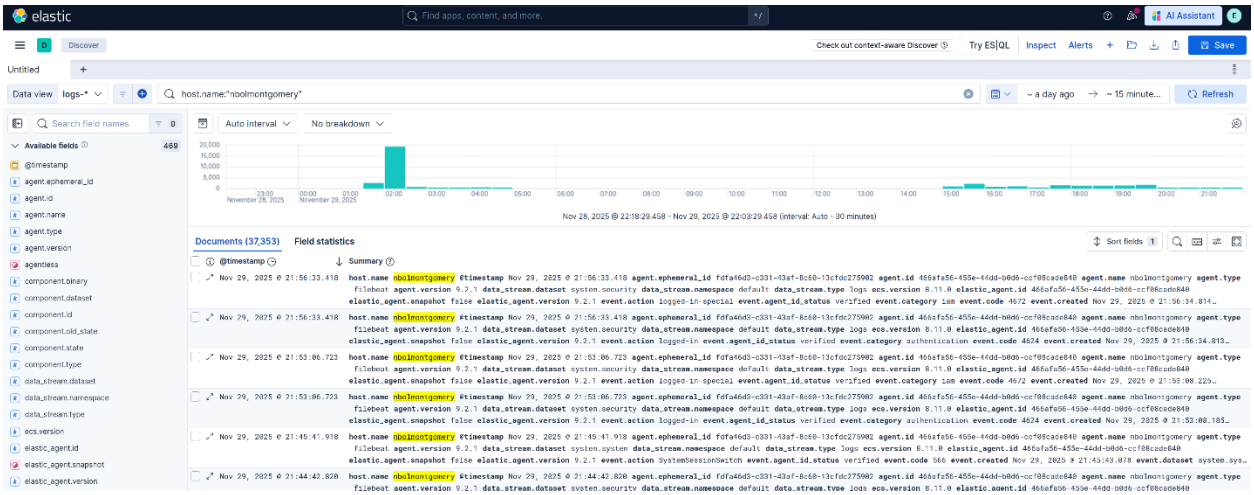


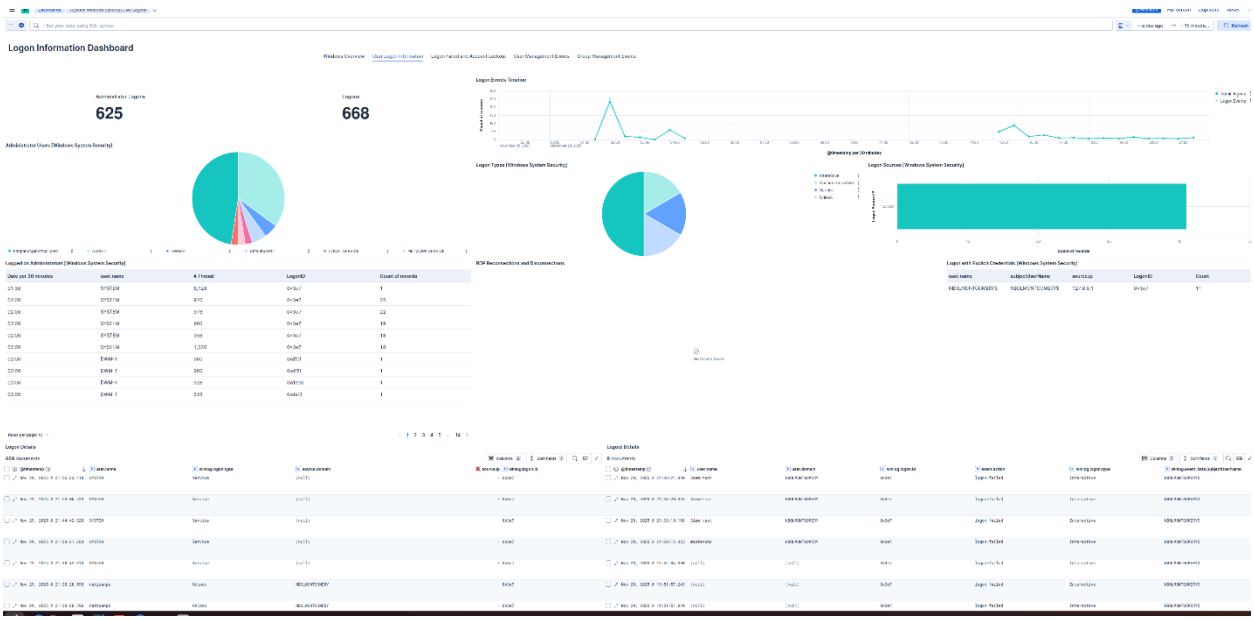
Fleet agents inventory: both endpoints are listed and Healthy; Windows host uses “Windows-Logs” policy; Ubuntu host uses the Fleet Server policy.



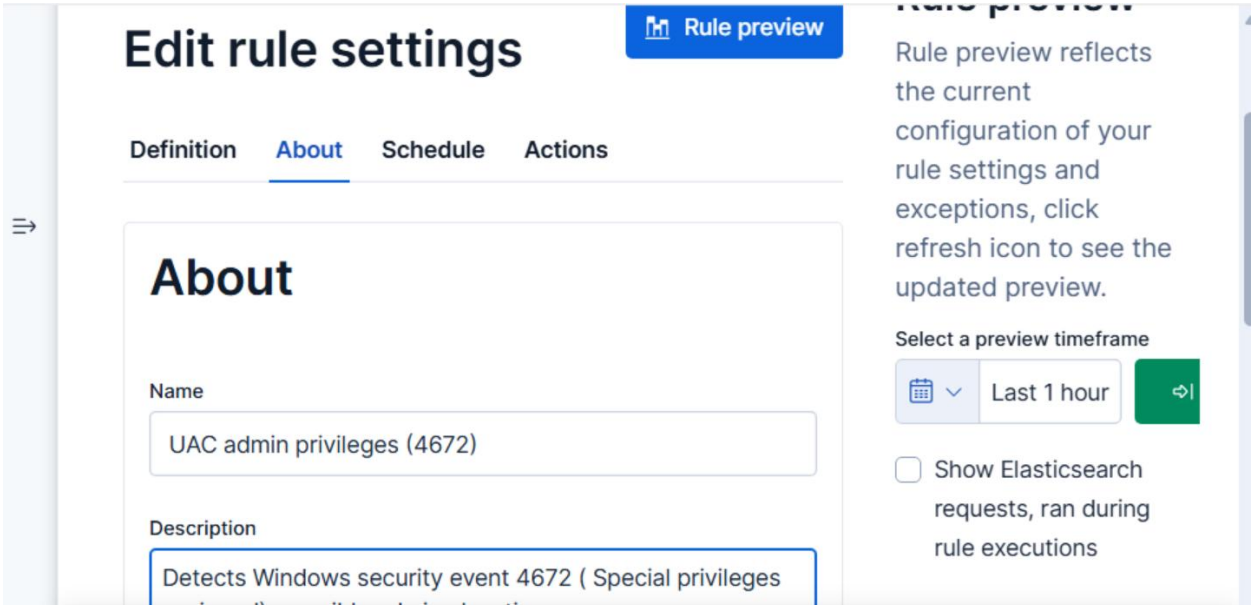


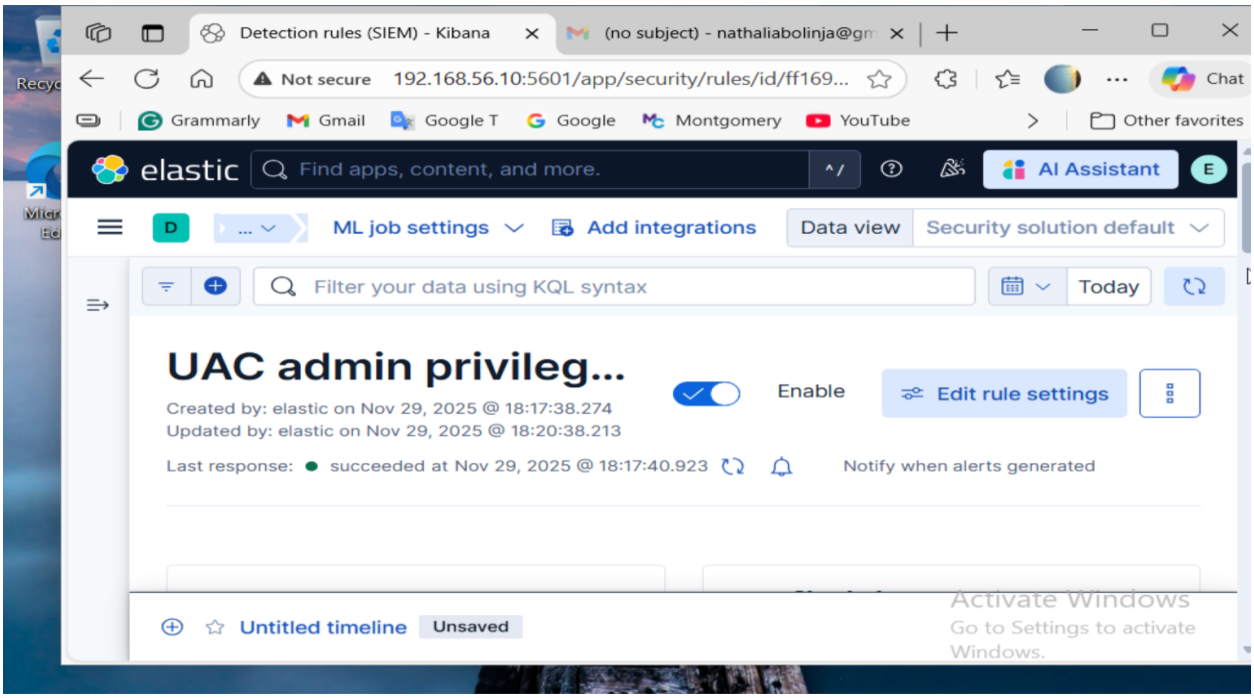
Windows logon dashboard: (System Windows Security) User Logons summarizes Windows logon events received from host nbolmontgomery.





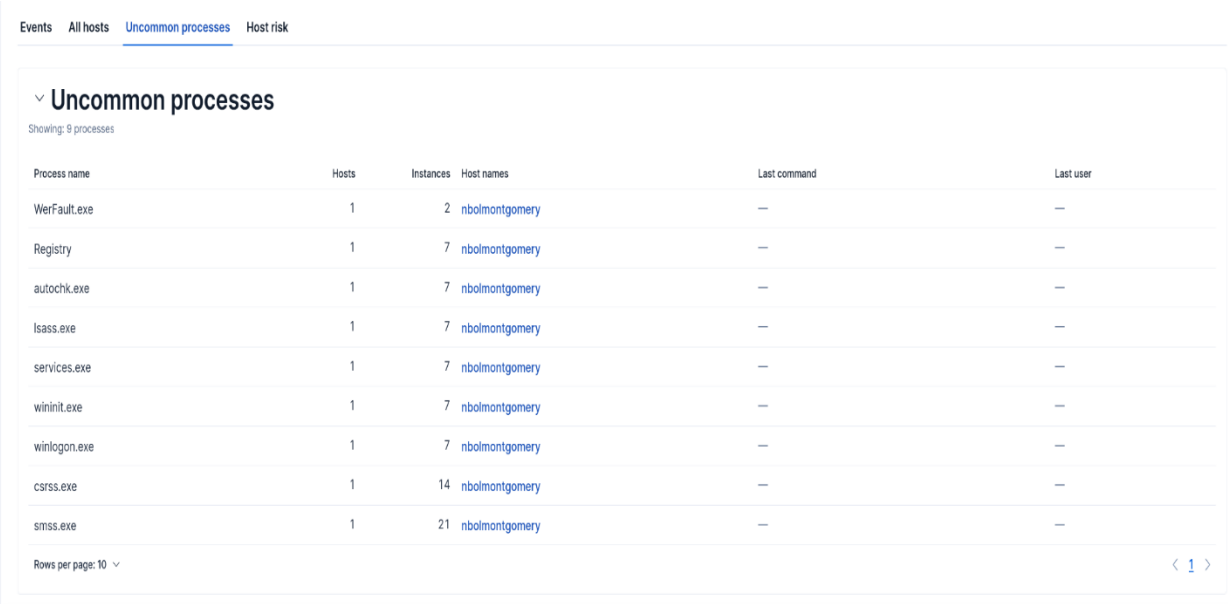
Detection alert: alert created for event.code 4672 to demonstrate privileged-elevation detection in the mini-SIEM.



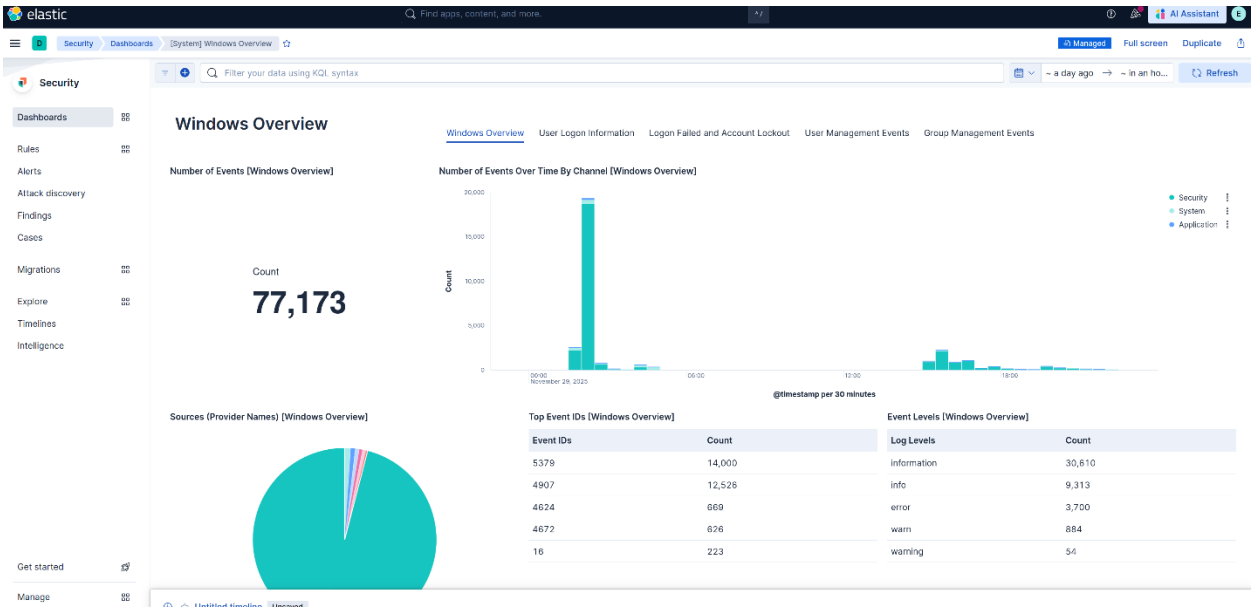


Hosts view: Elastic Security summary of observed hosts, unique IPs, and log-volume over time for the selected period.

“Uncommon processes” is simply a list of Windows processes (programs/services) that ran on the system and, within the time range you selected, appeared less frequently compared to what is typical for that host.



Windows Overview: high-level confirmation that Windows logs are being ingested; shows event volume, top event IDs (4624; 4672), and providers before deeper analysis in Discover.



Conclusion

For this project, I built a mini-SIEM lab using two virtual machines in Oracle VM VirtualBox. The Ubuntu VM runs Elasticsearch and Kibana, and the Windows 11 Pro VM generates Windows Security logs so I could confirm that my system can collect data, display it in dashboards (Kibana), and trigger detections.

Elasticsearch is the backend that receives and stores the Windows logs. It indexes the data so users can fast search, filter, and correlate events during an investigation. While Kibana provides the analysis and visualization tool. It connects to Elasticsearch and displays the collected logs in a user-friendly interface, allowing exploration in *Discover*, dashboard creation, and trend monitoring through charts and metrics (for example, highlighting spikes in failed logon attempts). Using Kibana, I was able to visually confirm log activity, including how many times a user entered the wrong password (failed logon attempts) and test basic detections/alerts based on specific Windows Security Event IDs. However, before getting to this point, I had to install Elasticsearch on Ubuntu and confirmed it was working by checking that the service was active (running). I also installed Kibana and verified it was active (running) and reachable locally at localhost:5601, which confirmed the web interface was available for analysis.

During the integration step, I ran into a networking issue: with NAT enabled, my Windows VM could not reach Kibana. To fix this, I added a Host-only adapter so the Ubuntu and Windows VMs could communicate directly on the 192.168.56.0/24 network. After that change, I was able to access Kibana from Windows using the Ubuntu Host-only IP.

The integration phase was challenging due to hardware limitations. My computer wasn't powerful enough to run these heavy tools reliably, so I ended up buying a more powerful machine to finish the project.

After I fixed the earlier issues, I moved on to the integration phase. From the Windows VM, I signed in to Kibana with my credentials and used Fleet to enroll and manage the Elastic Agents. During this step, Kibana generated a ready-to-run command to install and configure the Fleet Server on the Ubuntu VM. After I ran the command on Ubuntu, I checked Fleet and confirmed that two agents were enrolled and both were reporting Healthy. The Windows host (nbolmontgomery) was assigned the Windows-Logs policy, and the Ubuntu host (nbolinja-VirtualBox) was assigned the Fleet Server policy, confirming end-to-end communication between both endpoints and the Fleet Server.

Finally, I validated the results in Elastic Security. The (System Windows Security) User Logons dashboard showed logon activity from the Windows host in the selected time range. The Hosts and Windows Overview pages also confirmed ingestion volume and important Windows event IDs like 4624 (successful logon) and 4672 (special privileges). To demonstrate detection, I created an alert rule for event.code 4672 to flag privileged-elevation activity as part of the mini-SIEM deliverable.

In conclusion, Elasticsearch and Kibana are powerful tools, and they become even more effective when used together. The project was very challenging because there were many extra configurations needed to get the entire system working end to end. Even so, it was absolutely worth it. I also tried to explore more Elasticsearch and Kibana features, but I was limited because some capabilities require a premium subscription. It's impressive how a small set of tools can give you the ability to build a home security monitoring

environment that can collect logs, visualize activity, and detect suspicious behavior on your systems.