Wazuh SIEM

HOME LAB
SECURITY MONITORING
2026

Prepared by:
Daria Kurnosova

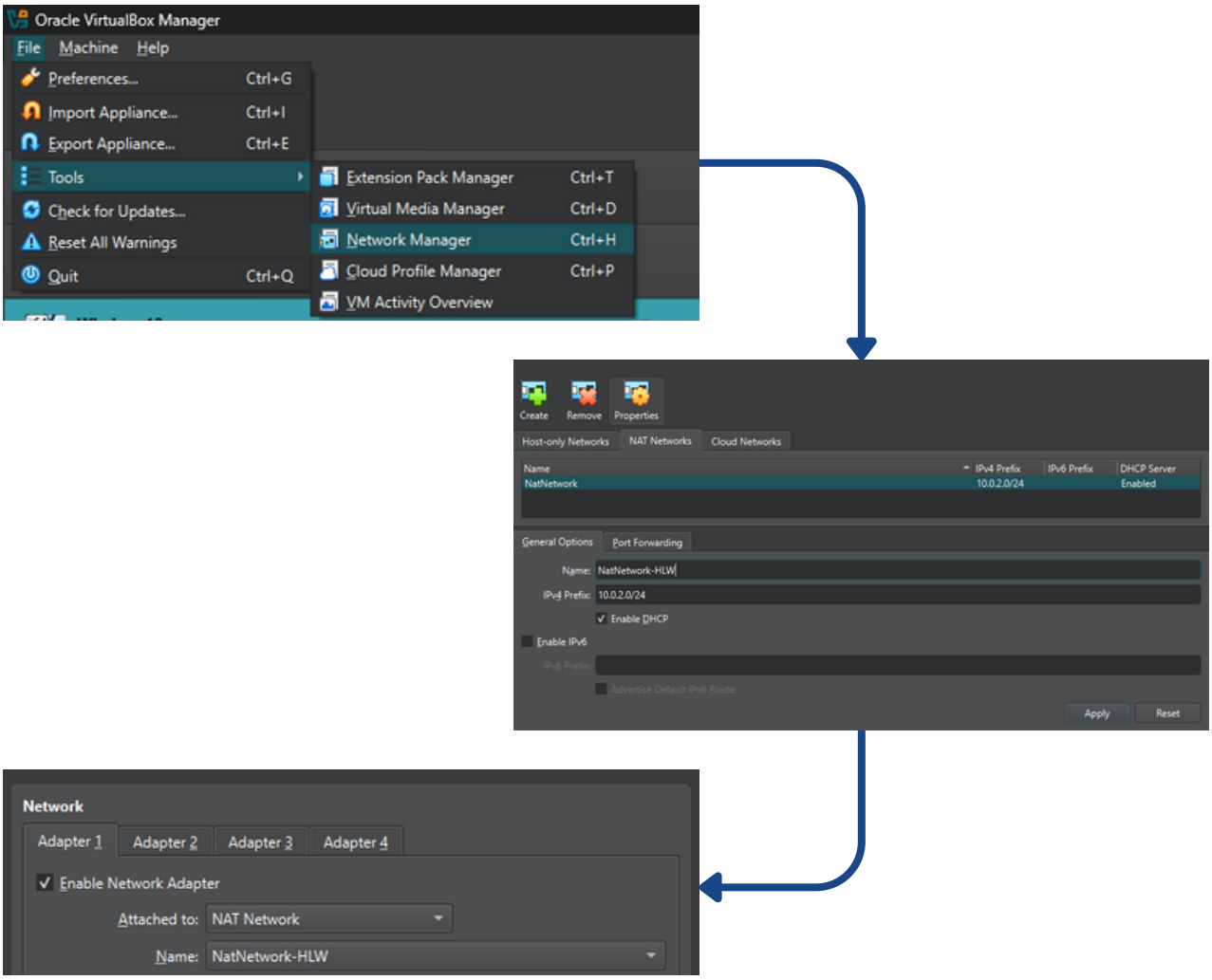
dariakurnosova.contact@gmail.com

Cracow, Poland

The purpose of this report is to guide you through my **Home Lab**. The idea lies in showing my practical knowledge of working with **Wazuh**, **Nmap**, **Hydra**, and different operating systems. I worked specifically with **Kali** Linux, **Ubuntu** Linux, and **Windows 10** here. So, I'm actually going to 'Play Hacker.' Windows 10 is the **victim**, Kali is the **attacker**, and Ubuntu is the **server**.

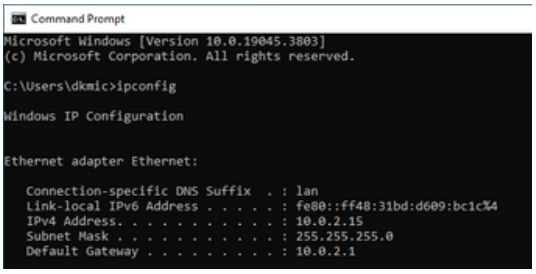
Actions speak louder than words, though.
LET'S GET TO WORK!:)

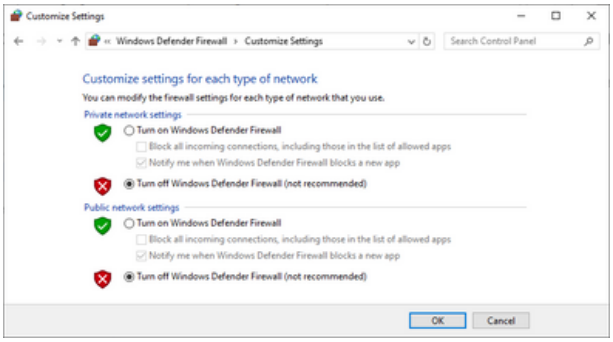
I. First of all, I set up a **NAT Network**, so my Virtual Machines could download updates and talk to each other.



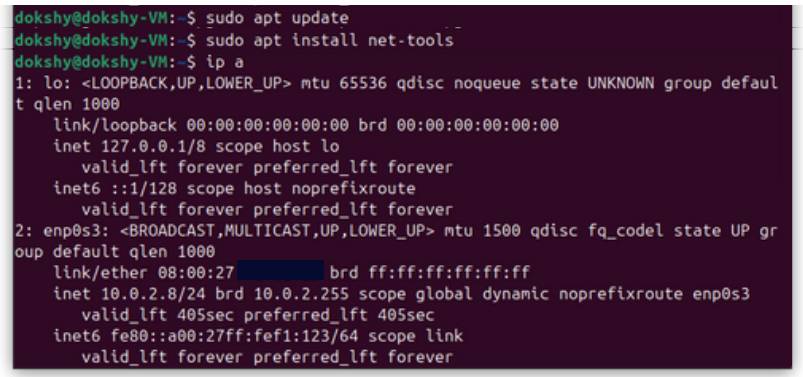
(Repeated for each VM)

II. The next step was to configure the Virtual Machines. So, I started with **Windows 10**, which is our "Victim" for this lab. I checked the IP Address and disabled the firewall.

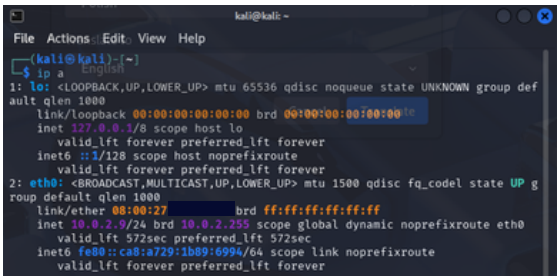


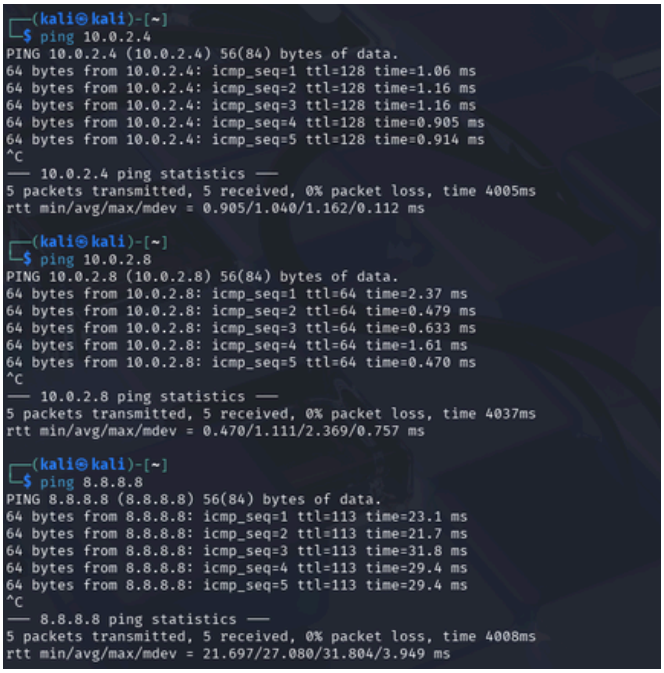


Then I checked Ubuntu **ip**, updated the system and installed essential tools.



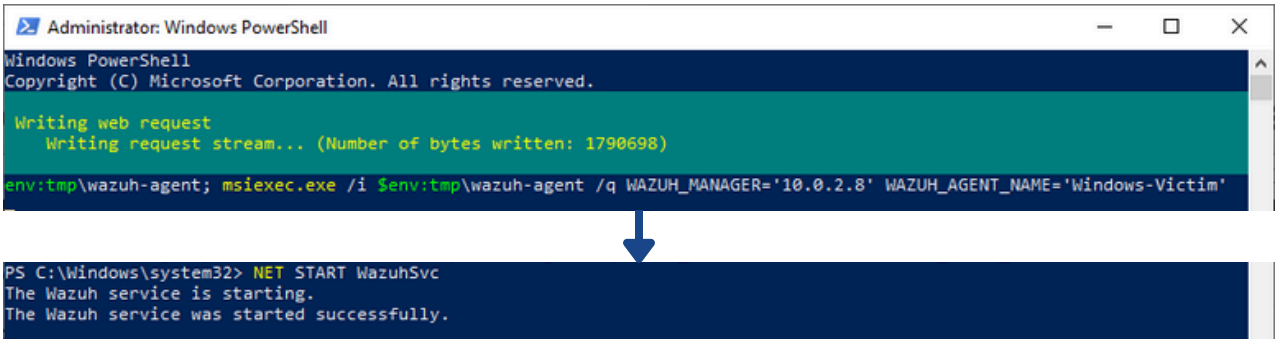
Checked Kali **ip**, updated the system and tried to **ping** Ubuntu and Windows. **SUCCESS!**





IV. Then I deployed new agent (Windows) through Wazuh Dashboard...

...and deployed the Wazuh Agent on the target Windows endpoint via PowerShell and established a connection to the Manager.



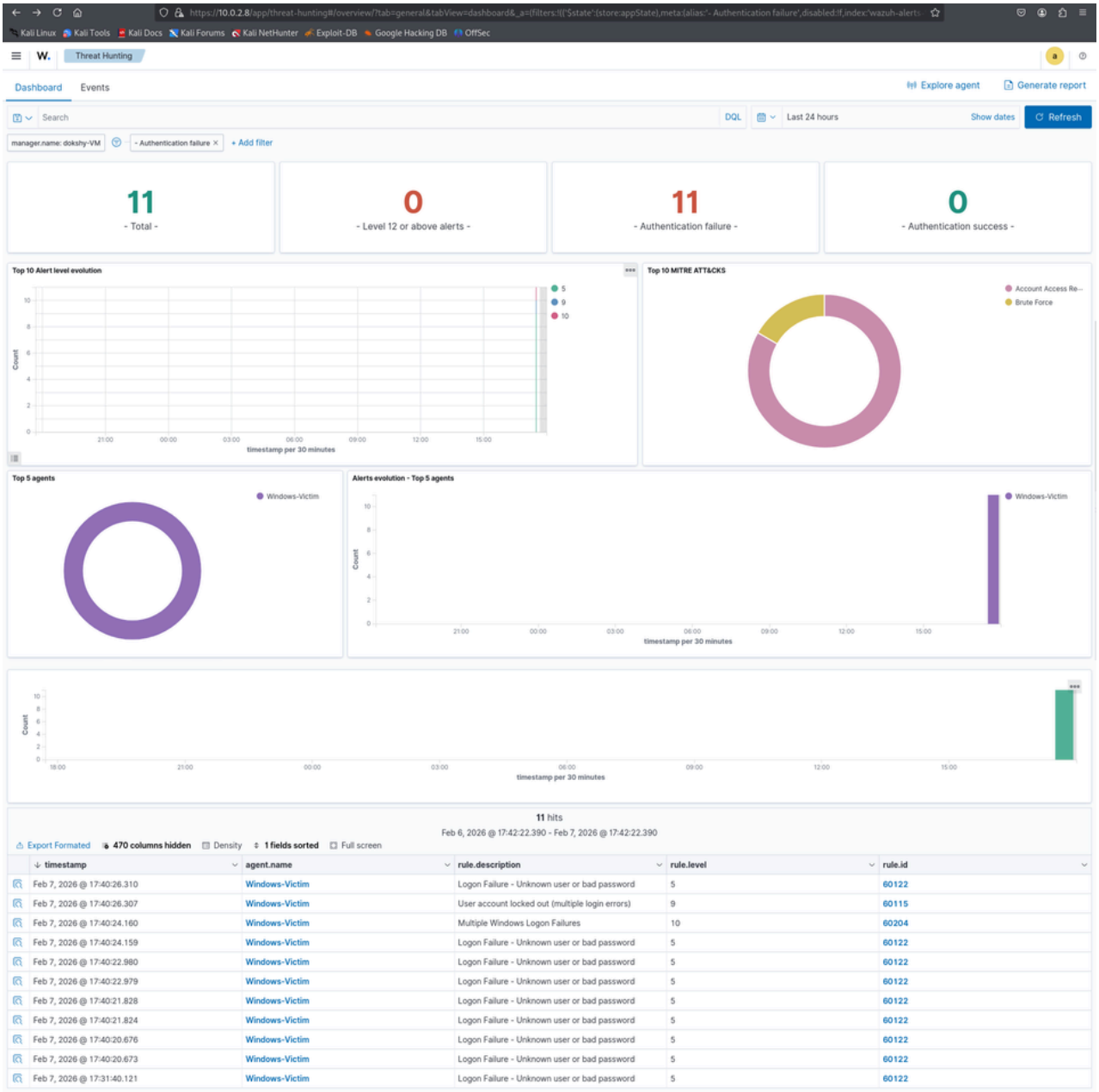
Let’s check... Here! Successful integration of the Windows Endpoint. The SIEM is now receiving telemetry logs in real-time!

And now comes the fun part: **The Red Team Attack**! I am going to use Kali Linux to attack Windows, and verify if Ubuntu (Wazuh) catches it.

V. I verified that Kali could see the Windows open ports using **Nmap** and tried to **brute-force** the Administrator password via SMB.

★ As I know, SMBv1 is an old, insecure version of the protocol, you can use crackmapexec instead.

The Moment of truth... Wazuh actually shows brute-force!



I focused on **External Threat Detection** to stop an attacker at the door. But a strong defense strategy must also include insider threats – threats that have already bypassed the perimeter or came from inside the network.

To address this, the next phase of the lab implements **File Integrity Monitoring (FIM)**. By establishing a cryptographic baseline of critical system files, we shift from detecting 'who is logging in' to detecting 'what is being changed,' ensuring that even if an attacker gains access, their actions—such as installing malware or modifying sensitive data—trigger an immediate alert.

★ I also made FIM project on Python, you can check it out here.

Thank you, and see you next time!