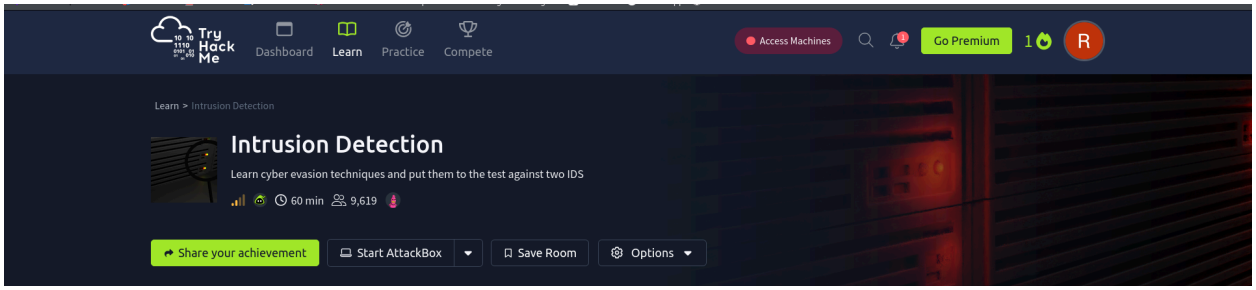
TryHackMe Write-up

Room: Intrusion Detection

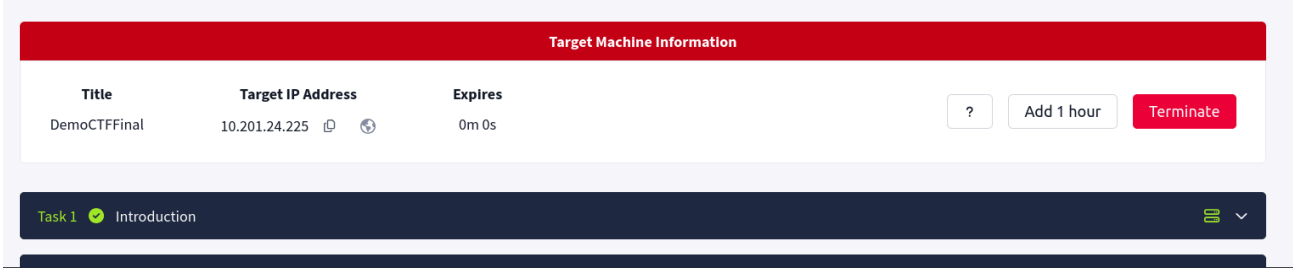
Prepared by: Raseena. R

Date: October 2, 2025



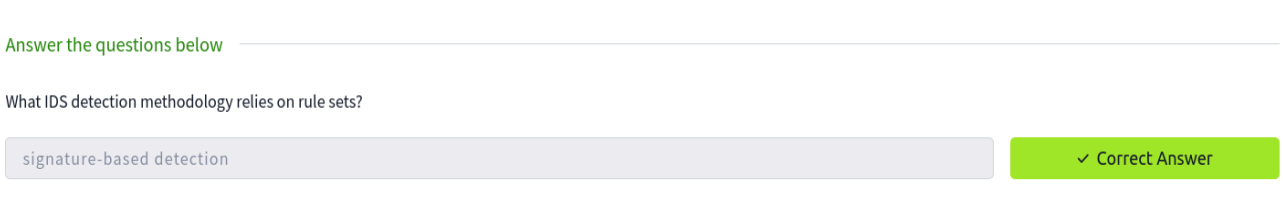
Task 1 - Introduction

Deployed the target machine and registered an account at `http://10.201.24.225:8000`. This room introduces IDS and cyber evasion techniques, allowing a full system takeover while testing evasion methods across the cyber kill chain.



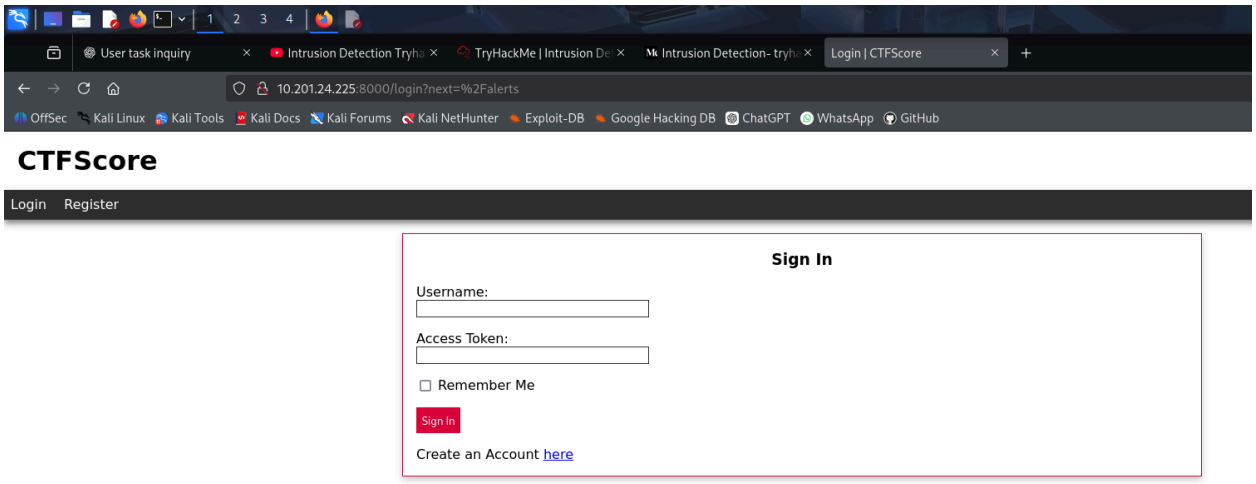
Task 2 - Intrusion Detection Basics

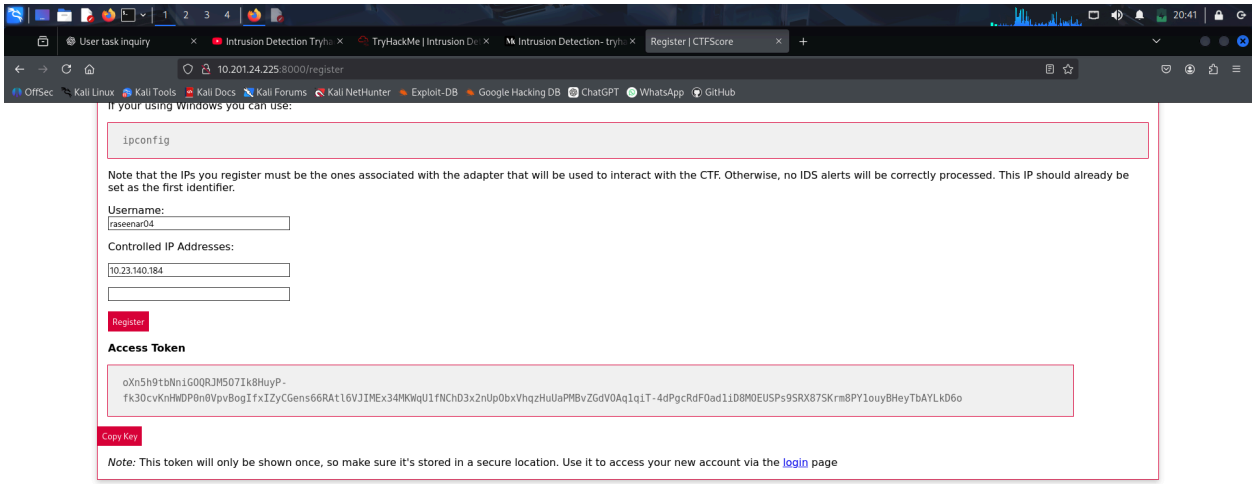
Intrusion Detection Systems (IDS) monitor network or host activity to detect suspicious behavior. Signature-based IDS use predefined rules to find threats, while anomaly-based IDS detect deviations from normal activity. Alerts generated by IDS can be sent to logging or visualization platforms like Graylog or ELK. In this demo, **Suricata** (network-based) and **Wazuh** (host-based) are used to illustrate signature-based detection.

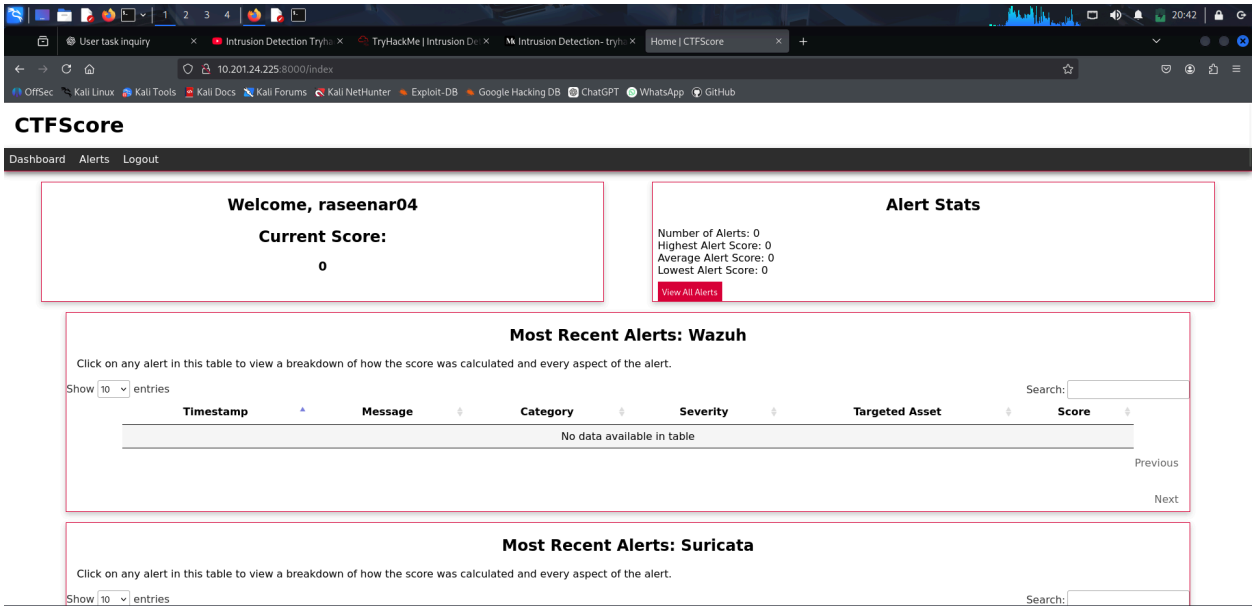


Task 3 - Network-based IDS (NIDS)

Network-based IDS (NIDS) monitor network traffic for malicious activity like malware, scanning, data exfiltration, and phishing. They can cover an entire network with a single deployment but may produce false positives due to high traffic volume. NIDS, such as **Suricata** in this demo, can also include IPS features to block attackers, though IPS is disabled here to allow testing. Alerts generated can be viewed at `http://10.201.24.225:8000/alerts`.







Answer the questions below

What widely implemented protocol has an adverse effect on the reliability of NIDS?

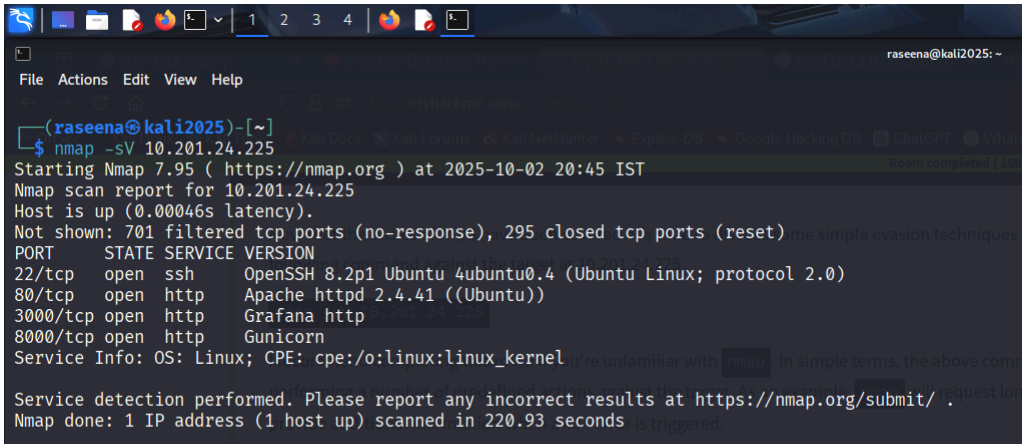
TLS ✓ Correct Answer Hint

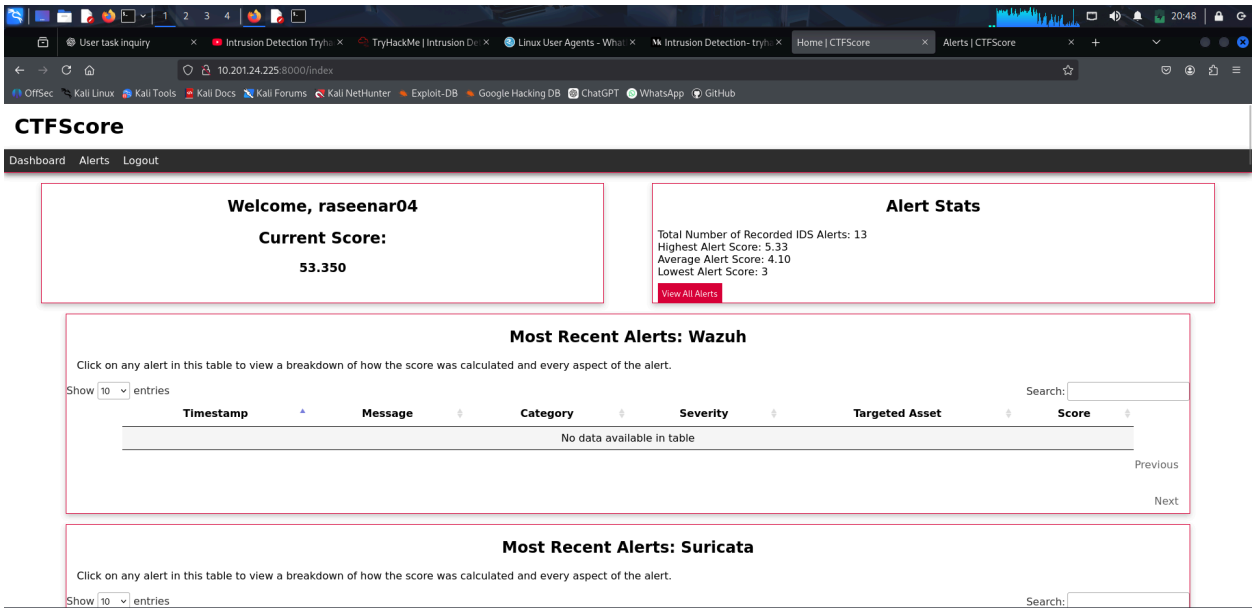
Experiment by running tools against the target and viewing the resultant alerts. Is there any unexpected activity?

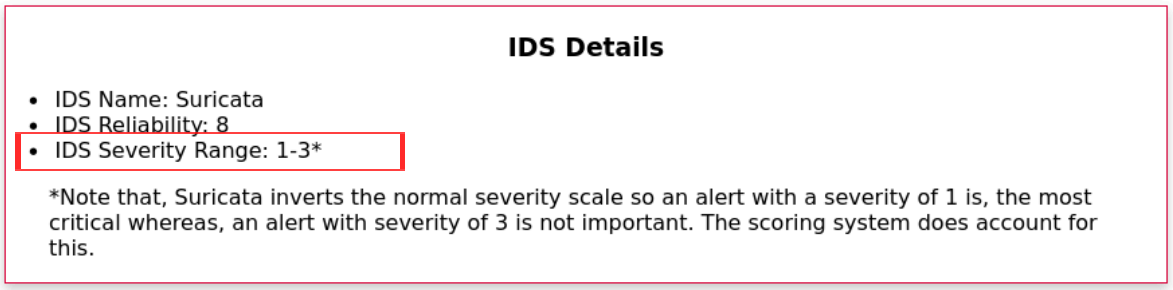
No answer needed ✓ Correct Answer Hint

Task 4 - Reconnaissance and Evasion Basics

Performed initial reconnaissance on the target using `nmap -sV 10.201.24.225` to enumerate services. This basic scan triggers IDS alerts from Suricata and Wazuh due to default nmap behaviors. To evade detection, a custom User-Agent was set with `--script-args http.useragent="Mozilla/5.0 (X11; U; Linux x86_64; de; rv:1.8.1.1) Gecko/20061223 BonEcho/2.0.0."`, reducing alert visibility. Stealth SYN scans (`-sS`) were also tested, which return less information but generate fewer alerts. This illustrates the trade-off between evasion and information gathering during reconnaissance.







Answer the questions below

What scale is used to measure alert severity in Suricata? (*-*)

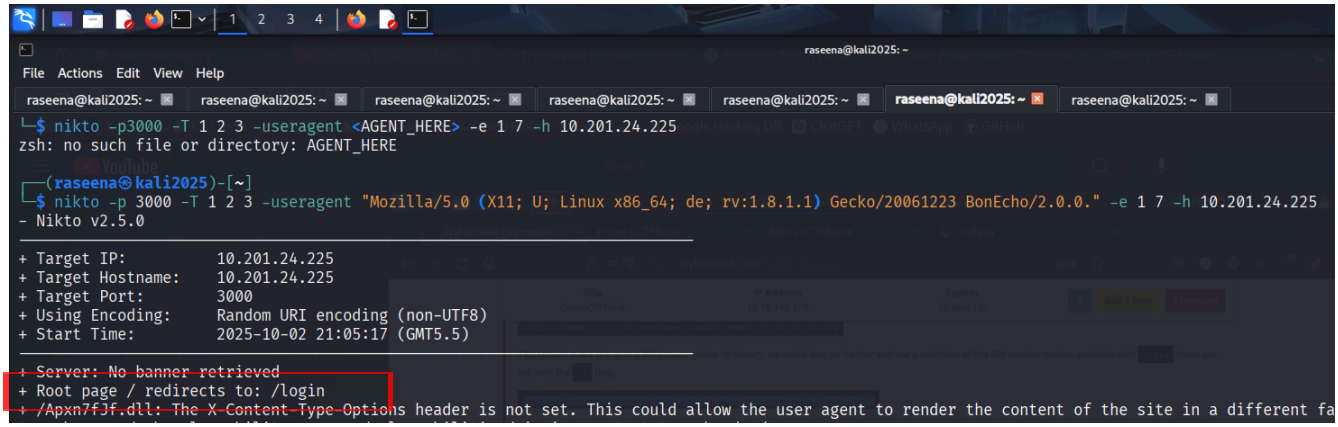
1-3 ✓ Correct Answer Hint

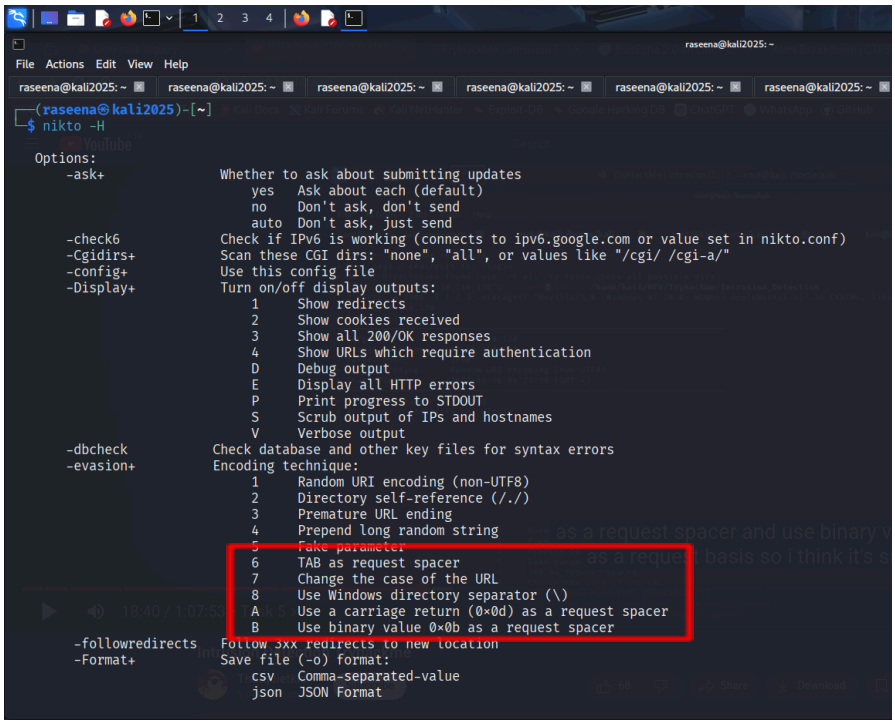
How many services is nmap able to fully recognise when the service scan (-sV) is performed?

3 ✓ Correct Answer Hint

Task 5 - Further Reconnaissance Evasion

Performed web reconnaissance using **Nikto** to scan the target's services on ports 80 and 3000. Initial scans generated a large number of IDS alerts (~7000). By narrowing the scan to port 3000 and specific vulnerability categories, the scan became faster and less noisy. Custom User-Agents and Nikto evasion options (-e) were applied, though some evasion techniques increased IDS detectability due to unusual packet data. This highlights the balance between aggressive scanning and avoiding IDS alerts.





Answer the questions below

Nikto, should find an interesting path when the first scan is performed, what is it called?

/login ✓ Correct Answer

What value is used to toggle denial of service vectors when using scan tuning (-T) in nikto?

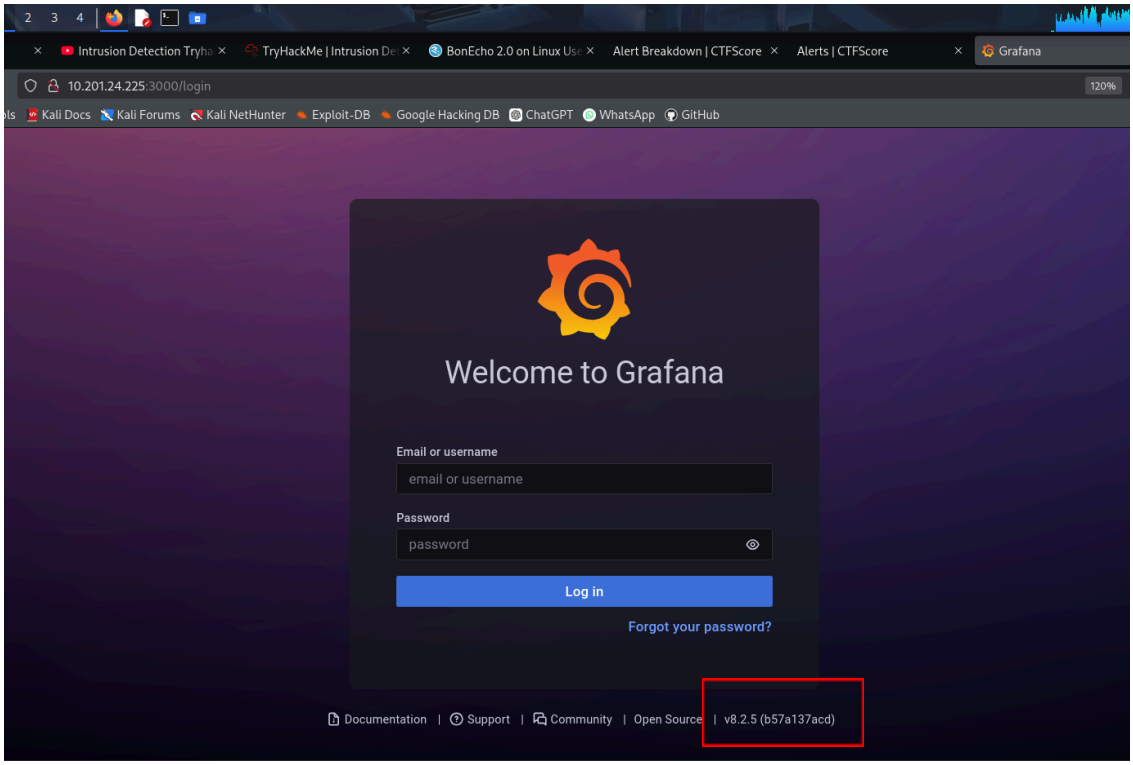
6 ✓ Correct Answer Hint

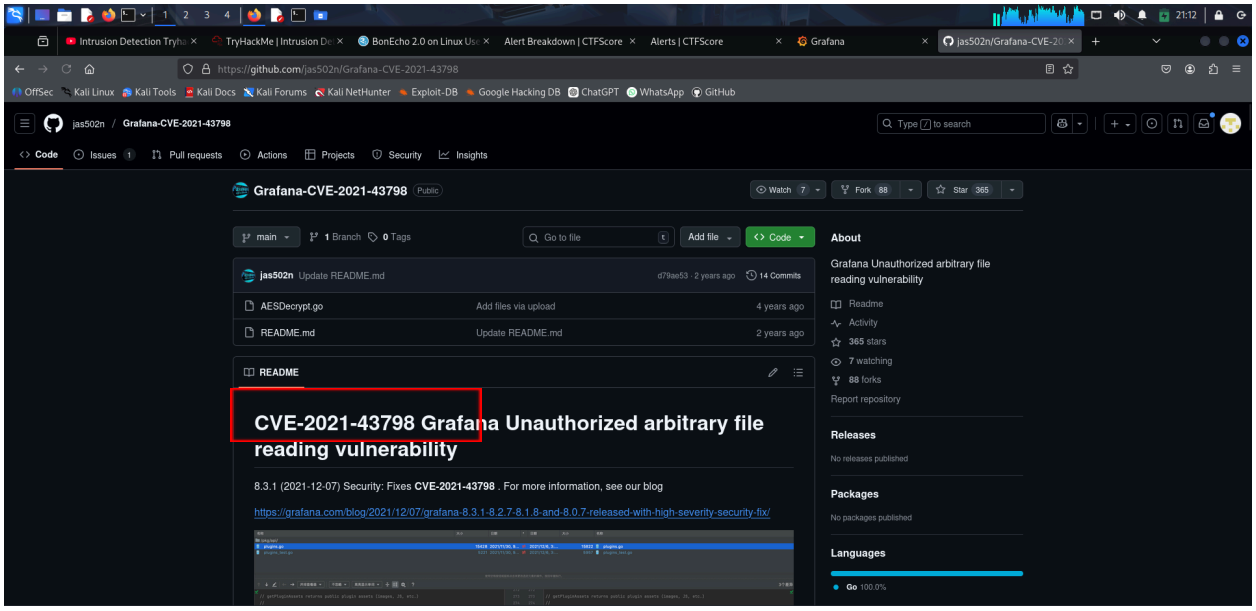
Which flags are used to modify the request spacing in nikto? Use commas to separate the flags in your answer.

6,A,B ✓ Correct Answer Hint

Task 6 - Open-source Intelligence

OSINT involves gathering information passively from publicly available sources, making it largely undetectable by IDS. Tools like **Shodan**, search engines, and WHOIS queries can reveal service details, subdomains, IPs, and hosting information. Public-facing sites may also expose technologies, server details, or tools used by the target. For this task, reconnaissance was performed using these passive methods to collect information without triggering alerts.





Answer the questions below

What version of Grafana is the server running?

8.2.5 ✓ Correct Answer Hint

What is the ID of the severe CVE that affects this version of Grafana?

CVE-2021-43798 ✓ Correct Answer Hint

If this server was publicly available, What site might have information on its services already?

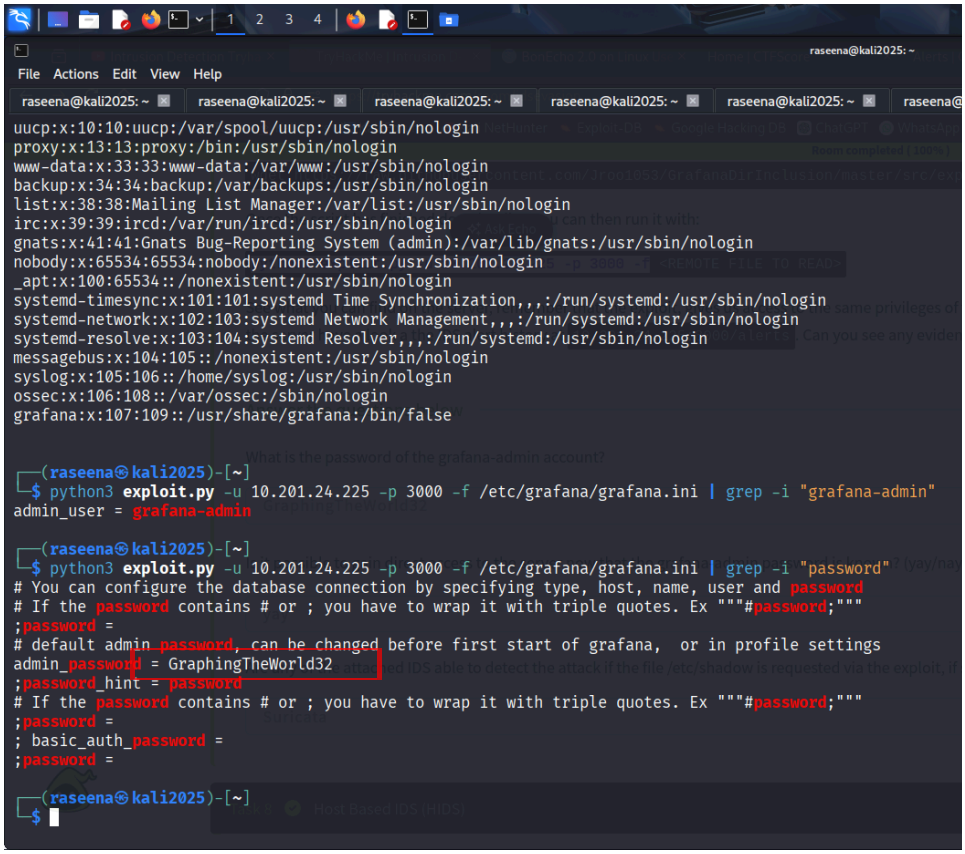
shodan ✓ Correct Answer

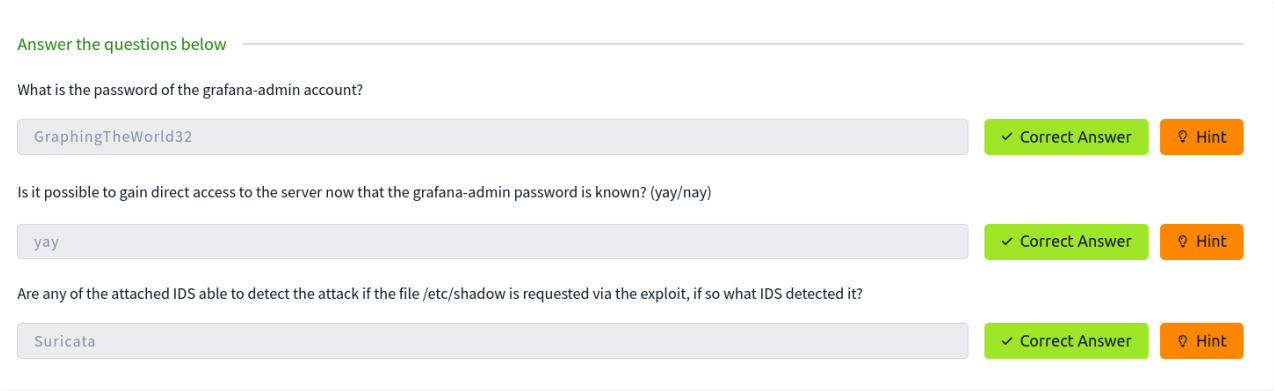
How would we search the site "example.com" for pdf files, using advanced Google search tags?

site:example.com filetype:pdf ✓ Correct Answer

Task 7 - Rulesets

Signature-based IDS rely on up-to-date and accurate rulesets to detect attacks, but imperfect rules can lead to false positives or missed threats. A critical vulnerability in the target allowed bypassing authentication and reading files on the server. The exploit script from GitHub (`exploit.py`) was run against port 3000, and the IDS alert history was checked to observe whether Suricata detected the attack. This demonstrates that even known exploits may not always trigger alerts due to ruleset limitations.

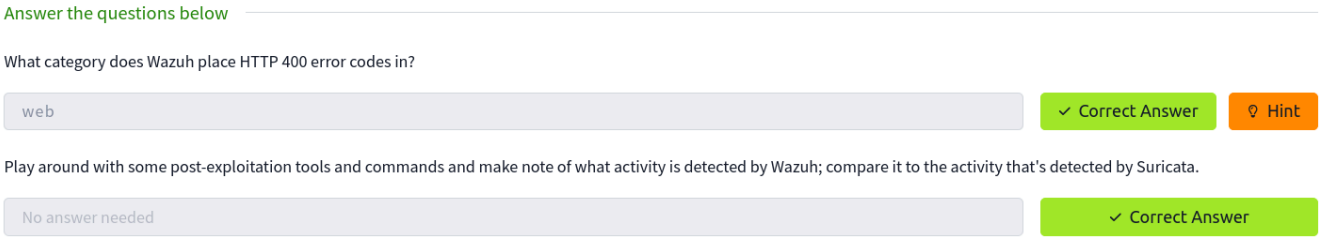




Task 8 - Host Based IDS (HIDS)

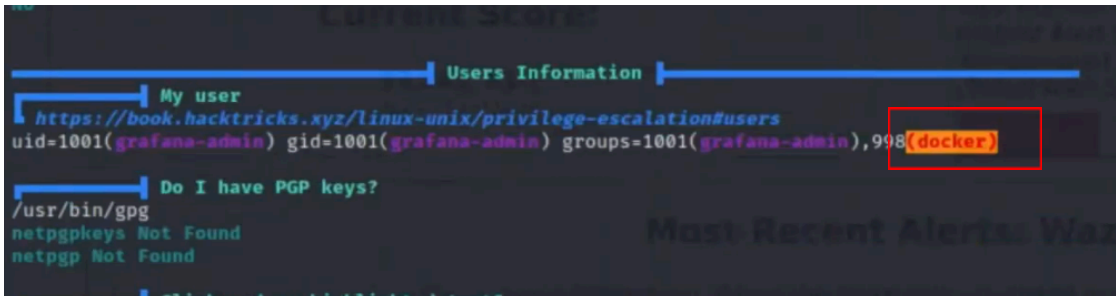
Host-based IDS (HIDS) monitor activity on individual systems, detecting malware execution, configuration changes, file integrity issues, and privilege escalation. HIDS agents collect data from logs, the registry, system metrics, and the file system, then forward it for analysis. In this demo, **Wazuh** detected insecure SSH attempts and HTTP error log entries generated by `nmap --script=vuln`, while Suricata focused on network

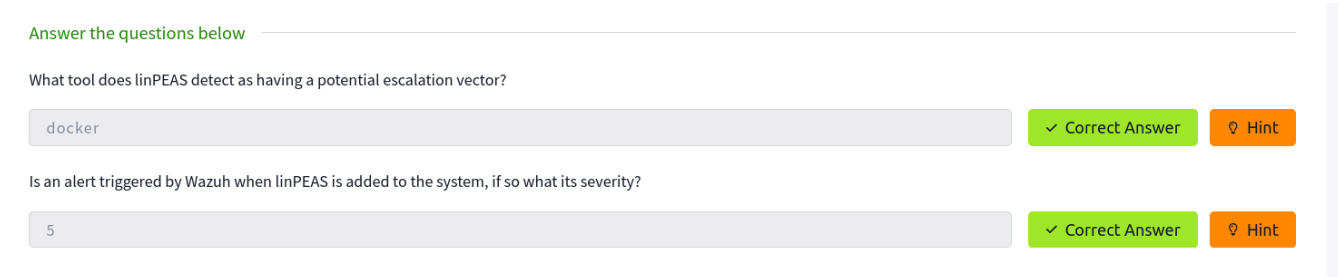
traffic. This highlights the complementary roles of HIDS and NIDS in security monitoring.



Task 9 - Privilege Escalation Recon

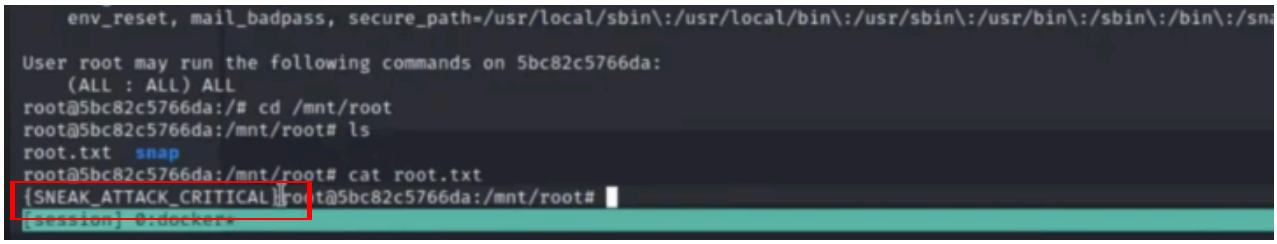
Checked user permissions using `sudo -l`, `groups`, and `cat /etc/group` to identify potential privilege escalation opportunities. These commands do not trigger network-based alerts, so Suricata remained silent, while Wazuh may log local activity. Running **linPEAS** provided a detailed system analysis and identified possible privilege escalation vectors, though it generated more IDS alerts due to the extensive reconnaissance it performs.

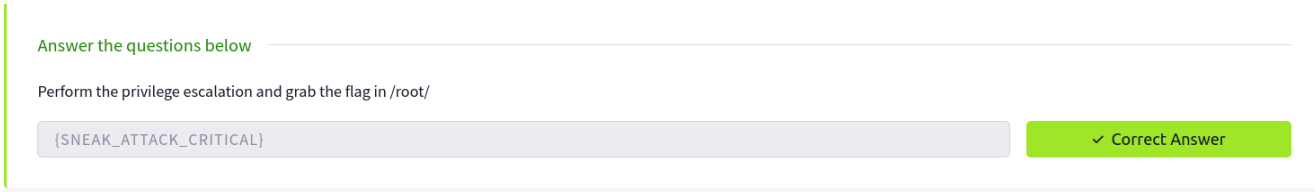




Task 10 - Performing Privilege Escalation

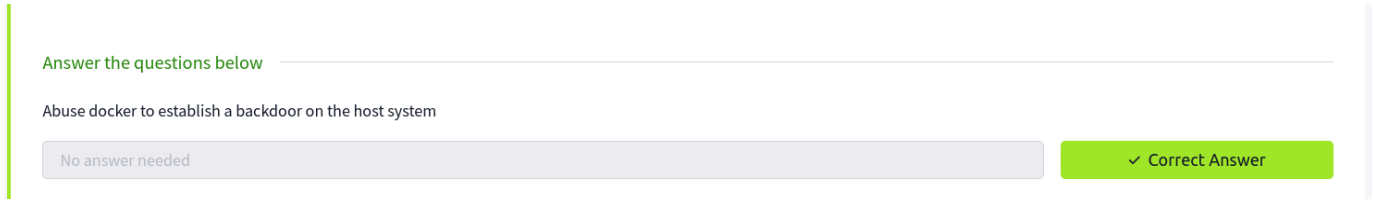
Used Docker as a privilege escalation vector by running a container with the host file system mounted. This allowed root-level access without sudo. Elevated privileges were granted by editing `/etc/sudoers` to add `grafana-admin`, while monitoring IDS alerts from Wazuh for any detected modifications. This demonstrates how container misconfigurations can lead to full system compromise.





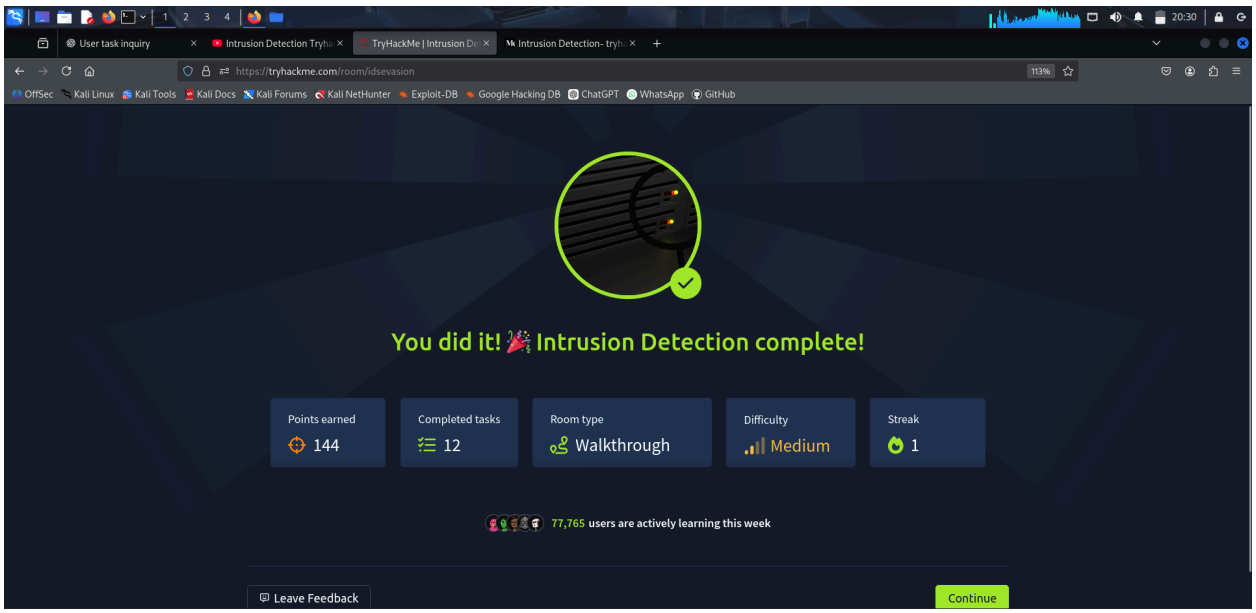
Task 11 - Establishing Persistence

Persistence was established on the compromised host by creating a Docker-based backdoor. A new container was configured via `docker-compose` to mount the host file system and spawn a reverse shell to the attack box. This method avoids modifying monitored files like `/root/.ssh/authorized_keys` and allows continuous access without relying on SSH credentials or vulnerable services, while bypassing HIDS file integrity monitoring.



Task 12 - Conclusion

This room provided hands-on experience with IDS evasion, reconnaissance, privilege escalation, and persistence techniques. It also demonstrated the new CTF scoring system, which tracks IDS alerts and evaluates attack actions.



"I have successfully completed the **TryHackMe: ID Evasion** room, covering all tasks and learning IDS evasion, reconnaissance, privilege escalation, and persistence techniques.✅"