

TryHackMe – “ID Evasion” Room

Completion Report

Participant: AKHIL NS

October 18, 2025

Summary

This document summarizes the completion of the TryHackMe: *ID Evasion* room. The room demonstrates IDS concepts, evasion techniques across the cyber kill chain, and culminates in full system takeover with demonstrated persistence. The following sections map directly to the room tasks and include short descriptions of actions and observations.

Task-by-task Summary

- Task 1: Introduction.** Deployed the target VM and registered an account on `http://10.201.102.178:8000`. The lab framework and objectives were verified.
- Task 2: Intrusion Detection Basics.** Reviewed IDS concepts: signature-based vs anomaly-based detection. Noted that Suricata (NIDS) and Wazuh (HIDS) were used in the lab; alerts are forwarded to logging/visualization components.
- Task 3: Network-based IDS (NIDS).** Observed Suricata capturing network traffic and generating alerts. Confirmed alert viewing via the lab alerts interface.
- Task 4: Reconnaissance and Evasion Basics.** Ran `nmap -sV 10.201.102.178`. Default scans triggered IDS alerts. Reduced detection by using a custom HTTP User-Agent (e.g. `-script-args http.useragent="Mozilla/5.0 (...)"`) and tested stealth SYN scans (`-sS`).
- Task 5: Further Reconnaissance Evasion.** Performed web scans with Nikto against ports 80 and 3000. Narrowing to port 3000 and specific test categories greatly reduced noise. Applied Nikto evasion flags and custom User-Agents; some evasion techniques increased suspiciousness due to uncommon packet contents.
- Task 6: Open-source Intelligence (OSINT).** Collected passive intelligence (Shodan/search engines/WHOIS) to minimize detection. Documented discovered service details and public artifacts.
- Task 7: Rulesets.** Executed an exploit script (from GitHub) against the vulnerable service on port 3000 to test detection. Observed the IDS alert history to evaluate Suricata rules coverage and noted where known exploits were not flagged.
- Task 8: Host-Based IDS (HIDS).** Inspected Wazuh alerts for local activity (e.g. insecure SSH attempts, HTTP error logs from active scans). Compared HIDS visibility vs

NIDS visibility for different actions.

Task 9: Privilege Escalation Recon. Performed local enumeration (`sudo -l`, `groups`, `cat /etc/group`) and ran linPEAS. Confirmed that some local recon produces more IDS/HIDS telemetry.

Task 10: Performing Privilege Escalation. Abused Docker misconfiguration by running a container with the host filesystem mounted, enabling root access. Modified sudoers to add `grafana-admin` as an elevated user and observed Wazuh alerts for the change.

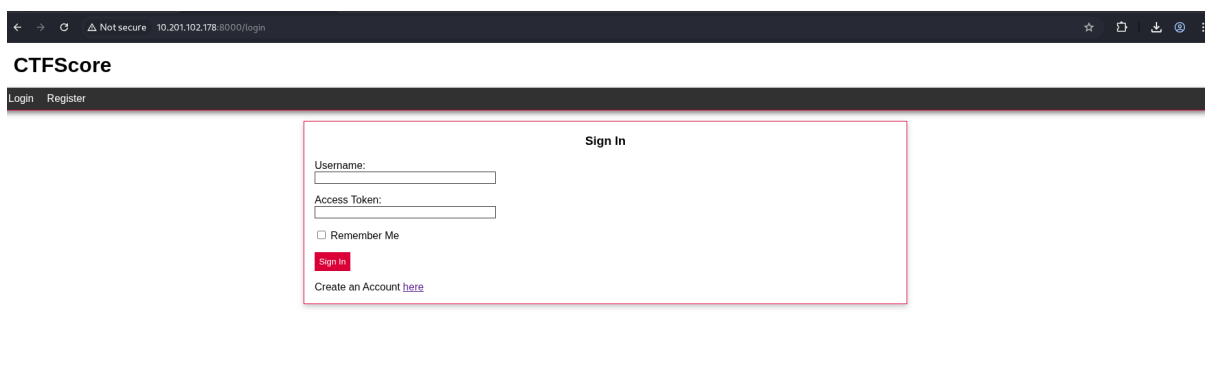
Task 11: Establishing Persistence. Created a Docker-based backdoor via `docker-compose` that mounts the host FS and spawns a reverse shell. This provided persistent access while avoiding obvious HIDS-monitored file changes.

Task 12: Conclusion. Reflected on the complementary roles of NIDS and HIDS, the importance of curated rulesets, and the trade-offs between aggressive reconnaissance and stealth. Verified the new CTF scoring system which tracks IDS alerts and penalizes noisy actions.

Commands and Tools (representative)

- Active reconnaissance: `nmap -sV 10.201.102.178`, `nmap -sS`, Nikto with `-useragent` and `-e` evasion flags.
- OSINT: Shodan, search engine queries, WHOIS.
- Local enumeration: `sudo -l`, `groups`, `cat /etc/group`, linPEAS.
- Exploitation and persistence: Python exploit script (`exploit.py`), Docker with host mount, `docker-compose`.
- IDS/HIDS observation: Suricata alerts (network), Wazuh alerts/logs (host).

Pictures



Task 3 login page

CTFScore

Login Register

Register

Create a new account with the system here. Make sure to register the computers that you will use to interact with the CTF. The system uses this information to isolate attacks from different users. So, make sure that this information is correct if you want an accurate score.

If you're using Linux you will be able to retrieve a list of all the IPs associated with your node by running the following commands:

```
ip a
```

or:

```
ifconfig
```

If you're using Windows you can use:

```
ipconfig
```

Note that the IPs you register must be the ones associated with the adapter that will be used to interact with the CTF. Otherwise, no IDS alerts will be correctly processed. This IP should already be set as the first identifier.

Username:

Controlled IP Addresses:

Register

Register page

```
(kali123@kali)-[/]  
$ nmap -sV 10.201.102.178  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-18 16:10 IST  
Nmap scan report for 10.201.102.178  
Host is up (0.30s latency).  
Not shown: 996 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)  
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))  
3000/tcp   open  http      Grafana http  
8000/tcp   open  http      Unicorn  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 26.02 seconds
```

Nmap version scan

CTFScore

Dashboard Alerts Logout

Welcome, akh\$kali

Current Score:

622.910

Alert Stats

Total Number of Recorded IDS Alerts: 151
Highest Alert Score: 5.33
Average Alert Score: 4.13
Lowest Alert Score: 3

View All Alerts

IDS alert

IDS Details

- IDS Name: Suricata
- IDS Reliability: 8
- IDS Severity Range: 1-3*

*Note that, Suricata inverts the normal severity scale so an alert with a severity of 1 is, the most critical whereas, an alert with severity of 3 is not important. The scoring system does account for this.

IDS Details

```
(kali23@kali):[/]  
$ nikto -p 3000 -T 1 2 3 -useragent="Mozilla/5.0 (X11; U; Linux x86_64; de; rv:1.8.1.1) Gecko/20061223 BonEcho/2.0.0.0" -h 10.201.102.178  
- Nikto v2.5.0  
  
+ Target IP: 10.201.102.178  
+ Target Hostname: 10.201.102.178  
+ Target Port: 3000  
+ Start Time: 2025-10-18 17:14:50 (GMT5.5)  
  
+ Server: No banner retrieved  
+ Root page / redirects to: /login  
+ No CGI Directories found (use '-C all' to force check all possible dirs)  
+ /robots.txt: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
```


Nikto scan



Welcome to Grafana

Email or username

Password

[Forgot your password?](#)

Grafana webpage


```

(kali123@kali)-[~]
$ python3 exploit.py -u 10.201.118.46 -p 3000 -f /etc/grafana/grafana.ini | grep "grafana-admin"
admin_user = grafana-admin

(kali123@kali)-[~]
$ python3 exploit.py -u 10.201.118.46 -p 3000 -f /etc/grafana/grafana.ini | grep "password"
# You can configure the database connection by specifying type, host, name, user and password
# If the password contains # or ; you have to wrap it with triple quotes. Ex """"#password;""""
;password =
# default admin password, can be changed before first start of grafana, or in profile settings
admin_password = GraphingTheWorld32
;password_hint = password
# If the password contains # or ; you have to wrap it with triple quotes. Ex """"#password;""""
;password =
; basic_auth_password =
;password =

```

Finding admin user and password



You did it! 🎉 Intrusion Detection complete!

Points earned 🎯 144	Completed tasks ✅ 12	Room type 👤 Walkthrough	Difficulty 📊 Medium	Streak 🔥 1
------------------------	-------------------------	----------------------------	------------------------	---------------

👤👤👤👤 80,494 users are actively learning this week

Conclusion