**(SIEM)Security Incident and Event Management**

**What is a SIEM?**

SIEM (pronounced "sim") stands for Security Incident and Event Management. As the name suggests, a SIEM is more than a collection of aggregated logs like a syslog server. Systems feed their logs directly or indirectly into a SIEM, which assesses them to find likely security events. Very often, a log aggregator is used to normalize (reformat) data before that data is sent to the SIEM. Security professionals use a SIEM to visualize security events across the entire enterprise and search for indicators of compromise (IOC) or attacks.

SIEMs are used for several key security functions.   
**Uses of SIEMs:**

✅ **Aggregating logs** – Collecting data from various sources for analysis.  
✅ **Normalizing data** – Standardizing logs for easier correlation and analysis.  
✅ **Visualizing security events** – Providing dashboards and reports for security insights.  
✅ **Searching for IOCs (Indicators of Compromise)** – Detecting signs of cyber threats.

**Wazuh (Elastic-SEIM) LAB**

**Part 1: Explore Wazuh**

In this project we will apply what we have learned so far to explore the Wazuh SIEM, analyze log events, and focus your analysis using search and filters. We will then simulate an attack and use Wazuh to analyze the log events generated by the attack.

1. On the Welcome to Wazuh splash screen, click **Log In** to log in.A login screen with a blue background

   AI-generated content may be incorrect.
2. When prompted to Select your tenant, click **Cancel**.
3. In the Security Information Management section, click **Security events** to open the Security Events Dashboard.
4. In the Date range section (top right), we will select **Last 24 hours**, converting it to *~ a day ago -> now*, then click ***~ a day ago*** to set the beginning of the date range filter.
5. Click the **Absolute tab** and set the Start Date to **Apr 7, 2025 @ 00:00:00.000**. A screenshot of a computer

   AI-generated content may be incorrect.
6. In the Date range section, select **now** to set the end of the date range filter. A screenshot of a chat

   AI-generated content may be incorrect.
7. Click the **Absolute tab** and set the end date to **Apr 7, 2025 @ 23:30:00.000**.
8. Click **Update** to apply your changes, showing only security events in the date range selected. A screenshot of a computer screen

   AI-generated content may be incorrect.

In this timeframe, there should be over 700 security events!

1. Review the **Alert Level Evolution chart** in the Data Visualization section. This chart shows that there are level 3 through level 8 events. By default, events are rated from 1 to 13, with 13 being the most serious.
2. Scroll down to the **Security Alerts list** at the bottom of the page. There are 75 pages of alerts in this time period.You should notice that many of the security alerts are related to Aparmor which is a Linux application security system.
3. Click one of the **Aparmor events** to expand it, then we will explore the details of this alert. In the next step, we will clear out these alerts to see if there is something else to focus on. A screenshot of a computer

   AI-generated content may be incorrect.
4. In the Search field at the top of the dashboard, type **not rule.description : Apparmor**\* and click **Update** to hide these alerts.

Notice that most events still remain after applying this search filter. The "not" search is a good way to clear noise. A screenshot of a computer

AI-generated content may be incorrect.

1. Below the search field, click **Add filter** to open the Edit Filter tool. A screenshot of a computer

   AI-generated content may be incorrect.
2. In the Edit Filter tool, we will set the Field to **rule.level**, set the Operator to **is not**, and set the Value to **3**, then click **Save** to apply the filter.

We now see that nearly 400 events remain. A screenshot of a computer screen

AI-generated content may be incorrect.

1. Add another filter to remove all **level 4** events.
2. Add another filter to remove all **level 7** events,

When a SIEM is first deployed, it will generate many events, most of which look scary, but are often not critical issues. While there may be important level 3, 4 and 7 events, i**t is good practice to review the highest level threats first, and then work back.**

In the Security Alerts list, we can see the remaining events indicate that the network card has entered into promiscuous (sniffing) mode. This can indicate that a network tool like Wireshark was used, or is could indicate something more intrusive.

In this lab, the alert is likely due to Docker, and is not malicious.

*Take note of* ***how many events remain****. You will need this information to answer one of the questions on the* ***Tasks*** *tab. 31*

**Part 2: Simulate an Attack**

In this part of the lab, we will simulate an attack and use the SIEM to discover what has happened.

1. Clear all **filters and searches**.

We should see over 700 Security Events.

1. Open a command-line, click the **Terminal Emulator icon** to open a terminal window.
2. In the open terminal window, we will run **./attack.sh** and press Enter to simulate an attack on the remote lab server. A screen shot of a computer

   AI-generated content may be incorrect.

When the attack completes, you will see **READY** in the terminal window.

**Note:** The simulated took **5 minutes** to complete.

1. When the attack is complete, we will return to the **browser window** and click the **calendar icon** (next to the Date range) and select **Last 15 minutes**. A screenshot of a computer screen

   AI-generated content may be incorrect.

Notice that there are many new recent events. In the **Top MITRE ATT&CKS** visualization on the left, we can see that most events relate to Brute Force.

1. Click the part of the Top MITRE ATT&CKS chart relating to **Brute Force** to filter on these events. A screenshot of a computer

   AI-generated content may be incorrect.

You should notice that under the Search field, a new filter was added:

**rule.mitre.technique: Brute Force**

In the Security Alerts section, we see both level 10 and 5 alerts. There are also other alerts, but we can ignore them for the purpose of this exercise.

In theory, level 10 alerts are **more serious**, as 10 is greater than 5. In this case, however, the level 10 alerts are based on aggregated data. Knowing that a user missed their password more than once is important, but it does not tell us everything we need to know. We can think of it like an alarm clock. We react to the alarm, but still need the data (the actual time, not the noise).

1. In the Security Alerts section, expand the details for one of the **level 5 sshd: authentication failed alerts**, we will then review the **full log**. A screenshot of a computer

   AI-generated content may be incorrect.

By itself, this one event is not very important, but collectively, logs like this can provide essential details on an attack.

*Note the* [*\*\*rule.id*](http://rule.id)*(5716)*\* value for the level 5 "sshd: authentication failed" alert. Also,

*note of the location of logs (/var/log/auth.log) for the level 5 "sshd: authentication failed" alert.*

1. In the Search field at the top, we will enter the following: **full\_log : "Failed password for root" AND rule.description : "sshd: authentication failed."** and click **Update** so we can filter for events related to authentication failures for the root user. A screenshot of a computer

   AI-generated content may be incorrect.

We should see 137 authentication failures for the root user.

In the Security Alerts section, the last authentication failure is the first event displayed.

1. Note of the time(23:55:06.434) of the last authentication failure, then click the **arrow next to Time** to re-sort the events.

Now, the first authentication failure is the first event you see.

Note the time of the first authentication failure (23:50:16.178). While the times will vary slightly, there were ~137 login attempts for the root user in under 5 minutes. This is very telling of an automated brute-force password attack. By focusing on the individual events, we can identify the start and end of the attack, as well as the number of attempts made. This is key information when reporting an incident.

Another critical piece of information is the origin of the attack. In this lab, we know the attack came directly from our own workstation after we ran the simulated attack script. In a production environment, you would want to include this information in your search

( full\_log:"Failed password for root from [IP ADDRESS]" ).

**Summary**

In this lab, we learned to navigate the primary features of the Wazuh SIEM to analyze log events.

**CHALLENGE EXERCISE**

In this challenge exercise, we will run another script to simulate a different attack, then use the results in the SIEM to answer the questions below. A screenshot of a computer

AI-generated content may be incorrect.

In the browser window, we will clear all filters and searches. In the open terminal window, we will run **./.challenge.sh** . We will wait for the attack to complete, then return to the browser window and set the date to Last 30 minutes. Also, for the purposes of this exercise, we will exclude rule levels 3, 4, 7, and 10. We will also use a search phrase “Attempt to login a not existent user.”

**TASKS**

1. What is the Rule ID for the “sshd: Attempt to login using a non-existent user” alerts?

**5710**

1. What was the total number of login attempts for a non-existent user?

**86**

A screenshot of a computer

AI-generated content may be incorrect.