

CYBER FORGE

# SOC Analyst Workbook

November 2026 | [Wazuh SIEM](#) | [cyberforge.kennethnweke.com](#)

---

This workbook walks you through five hands-on tasks using the live Wazuh dashboard at **cyberforge.kennethnweke.com**. Most alerts come from real, publicly available historical log data; the final task uses a short, crafted attack scenario so you can practise reconstructing a full incident. The skills transfer directly to real SOC analyst work.

Each task gives you an objective, the dashboard steps, and DQL (Dashboard Query Language) queries to get started. Try to work out each answer yourself before checking the Answer Key at the back.

Kenneth Nweke · Network & Cybersecurity Engineer

## Before You Start

---

Log in at **cyberforge.kennethnweke.com** with the shared guest credentials shown on the Cyber Forge page. The account is read-only: you can explore and query freely, but you can't change any settings, so there's nothing to break.

### Where things live:

- **Security events** (left menu): the main alert feed. Most tasks happen here.
- The search bar accepts **DQL**. You can filter on any field in an alert, for example `rule.id:5710`.
- Click any alert to expand it and see every field Wazuh extracted, such as `srcip`, `rule.description` and `rule.mitre.id`. Those fields are exactly what you can query on.
- The time picker (top right) sets the window. Use **Last 24 hours** or wider. The lab data replays continuously, so alerts are always arriving.
- **Severity cards** at the top of Security events show Critical (level 15+), High (12-14), Medium (7-11) and Low (0-6) counts. Click one to filter.

Because the data loops, the totals you see are multiples of a single pass. Compare rankings (which IP is highest), not exact counts.

### A quick DQL primer:

|                                                  |                     |
|--------------------------------------------------|---------------------|
| <code>rule.id:5710</code>                        | match an exact rule |
| <code>srcip:"183.62.140.253"</code>              | match an exact IP   |
| <code>rule.level &gt;= 12</code>                 | numeric comparison  |
| <code>rule.groups:"authentication_failed"</code> | match a group       |
| <code>rule.id:5710 and srcip:"1.2.3.4"</code>    | combine with 'and'  |
| <code>rule.id:5710 or rule.id:5503</code>        | combine with 'or'   |

## TASK 1

# Triage - Find the SSH Brute-Force Source

**Objective:** Identify which IP address is responsible for the SSH brute-force activity, and determine whether it ever got in.

---

## Steps:

Search:

```
rule.id:5710
```

- This rule fires on "*sshd: Attempt to login using a non-existent user*", a core brute-force signal.
- Look at the `srcip` field (click it in the left field list to see the top values) and find which single IP appears far more often than any other.
- Then check whether that IP ever succeeded:

```
rule.groups:"authentication_success" and srcip:""
```

**Write down:** (a) the attacking IP, and (b) whether it ever logged in successfully.

**Bonus:** find the High-severity alert that flags this as a *sustained* brute force. How does it differ from the individual alerts?

## TASK 2

# Cross-Reference a Second Incident

**Objective:** Decide whether a separate set of authentication failures, logged through PAM rather than sshd, is part of the same attack as Task 1.

---

## Steps:

```
rule.id:5503
```

- This rule is "*PAM: User login failed*": a different detection path from a different log source.
- Find the top offending `srcip`, the same way as in Task 1.
- Compare it with the IP you found in Task 1.

**Write down:** the top IP here, whether it matches Task 1, and what that tells you about whether these are one campaign or two unrelated incidents.

### TASK 3

## Threat Hunting - Which Successful Login Is the Compromise?

**Objective:** The data contains more than one successful SSH login. Hunt them down and decide which one, if any, is malicious.

---

### Steps:

```
rule.groups:"authentication_success"
```

- This is the essence of threat hunting: you aren't starting from an alert someone flagged as severe. You're going looking for something specific.
- List every distinct `srcip` and `srcuser` that logged in successfully.
- For each one, check whether the same IP also appears in failed-login alerts just before the success:

```
srcip:"" and rule.groups:"authentication_failed"
```

**Write down:** each successful login, and for each whether it followed a burst of failures (suspicious) or stands alone (probably legitimate). What made you decide?

#### TASK 4

## Web Server Investigation - Scanning vs. Targeted Attack

**Objective:** Investigate repeated access attempts against the web server and decide whether this is one attacker or distributed scanning.

---

### Steps:

```
rule.id:30106
```

- This rule is "*Apache: Attempt to access forbidden directory index*": a different data source and attack surface from Tasks 1-3.
- Note the exact path being targeted, and look at how many *distinct* source IPs appear.
- Compare the shape of this activity with Task 1. There, one IP made hundreds of attempts. Here, is it a few IPs repeating, or many IPs each trying once?
- Find the High-severity alert about distributed web scanning. What threshold triggered it?

**Write down:** the targeted path, whether sources repeat, and your conclusion: targeted attacker or automated scanning? Justify it from the data.

## TASK 5

# Incident Response - Reconstruct the Attack Chain and Report It

**Objective:** Work the critical-severity alerts, rebuild what the attacker did step by step, and write a short handoff report.

---

## Steps:

```
rule.level >= 15
```

- Open the critical alert and note the source IP. Then pull everything related and sort by time, oldest first, to read the story in order:

```
srcip:"" or rule.groups:"cyberforge"
```

- Work out the sequence: how did the attacker get in, what did they create to keep access, how did they escalate, and how did they try to hide it?
- For each step, record the rule that fired and its `rule.mitre.id`.
- Write a short paragraph (4-6 sentences): what happened, which host was affected, the techniques involved, and your recommended actions (for example: isolate the host, disable the new account, reset credentials, preserve evidence).

There is no single correct wording. The goal is turning raw alerts into a decision someone else can act on without redoing your investigation.

## ANSWER KEY

# Check Your Work

---

Verified against the dataset. Use this to confirm your findings, not as a shortcut past the tasks.

| Task                                     | Answer                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1 - SSH brute-force source</b>        | <b>183.62.140.253</b> : by far the most attempts (about 3.5x the next IP in the source data). It never logged in successfully. Bonus: the High-severity rule groups 30+ invalid-user attempts from one source inside 2 minutes.                                                                                                                       |
| <b>2 - Second incident (PAM)</b>         | <b>150.183.249.110</b> . A different IP from Task 1, so a separate, unrelated incident.                                                                                                                                                                                                                                                               |
| <b>3 - Which login is the compromise</b> | <b>root from 203.0.113.77</b> is the compromise: it logged in right after 10 rapid failures. <b>fztu from 119.137.62.142</b> stands alone with no failure burst, so it is probably legitimate.                                                                                                                                                        |
| <b>4 - Apache scanning</b>               | Target path <b>/var/www/html/</b> . Each probe comes from a different source IP, which points to distributed or automated scanning rather than one attacker. The High rule fires at 15+ probes in 5 minutes.                                                                                                                                          |
| <b>5 - Attack chain (203.0.113.77)</b>   | 1) Brute force, then root login (T1110.001, T1078, <b>critical</b> ). 2) New account <b>support_svc</b> created (T1136.001). 3) Added to sudo and a root shell opened (T1548.003). 4) <b>shred</b> run on /var/log/auth.log to erase evidence (T1070.002). Recommended: isolate the host, disable support_svc, reset root credentials, preserve logs. |