**KYPO Scenario Implementation Plan: NG-SOC Module 8 – Incident Response & Forensics**

**Roles and Module Alignment**

The scenario will incorporate **five key cybersecurity roles** mapped to the five exercises in Learning Module 8 – *Incident Response and Forensics*. Each exercise places the learner in a different role with tasks reflecting that role’s perspective on the same cyber incident. The roles and their corresponding exercises are:

* **Exercise 1 – SOC Analyst:** Focus on initial detection and monitoring. The learner reviews SIEM alerts and logs, identifies a potential incident, and escalates it for response.
* **Exercise 2 – Incident Responder (IR):** Emphasis on containment and forensic evidence collection. The learner isolates affected systems, acquires disk/memory images, and eradicates malware on the compromised host.
* **Exercise 3 – Cyber Threat Intelligence (CTI) Analyst:** Focus on enrichment of the incident with threat intel. The learner pivots on indicators (e.g. malicious IPs, hashes) using a Threat Intelligence Platform (MISP) to identify threat actors, TTPs, or related campaigns, and feeds this back into the investigation.
* **Exercise 4 – Penetration Tester:** Emulates adversary techniques and assesses vulnerabilities. The learner replicates the attacker’s likely exploit (in a safe manner) using penetration testing tools (e.g. from a Kali VM), validates the exploited vulnerability or misconfiguration (CVE), and tests that defenses can now detect/prevent such an attack.
* **Exercise 5 – IR Coordinator:** Focus on big-picture coordination and post-incident management. The learner, acting as the CSIRT coordinator, compiles findings from all teams, documents the incident timeline, ensures each IR phase was completed, and formulates a post-incident report with lessons learned.

This mapping ensures coverage of **the full incident lifecycle** from detection through recovery, aligning with Module 8’s learning goals of understanding IR phases and roles. It also gives the learner hands-on experience in each role’s responsibilities while **mirroring a coordinated CSIRT response**.

**Exercise 1: SOC Analyst – Detection & Escalation**

*You are a Security Operations Centre (SOC) Analyst monitoring your company’s security alerts. Your goal is to* ***detect a potential incident*** *and* ***escalate it*** *for response.*

**Step 1:** Access the **SIEM dashboard** on the SOC server to review alerts. In the KYPO platform, open the SOC server’s web interface for Kibana (Wazuh SIEM). If a login screen appears, log in with the provided credentials (e.g. *admin / admin* for Kibana).

**You should see:** The Kibana dashboard showing recent security alerts. Look for any **high-priority alert** that indicates suspicious activity, such as unusual logins or malware detection.

**Tip:** Ensure the dashboard’s time range is correct (e.g. “Last 24 hours”) so you don’t miss the alert. If nothing is showing, refresh the page or check that you opened the correct SIEM app.

**Step 2:** Identify the **notable alert** and investigate its details. Click on the alert to drill down into log data. For example, you might see an alert about a user PC communicating with an unknown IP address or an antivirus event.

**You should see:** Detailed log information related to the alert. For instance, it may show that around **10:30** this morning, the user workstation (e.g. user-pc at IP 10.10.10.50) executed a file named **invoice.pdf.exe** and made a connection to an external IP **192.0.2.123** (the attacker’s server).

**Tip:** Common indicators of a breach include odd program names (like a PDF file ending in .exe), or connections to IPs you don’t recognise. Make a note of any suspicious **Indicators of Compromise (IOCs)** such as file names, file hashes, IP addresses, or domains that appear in the alert.

**Step 3:** **Collect initial evidence** from the SIEM. From the logs, copy or write down the key IOCs you found. In our case these are likely:

* The suspicious file’s **name** (e.g. invoice.pdf.exe) and possibly its **hash** (a long string if provided by the SIEM).
* The attacker’s **IP address** (e.g. 192.0.2.123).
* Any affected **hostnames** (e.g. the user’s PC name).

**You should see:** In the log details, there may be a field for file hash (e.g. an SHA-256 hash) if the endpoint protection logged it. It could look like a long alphanumeric string. For example: File SHA-256: 3d2e467.... Make sure to note it correctly if present. You will use these details to raise an incident.

**Tip:** If the SIEM doesn’t show the file hash directly, you might extract it later from the file itself. But at minimum, record the filename and the malicious IP. These will form the basis of your incident report.

**Step 4:** **Escalate the incident** by creating a case in TheHive (an incident response management system). On the SOC server, open TheHive web interface (often via a browser on the SOC VM, e.g. http://localhost:9000 or a link provided in KYPO). Log in with the provided TheHive credentials (for example, *admin / <password>* as given by your instructor).

**You should see:** TheHive dashboard. Navigate to “Cases” and start a **New Case**. A form will appear for entering incident details.

**Step 5:** Fill in the **incident case form** in TheHive:

* **Title:** Give the case a name, e.g. *“Incident 2025-07: Malware outbreak on user-pc”*.
* **Description/Summary:** Briefly describe what you detected. For example: *“SOC alert at 10:30 identified suspicious file execution (invoice.pdf.exe) on user-pc and outbound traffic to IP 192.0.2.123. Suspected malware infection via phishing.”*
* **Severity/Priority:** Choose an appropriate level (e.g. *High* if a breach is likely).
* **Affected assets:** List the affected host (e.g. user-pc, and possibly the internal server if you suspect it was targeted).

**You should see:** Fields to add **Observables** (indicators). Add the IOCs you collected:

* Create an **“Observable”** for the suspicious file hash (choose type *File hash* and paste the value).
* Add another observable for the attacker IP (choose type *IPv4* and enter 192.0.2.123).

**Tip:** Mark these observables as *IoC* (Indicator of Compromise) if TheHive asks. This flags them as important indicators that other tools (like Cortex or MISP) can use for analysis later.

**Step 6:** **Save and assign** the case. Once you have filled in all details, save the new case. If prompted, assign it to the appropriate team or simply to “Incident Response”. This step simulates handing off the case to the incident response team.

**You should see:** The case appears in TheHive’s case list with a unique case number. Inside the case, you should see the details you entered (description, observables, etc.). The incident is now officially escalated and ready for the IR team to handle.

**Tip:** In a real SOC, at this point you would notify the on-call Incident Responder or IR coordinator about the new case. TheHive might send a notification automatically. For now, proceed to the next phase knowing the IR team has the info.

**Exercise 2: Incident Responder – Containment, Eradication & Forensics**

*Now you are the* ***Incident Responder****. The SOC has handed over the case with initial findings. Your task is to* ***contain the threat****,* ***eradicate the malware****, and* ***collect forensic evidence*** *from affected systems.*

**Step 7:** **Access the compromised host (user-pc).** This is the Windows 10 workstation that was identified as infected. Use Remote Desktop (RDP) or the KYPO web console to connect to **user-pc** (IP 10.10.10.50). Use the provided credentials (for example, Username: Administrator, Password: provided in scenario) to log in.

**You should see:** The Windows desktop of the user’s PC. It should look like a normal user workstation environment. From here, you can investigate running processes and files.

**Tip:** If RDP isn’t available, KYPO might provide a VNC or SSH connection to the Windows host. Follow the platform instructions to access the machine’s console. It may take a moment for the desktop to load.

**Step 8:** **Find the malicious process** running on the user’s PC. Open **Task Manager** (e.g. right-click the taskbar and select *Task Manager*, or press Ctrl+Shift+Esc). Click “More details” to see all processes. Look for any unusual or suspicious process name. For example, if you see a process called **evil.exe** (or the same name as the suspicious file, e.g. invoice.pdf.exe), that is likely the malware.

**You should see:** A process in the list that stands out, possibly with an odd name or running from an unusual location. In our scenario, assume **evil.exe** is the malware process. It might be consuming CPU or memory, or just idle.

Image Name PID Session Name Mem Usage

-----------------------------------------------

evil.exe 1234 Console 15,000 K

**Tip:** Sort processes by name or CPU to help spot malicious processes. Malware often doesn’t have a familiar publisher or icon. If you are unsure, cross-check the process name against known Windows processes (for instance, explorer.exe and svchost.exe are legitimate; evil.exe is not!).

**Step 9:** **Contain the threat by killing the malicious process.** In Task Manager, select the suspicious process (e.g. evil.exe) and click *“End Task”*. Confirm if prompted. This will stop the malware from running (at least until the next reboot or if it has persistence).

**You should see:** The process disappear from the Task Manager list. The malware is no longer actively running. This is an immediate containment step to halt any ongoing malicious activity on the host.

**Warning:** Simply killing the process doesn’t remove the malware permanently. Avoid rebooting the machine at this stage (some malware might restart on boot). We will fully remove it soon. Also, by terminating the malware quickly, you reduce further harm but ensure you capture evidence **before** removing it completely.

**Step 10:** **(Optional) Isolate the host from the network.** To prevent the attacker from controlling this machine or the malware from spreading, you might disconnect the machine from the network: for example, unplug the network (if physical) or disable the network adapter in Windows. One quick method on Windows is to open an Administrator Command Prompt and run:

ipconfig /release

This will drop the IP address (making the machine offline).

**You should see:** The machine’s network status change (you might see a network disconnected icon). The user-pc will no longer have network connectivity.

**Warning:** If you disable the network while connected via RDP/VNC, **you will lose your connection!** Only isolate the machine after you’ve killed the malicious process and if you have another way to reconnect (or no further live response is needed from that machine). If you isolate it and get locked out, you may need to use the platform controls to re-enable the network or skip this physical isolation in a virtual exercise.

**Tip:** In many cases, responders hold off full network isolation until after collecting volatile evidence (like memory) to avoid losing access. Here, if you choose not to disconnect the network, assume the malware is contained by ending its process. Proceed with evidence collection but be mindful the machine is still theoretically online.

**Step 11:** **Perform a memory dump of the infected host.** A memory dump preserves volatile evidence (running processes, malware in memory) for later analysis. Use a tool provided on the user-pc for memory capture. For example, if WinPmem or DumpIt is available on the desktop or in the tools folder:

* Run the memory capture tool as Administrator. (For WinPmem, you might open Command Prompt as Administrator and execute winpmem.exe with appropriate flags, or simply double-click DumpIt if that’s provided.)
* Follow any on-screen instructions to save the memory dump. Choose a destination (the tool might save a file like memory.dump to a specified folder).

**You should see:** The tool running and then a confirmation that a memory image has been saved (e.g. a file like memory.raw or dump.bin will appear). The file might be several hundred MBs or more, depending on RAM size.

**Tip:** This step may take a few minutes and the file will be large. Ensure you have enough disk space (the exercise environment should have it). Label the memory dump clearly if you can (for instance, include the hostname and date in the filename, if that option exists). In a real case, you would later analyse this with forensic tools, but we will rely on simpler indicators for this exercise.

**Step 12:** **Secure a copy of the suspicious malware file from disk.** Now, find the file invoice.pdf.exe (as identified in Step 2) on the user’s PC. Common places to check: the Downloads folder or a temporary directory. In our scenario, the file was likely dropped in the user’s Temp folder.

* Open File Explorer and navigate to the user’s Temp directory (e.g. C:\Users\<Username>\AppData\Local\Temp). Look for **invoice.pdf.exe** or any file with a similar name and an *.exe* extension.
* Once found, **copy** this file to a safe location for analysis. For example, create a folder C:\IR\_Evidence and paste the file there. Alternatively, if a network share or the SOC server is accessible, you could transfer the file there for analysis (ensure it’s a controlled location).

**You should see:** The malicious file invoice.pdf.exe in its original location. After copying, you will have a duplicate of this file in your evidence folder. **Do not double-click or run the file!** It’s malicious – you only need a copy of it.

**Tip:** To be extra careful, you might compress the malware file into a password-protected ZIP (password “infected”) before moving it, to avoid accidental execution. In a real-world scenario, always handle malware samples carefully. For this exercise, simply copying is fine.

**Step 13:** **Check for persistence mechanisms** that could allow the malware to survive reboots. A common persistence on Windows is a Registry Run key or a scheduled task that relaunches the malware. We will inspect a typical location in the registry:

* Open the Start menu, type **regedit** and press Enter to launch the Registry Editor. **Be cautious** when navigating the registry.
* Navigate to: HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Run. (This is where programs set to run on login for the current user are listed.)
* Look for any entry that references the malware, e.g. a value with name like “Updater” or random text that has data pointing to invoice.pdf.exe or evil.exe.

**You should see:** If the malware set up persistence, there will be an entry under that Run key (or sometimes HKEY\_LOCAL\_MACHINE\Software\...Run for all users). For example: an entry named "**EvilApp**" with data C:\Users\<User>\AppData\Local\Temp\invoice.pdf.exe. This would mean the malware tries to start every login.

**Step 14:** **Remove the persistence entry** to prevent the malware from restarting. In Registry Editor, right-click on the suspicious value (the one pointing to the malware file) and choose **Delete**. Confirm the deletion.

**You should see:** The malicious run key entry is gone from the registry. Now the malware will not automatically start if the system reboots or the user logs in again.

**Tip:** Persistence can also hide elsewhere (Scheduled Tasks, Services, startup folders). For this scenario, assume the registry Run key was the only persistence. Removing it is an important eradication step. Always double-check you’re deleting the correct entry – do **not** delete random keys, as it may affect the system.

**Step 15:** **Investigate the internal server for signs of compromise.** The initial alert suggested the attacker also attempted to exploit an internal server (internal-server at IP 10.10.10.60). Now that the user PC is handled, connect to the Linux server to see if it was affected:

* Use SSH to connect to internal-server. For example, on the SOC server or Kali attacker, run:

ssh ubuntu@10.10.10.60

(Replace with the correct username provided, e.g. *ubuntu* with password or key as given in the scenario.)

* Once logged in, check for any **new user accounts or suspicious changes**. A quick way: open the system password file by running:

sudo grep -v "nologin" /etc/passwd

This lists users with login shells. See if there’s an unexpected account (e.g. an account that wasn’t supposed to be there). Our scenario suggests the attacker created a backdoor account.

**You should see:** If an attacker added a user, an unfamiliar username will appear in the output. For example, you might see a username like serviceman:x:1002:1002::/home/serviceman:/bin/bash which wasn’t there before. This indicates a rogue account. Also, check if any processes are odd (you can run ps -aux to list processes and look for unusual ones).

**Step 16:** **Neutralise any threat on the server.** If you found a suspicious user account, disable or remove it:

* For example, remove the account by running:

sudo userdel serviceman

*(Replace "serviceman" with the actual malicious username you found.)*

* Also remove any related files. If the user had a home directory or SSH keys, you can delete those (sudo rm -r /home/serviceman if applicable).
* Check for any malicious processes on the server (from ps output). If something like a reverse shell is running, kill it with sudo kill <PID>.

**You should see:** The malicious user account removed (you can cat /etc/passwd again to confirm it’s gone). Any suspicious process should no longer be running. At this point, the internal server is cleaned of obvious threats, and the user PC is contained and cleaned as well.

**Tip:** On Linux, also verify key config files for changes (e.g. /etc/ssh/sshd\_config for added users or backdoors). In our simple scenario, the main change was a new user. Removing it closes that backdoor. Always document what you remove or change.

**Step 17:** **Document your actions and findings in TheHive.** Now that containment and eradication are done, update the incident case for record-keeping and to inform others:

* Go back to TheHive case created in Exercise 1 (it should still be open on the SOC server’s browser).
* Add a **Case Note** (journal entry) describing what you did. For example: *“Contained infection on user-pc: terminated malicious process evil.exe, dumped memory, and collected malware sample. Removed persistence (registry key) and quarantined host. Investigated internal-server: found and removed suspicious user account.”* Include the time if possible for each action.
* In the case’s **Task** or **Status** section, mark relevant tasks as completed. If the case template has tasks like “Containment”, “Eradication”, or “Evidence Collection”, mark them *Done*. This shows you have finished those phases.

**You should see:** Your notes added to the case, and task statuses updated to completed. The case timeline now reflects detection (earlier) and the response actions you just performed.

**Tip:** Good documentation is crucial. It helps the next person (or your future self) understand what was done. In a real incident, these notes provide a chain of custody for evidence and a record for a post-incident report. Don’t worry about perfect wording – just clearly state actions and findings.

**Exercise 3: CTI Analyst – Threat Intelligence & Enrichment**

*In this phase, you act as a* ***Cyber Threat Intelligence (CTI) Analyst****. The incident has been contained, and you have collected IOCs (file hash, IP address, etc.). Your job is to* ***research these indicators*** *to find out who might be behind the attack and add that intelligence to the case.*

**Step 18:** Open the **Threat Intelligence Platform (MISP)** on the SOC server to research the IOCs. MISP (Malware Information Sharing Platform) is a tool for sharing and querying threat intel. In a web browser on the SOC server, navigate to the MISP interface (for example, https://localhost/misp or a provided link). Log in with the MISP credentials given (e.g. *admin / admin* or as provided by instructor).

**You should see:** The MISP dashboard or events list. There will be a search bar or menu for looking up indicators.

**Tip:** If you are unfamiliar with MISP, think of it as a database of known bad stuff (malware, threat groups, etc.). You can search by hash, IP, domain, etc. The platform might already have some data loaded relevant to our incident.

**Step 19:** **Search for the malware file hash** in MISP. Copy the hash of invoice.pdf.exe (from your earlier notes or TheHive observables) and enter it into the MISP search field. Start the search.

**You should see:** An event or indicator in MISP matching that hash if threat intel is available. For example, there might be an event titled *“Snake Malware – Trojan”*. Opening it could show details like: “**Malware Name:** Snake (a Remote Access Trojan), **Associated Threat Actor:** Cobra group, **First seen:** 2025-06, **Description:** spreads via phishing attachments.” It may also list the hash as an attribute and possibly other related IOCs.

**Tip:** The names “Snake” and “Cobra” here are just examples. Threat intel often gives fancy codenames to malware (Snake) and hacker groups (Cobra). Don’t be thrown off by the names – focus on the details linking them to your IOCs. If the hash search returns nothing, try searching by the filename or other indicators.

**Step 20:** **Look up the attacker’s IP address** in MISP or via OSINT tools. Similarly, search MISP for 192.0.2.123 (the suspicious IP). If MISP has no info, you might use an OSINT lookup (some cyber ranges provide tools or you might simulate this):

* In MISP, an IP search might show if that IP is in any event (e.g. it could appear in the same Snake malware event as a Command-and-Control server).
* Alternatively, you could use an external lookup like a WHOIS or an online blacklist (only if the environment allows internet access or if pre-loaded data is available). For this exercise, assume MISP knows about it.

**You should see:** If the IP is known, information linking it to malicious activity. For example: *“192.0.2.123 – identified as Command-and-Control server for Snake RAT, used by Cobra threat group.”* This ties the IP to the same threat you found via the hash. If available, note which country or provider the IP is associated with, as it might be mentioned (e.g. an ISP or a hosting service often used by that group).

**Tip:** Piecing together intel: By now you likely learned the malware (invoice.pdf.exe) is a RAT called *Snake*, and it’s used by a group dubbed *Cobra*. The IP 192.0.2.123 is one of their control servers. This suggests a **targeted attack** by an advanced adversary, not just random malware. Such context is valuable for your report and for possibly alerting authorities or industry peers.

**Step 21:** **Research the server vulnerability** that was exploited. From the earlier steps, we suspect the internal server was compromised via an unpatched vulnerability (say in Apache web server). Let’s identify it:

* Recall from our penetration testing step (upcoming) or from intel: the likely vulnerability was **CVE-2025-12345** affecting Apache 2.4.10. Use MISP or a provided CVE database to look up this CVE.
* In MISP, search for “CVE-2025-12345”. If there’s info, it might show severity, description, and related threat actors or malware. If not, you might have a PDF report in the materials.
* Note what the vulnerability is (e.g. *“Apache path traversal leading to remote code execution”*), its severity (Critical), and any known usage by attackers (perhaps the Cobra group started exploiting it in the past month).

**You should see:** Details confirming that the server’s Apache version was vulnerable. For instance: *“CVE-2025-12345: Apache HTTP Server 2.4.10 remote code execution vulnerability. Patch available in Apache 2.4.52. Known to be exploited by advanced threat actors in mid-2025.”* This aligns with how the attacker got in.

**Tip:** Understanding the **root cause** (the unpatched CVE) helps justify recommendations later (patch your servers!). Even if you’re not deeply familiar with CVE details, knowing that an exploit existed for the software version running on the server is enough for now.

**Step 22:** **Enrich the case with your findings.** Return to TheHive and update the incident case with the threat intelligence information:

* Add a new **Case Note** or update the existing notes to include intel. For example:  
  *“Threat Intelligence analysis: The malware sample (hash ending in …abcd) is identified as* ***Snake RAT****, associated with the* ***Cobra*** *threat group. The attacker’s IP (192.0.2.123) is a known Cobra Command-&-Control server. The internal server exploit corresponds to* ***CVE-2025-12345****, which Cobra has used recently.”*  
  *“This suggests a targeted attack by a known group. Recommend sharing these IOCs with our industry CERT.”*
* If TheHive is integrated with MISP, you can also **link observables to MISP** or import the MISP event. (For instance, mark the observables and click “Enrich” or “Export to MISP” if available, so the case now has that contextual info attached.)

**You should see:** TheHive case now contains your intel notes. The observables (IP, hash) might show extra info or tags (like the malware name or threat actor) if you enriched or tagged them. The case is now richer in detail, which will help in the final report.

**Tip:** In a real environment, a CTI analyst might also share these findings with external partners. For our exercise, simply noting that “IOCs have been shared” or would be shared is enough. You could simulate this by marking the observables with a tag like TLP:AMBER (if you intend to share under certain restrictions) or just writing that you would notify law enforcement if appropriate.

**Exercise 4: Penetration Tester – Vulnerability Analysis & Security Testing**

*Next, step into the shoes of a* ***Penetration Tester*** *(ethical hacker). The incident is contained, and now you need to* ***verify the vulnerability*** *that the attacker exploited and ensure it’s patched. You’ll simulate the attack to confirm the root cause and test defenses.*

**Step 23:** **Reconnaissance – scan the internal server for vulnerabilities.** Use the attacker machine (Kali Linux) provided in the scenario to replicate how an external attacker would scan the network:

* Access the **Kali VM** (it might be labeled “attacker” in KYPO). This could be through a VNC session or SSH (e.g. ssh debian@10.10.30.10 if allowed).
* Once you have a terminal on Kali, run an Nmap scan on the internal server:

nmap -sV -p1-10000 10.10.10.60

This scans common ports (1–10000) on the internal-server and attempts to detect service versions (-sV).

* Wait for the scan to complete (it may take a minute or two).

**You should see:** Nmap’s output listing open ports and services on 10.10.10.60. For example, it might show something like:

PORT STATE SERVICE VERSION

80/tcp open http Apache httpd 2.4.10 ((Ubuntu))

22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu2.1 (Ubuntu Linux)

...

Focus on the **Apache 2.4.10** on port 80. This version is outdated and (based on our intel) vulnerable to CVE-2025-12345. Nmap might even list known vulnerabilities if certain scripts are enabled.

**Tip:** The scan confirms the server is running an old Apache. As a pen tester, you suspect this is the entry point. Always ensure you have permission in a real scenario before scanning – here it’s your own range, so it’s fine. If other ports show up, note them but concentrate on the one likely used by the attacker.

**Step 24:** **Identify the exact vulnerability (CVE).** Use an exploit database or search tool to find if an exploit exists for Apache 2.4.10:

* On Kali, run **Searchsploit** (a tool for searching known exploits) for Apache 2.4.10:

searchsploit Apache 2.4.10

* Alternatively, search by the CVE if you know it (CVE-2025-12345). The output will show if there are public exploits or references.

**You should see:** Search results indicating an exploit. For example: *“Apache HTTPD 2.4.10 - Remote Code Execution (Metasploit)”* or a reference to CVE-2025-12345. This confirms that an exploit exists for the version the server is running.

**Tip:** In practice, you’d also manually confirm the vulnerability (e.g. reading a CVE description). We’ve essentially done that in the intel phase. As a penetration tester, your next step is to safely exploit it to prove the risk.

**Step 25:** **Launch an exploit to verify the vulnerability.** We will use Metasploit, a penetration testing framework, which likely has a module for this exploit:

* Start Metasploit on Kali by typing msfconsole. Wait for the msf> prompt.
* Search within Metasploit for the exploit module:

search CVE-2025-12345

(Replace with relevant terms if needed, e.g. search Apache 2.4.10.)

* Load the chosen exploit module using the use command (for example, use exploit/linux/http/apache\_cve2025\_12345 – the actual name will be shown by the search).
* Set the target options:

set RHOSTS 10.10.10.60

set RPORT 80

(And any other required settings the module info shows, such as payload specifics. Often a reverse shell payload is default.)

* Execute the exploit by typing run (or exploit).

**You should see:** Metasploit output indicating it’s attempting the exploit. If successful, you’ll get a **Meterpreter session** or shell on the target. For example, it may say *“Meterpreter session 1 opened at 10.10.30.10:4444”* which means you now have remote control of the internal server as the attacker did. You can confirm access by running a simple command in the session, such as getuid or whoami (which might show a low-privilege user, e.g. www-data, indicating you’ve exploited the web server).

**Tip:** In a controlled environment, the exploit might just simulate success (perhaps by creating a file or giving a shell with limited commands). Even a simulation is fine – the key is you proved the vulnerability could be exploited. **Do not use the access to do anything destructive.** Just note that you were able to “own” the server, which is exactly what the attacker did initially.

**Step 26:** **(Optional) Test the security improvements.** Now that you know the flaw, let’s ensure it can be fixed:

* If possible, **apply a patch or mitigation** to the server. For example, you could update Apache to the latest version. On the internal server (if you still have shell access or via SSH), you might run:

sudo apt-get update && sudo apt-get install apache2

This could update Apache to a newer version (assuming the environment allows package installation).

* After patching, run the exploit again or re-scan with Nmap. The expectation is that the vulnerability is closed.

**You should see:** The exploit should **fail** after the patch (e.g. Metasploit might not open a session, or Nmap now shows Apache updated to a non-vulnerable version). If you updated Apache, you can check the version by running apache2 -v on the server – it should show a higher version number (e.g. 2.4.54).

**Tip:** If actual patching in the exercise isn’t feasible, mentally simulate this step. The important takeaway is that **patching the software removes the vulnerability**. In a real report, you would note that the system was updated and is no longer exploitable.

**Step 27:** **Perform a quick sweep for other vulnerabilities.** As a bonus, scan the user’s PC (10.10.10.50) for open ports or obvious issues (since an attacker could target multiple systems):

* Run a basic Nmap scan on the user PC:

nmap -sV 10.10.10.50

* Check if any high-risk services are exposed (for example, maybe port 3389 RDP is open, or an outdated service). In our scenario, assume nothing critical is exploitable on that host beyond the phishing malware which was already the entry.

**You should see:** Common ports on the user PC (perhaps RDP on 3389, file sharing on 445, etc.). No immediate critical vulnerabilities should stand out. If something does (like an old SMB service), note it down.

**Tip:** Even if these weren’t used in this incident, a good penetration tester records any weaknesses. For instance, “SMB v1 enabled on user-pc – while not exploited this time, recommend disabling or updating it.” This shows thoroughness and improves overall security.

**Step 28:** **Report the penetration test findings.** Summarise what you discovered and fixed, as this will feed into the final incident report:

* **Root Cause Confirmed:** Write down the primary vulnerability that led to the incident (e.g. *“Unpatched Apache 2.4.10 on internal-server (CVE-2025-12345) allowed remote code execution. Attacker used this to gain a foothold.”*).
* **Exploitation Replicated:** Note that you successfully reproduced the exploit, proving the weakness was real. (e.g. *“Confirmed exploit by obtaining a shell on the server via Metasploit.”*).
* **Additional Findings:** List any other issues found (e.g. *“Found port 445 open on user-pc (SMB service) – not exploited in this attack but should be reviewed.”*).
* **Fixes Implemented:** Document what was done to fix or mitigate issues (e.g. *“Updated Apache to latest version 2.4.54, which patches CVE-2025-12345. Verified exploit no longer works.”*).

Add these notes either in a separate **Pen Test Report** document or directly into TheHive case as a note or attachment. If TheHive has a section for “Observations” or you simply want to append to your previous notes, that’s fine.

**You should see:** Your report of findings saved. If in TheHive, the case now has a complete history from initial detection to vulnerabilities addressed. This will make writing the final report easier.

**Tip:** Penetration testing after an incident is often referred to as *“validation”* or *“lessons learned”* testing. It ensures the holes are truly closed. Always double-check that your actions (like patching) didn’t inadvertently cause issues – in our sandbox, it’s controlled, but in production, coordinate with system owners before patching.

**Exercise 5: IR Coordinator – Post-Incident Review & Reporting**

*Finally, you are the* ***Incident Response Coordinator****. The incident is resolved, and it’s time to* ***review, document, and communicate*** *what happened. You will create an incident timeline, write a summary report, and identify lessons learned.*

**Step 29:** **Compile the incident timeline.** Gather information from all previous steps (your notes in TheHive, logs timestamps, etc.) and list the key events in chronological order. This timeline should include detection, containment, eradication, recovery, and closure milestones. For example:

* **10:30** – SOC detects suspicious activity on user-pc (SIEM alert triggers).
* **10:40** – SOC Analyst creates incident case and escalates to IR team.
* **10:50** – Incident Responder kills malware process and begins containment on user-pc.
* **11:00** – Memory dump and malware sample collected from user-pc.
* **11:15** – Persistence removed on user-pc; user-pc isolated from network.
* **11:20** – IR investigates internal server; finds and removes malicious user account.
* **11:45** – Threat intel identifies malware as Snake RAT used by Cobra group; IoCs enriched.
* **12:00** – Patch applied to internal server (Apache updated).
* **12:10** – Penetration test confirms vulnerability is fixed (exploit attempt now fails).
* **12:30** – Incident declared contained and eradicated.
* **13:00** – Incident case closed in TheHive after documentation.

Write this timeline in the provided **Incident Timeline** template or text area in the KYPO platform (or on paper, if instructed). Make sure each entry has a time (even if approximate) and an event description.

**You should have:** A clear sequence of events from start to finish. This helps anyone reading the report quickly grasp what happened and how quickly it was dealt with.

**Tip:** Use 24-hour time format for clarity (as shown above). If you are unsure of exact times, use the timestamps from SIEM alerts or when you noted actions in TheHive. The exact times are less important than the order and inclusion of all major steps.

**Step 30:** **Write the Incident Report.** This is the formal summary of the incident and how it was handled. Structure your report in paragraphs covering the following:

* **Summary:** A high-level overview of the incident. E.g.: *“On 1 July 2025, Company X experienced a targeted cyber attack starting with a phishing email. An attacker (identified as the ‘Cobra’ group) compromised a user’s Windows PC via a malware attachment and then breached an internal server via an unpatched vulnerability. The incident was detected promptly by SOC monitoring and contained before major damage occurred.”*
* **Detection and Analysis:** Briefly describe how it was detected and what analysis was done. E.g.: *“SOC analysts observed an alert for suspicious file execution on a user PC and outbound traffic to an unknown IP. They escalated the incident and provided initial IOCs (malicious file hash and IP).”*
* **Containment and Eradication:** Summarise the IR actions. E.g.: *“Incident responders isolated the affected PC, terminated the malicious process, collected forensic evidence (memory image and malware sample), and removed the malware and its persistence. They also checked the internal server, removing a backdoor account left by the attacker.”*
* **Recovery and Patching:** E.g.: *“The team identified the root cause on the internal server (outdated Apache software) and applied the necessary patch (upgrading Apache to 2.4.54) to close the vulnerability.”*
* **Threat Intelligence Findings:** E.g.: *“Threat intel analysis revealed the attack was likely by a known threat actor (Cobra group) using Snake malware. This provided context about the attacker’s motives and confirmed the vulnerability exploited (CVE-2025-12345) was recently leveraged by this group.”*
* **Testing and Validation:** E.g.: *“A follow-up penetration test was conducted. The tester reproduced the attack to verify the vulnerability and then confirmed that the patch successfully mitigated the issue. No further exploitable weaknesses were found on the affected systems.”*
* **Impact:** State what the impact was or could have been. E.g.: *“The attacker accessed a limited part of an internal server and a small data file, but thanks to quick action, no sensitive data was exfiltrated. Business operations were not significantly affected beyond the isolated user PC.”*
* **Lessons Learned:** List improvements to prevent future incidents. For example:
  + *“Deploy advanced email filtering and conduct phishing awareness training for staff to reduce the chance of successful phishing.”*
  + *“Improve patch management: critical server patches should be applied within a week of release to avoid exploitation of known vulnerabilities.”*
  + *“Enhance network monitoring and intrusion detection to catch suspicious lateral movement faster.”*
  + *“Share the attack indicators (malware hash, IP, etc.) with the broader security community through our CERT to help defend others.”*

Write this report in a clear, non-technical style since it might be read by management. Use the Incident Report template provided or simply enter it into the KYPO platform’s report field.

**You should have:** A coherent incident report touching all key points. It doesn’t need to be long – a few paragraphs covering the above points is sufficient, as long as it’s precise and understandable.

**Tip:** Think of explaining to a non-IT manager what happened and how your team handled it. Avoid jargon like “RCE” or “APT” without explanation – instead say “critical vulnerability” or “state-sponsored hacker group” if needed. The goal is to convey that the team was effective and outline how to strengthen defences going forward.

**Step 31:** **Close the incident case in TheHive.** Now that reporting is done, formally close out the case:

* In TheHive, change the case status to **“Resolved” or “Closed.”** You might add a final note like “Incident resolved. Case closed on [date].”
* Ensure all tasks are marked completed and all relevant evidence (files, notes, reports) have been attached or stored as needed.
* If there’s an option to export the case or send a final notification, do so now.

**You should see:** The case marked as closed in TheHive (it may be moved to a closed cases list or simply marked with a closed status). This indicates the incident is officially wrapped up.

**Tip:** In a real scenario, closing a case might trigger a summary email to stakeholders or a meeting to debrief. Here, just make sure you’ve captured everything. It’s good practice to revisit whether all IOCs were shared and if any longer-term actions (like further monitoring of systems) are needed.

**Step 32:** **Complete any final quizzes or questions** on the platform. The exercise might end with a short quiz or reflection to ensure you grasp the IR process. For example, you might be asked: *“Which phase of the incident response involved notifying stakeholders?”* or *“What was the root cause of this incident?”* Use the knowledge you’ve gathered:

* (For the example question above, the answer would be along the lines of *“The Recovery/Lessons Learned phase involves stakeholder notification and reporting.”*) Provide the best answer based on your experience in this scenario.
* Double-check your timeline and report for these details if needed.

**You should have:** All questions answered and a completed scenario. Once you submit any quizzes or required information, the platform will consider the module finished.

**Congratulations!** You have successfully navigated a full incident response lifecycle, from initial detection to final reporting. You acted as a SOC analyst, incident responder, threat intel analyst, pen tester, and coordinator. This hands-on experience has given you insight into how each role contributes to managing cyber incidents.

Take a moment to reflect on the entire process:

* How early detection by the SOC can drastically reduce impact.
* The importance of quick containment and thorough forensic gathering by responders.
* The value of threat intelligence in understanding the adversary and strengthening your investigation.
* The role of testing and validation in ensuring the incident is truly resolved.
* The necessity of documentation and communication to learn and improve from each incident.

By following this implementation plan, the **KYPO Cyber Range scenario** for NG-SOC Module 8 will deliver a realistic, end-to-end incident response exercise. It integrates the necessary technical infrastructure (as referenced in the module diagrams: SIEM, TheHive, MISP, SOAR, etc.) with pedagogical structure (individual exercises with coordinated narrative). The final product will allow learners to not only practice technical skills but also the decision-making, documentation, and collaboration that are essential in real-world incident response. This ensures the training scenario is both technically rigorous and educationally effective, fully reflecting the intent of Module 8 and the NG-SOC project’s goals.

**References:**

* NG-SOC Deliverable D5.1 – *Module 8: Incident Response and Forensics* (skills and learning outcomes).
* KYPO “Junior Hacker” reference scenario (topology structure for attacker vs. target networks).
* TheHive Project documentation and blog – integration of TheHive, Cortex, and MISP for incident response.