

Réseaux et Télécoms

iut Nord Franche-Comté

Réalisé par :

Nicolas
RABERGEAU



NUMERICA

Pôle Numérique de Bourgogne-Franche-Comté

Rapport de Stage



Cr Louis Leprince-Ringuet, 25200 Montbéliard



Responsable Pédagogique - Boujemaa AOUBIZA



Responsable Professionnel - Vincent PAN



du 15/04/2024 au 07/06/2024

Sommaire

1. Remerciements.....	3
2. Introduction.....	4
3. Entreprise.....	5
3.1. Présentation d'entreprise.....	5
3.2. Mon rôle dans l'entreprise.....	6
3.3. Découverte d'entreprise.....	7
4. Cartographie.....	8
5. Projet du Firewall.....	9
5.1. Début du projet.....	9
5.2. Choix des Logiciels.....	9
5.3. Btrfs.....	10
5.4. Suricata.....	12
5.4.1. C'est quoi Suricata ?.....	12
5.4.2. L'installation et utilisation Suricata.....	12
5.5. Ntopng.....	14
5.5.1. C'est quoi Ntopng ?.....	14
5.5.2. L'installation et utilisation de Ntopng.....	14
5.6. Intégration Suricata avec Ntopng.....	16
5.7. Tests de fonctionnalité.....	17
5.7.1. Source routing.....	18
5.8. Configuration syslog-ng pour Suricata.....	19
5.9. Fin du Projet.....	20
5.9.1. Problème rencontrée Fin du Projet.....	20
5.9.2. Solution Fin du Projet.....	21
5.9.3. Conclusion Fin du Projet.....	21
6. Activités annexes.....	22
6.1. Conteneurisation.....	22
6.2. Mise en place réseau pour une événement.....	23
7. Conclusion.....	24
8. Annexes.....	25
9. Bibliographie.....	32



1. Remerciements

Je tiens à exprimer mes sincères remerciements à mon tuteur, M. Vincent PAN, directeur technique, pour son encadrement et son soutien tout au long de mon stage. Grâce à sa supervision et à ses conseils, j'ai pu réaliser mon travail dans de bonnes conditions et acquérir des compétences précieuses.

Merci également à M. Boujemaa AOUBIZA, Professeur de mathématiques du département R&T et mon tuteur académique, pour son aide et son soutien en cas de besoin.

Enfin, je souhaite exprimer ma gratitude envers l'entreprise SEM Numerica, pour m'avoir offert cette opportunité de stage enrichissante.



2. Introduction

Le but de ce rapport de stage est de décrire les projets qu'on a effectués pendant la période passée en stage chez l'entreprise Numerica. Ce stage s'est déroulé du 15 avril au 7 juin 2024. J'ai travaillé en équipe avec Mathis Guesdon qui était également en Stage de 2^{ème} année de R&T et Vincent Pan qui est le Directeur Technique de l'entreprise et aussi notre tuteur du stage. J'ai eu l'opportunité de mettre en pratique mes connaissances théoriques et de développer de nouvelles compétences professionnelles. Les étapes de mise en œuvre seront décrites en détail, mettant en lumière les défis et les solutions.



3. Entreprise

3.1. Présentation d'entreprise

Numerica est une société d'économie mixte qui opère dans le domaine de la transition numérique en Bourgogne-Franche-Comté. Elle propose une gamme de services et de programmes visant à accompagner les entreprises dans leur transformation digitale. Grâce à ses partenariats avec divers acteurs privés et institutionnels, Numerica offre des solutions telles que l'hébergement d'infrastructures informatiques, la formation professionnelle et le soutien à l'innovation numérique. Son objectif est de contribuer au développement économique régional en favorisant l'adoption des technologies numériques.



Les missions et services

Les missions

DÉVELOPPER les usages numériques

SENSIBILISER aux enjeux et opportunités offerts par le numérique

FAIRE ÉMERGER des projets collaboratifs et innovants

Comment ?

Programmes d'intérêt général



Services



Numerica DATA



Hébergement d'infrastructures informatiques

- Externalisation du système d'informations
- Location d'emplacements
- Implantation de point de présence opérateur (POP)

Internet Très Haut Débit

- Fournisseur d'accès Internet (Arcep 2010)
- Service aux entreprises et collectivités
- Spécialisé sur les réseaux d'initiative publique
- Service fibre optique et pont radio
- Très Haut Débit symétrique dédié et garanti

Étude et conseil

- Maîtrise d'oeuvre
- Maîtrise d'ouvrage
- Accompagnement métiers techniques



Un des trois services fournis est le « Numerica DATA » d'où je fais partie avec mon collègue Mathis Guesdon et le responsable de ce service, M. Vincent PAN.

Dans le service Numerica Data, le rôle principal est de garantir la disponibilité et la fiabilité des services Internet très haut débit ainsi que de superviser l'hébergement des infrastructures informatiques des clients. Cela implique également d'apporter un soutien technique aux clients, de contribuer à la gestion de projets liés à l'implémentation de nouvelles infrastructures, et de travailler en étroite collaboration avec les équipes techniques internes et externes. En résumé, le travail consiste à assurer la qualité et la sécurité des services fournis aux clients de Numerica Data.

3.3. Découverte d'entreprise

La première journée de stage, nous avons découvert les trois bâtiments. Il y a tout d'abord le bâtiment principal de Numerica, comprenant un rez-de-chaussée, un premier et un deuxième étage. Ensuite, le deuxième bâtiment (Bâtiment A) abrite Rubika, une école de design numérique. Enfin, le troisième bâtiment (Bâtiment B) est dédié aux bureaux libres et sert de studio. Notre tuteur de stage nous a également présentés au personnel et nous a attribué, à Mathis Guesdon et à moi-même, chacun un espace de travail dans son bureau, équipé de nouveaux ordinateurs portables.

Mon quotidien consistait à me réveiller et à suivre ma routine matinale, quittant mon logement du CROUS à 7h55 pour arriver au bureau à 8h (idéalement). Je prenais mon café du matin à l'entreprise et travaillais jusqu'à midi. Ensuite, j'ai marché jusqu'au Resto-U pour manger un plat complet pour seulement 1 euro. Après le repas, il était généralement 12h30 et nous reprenions le travail à 13h30. Pendant mon heure de pause, je retournais soit dans mon logement pour faire une petite sieste, soit je visitais l'association étudiante de R&T avec Mathis Guesdon (mon collègue de travail aussi le président de l'association) pour socialiser avec les étudiants de première année ou les alternants en deuxième année. Enfin, je quittais mon travail à 17h et rentrais à pied chez moi.

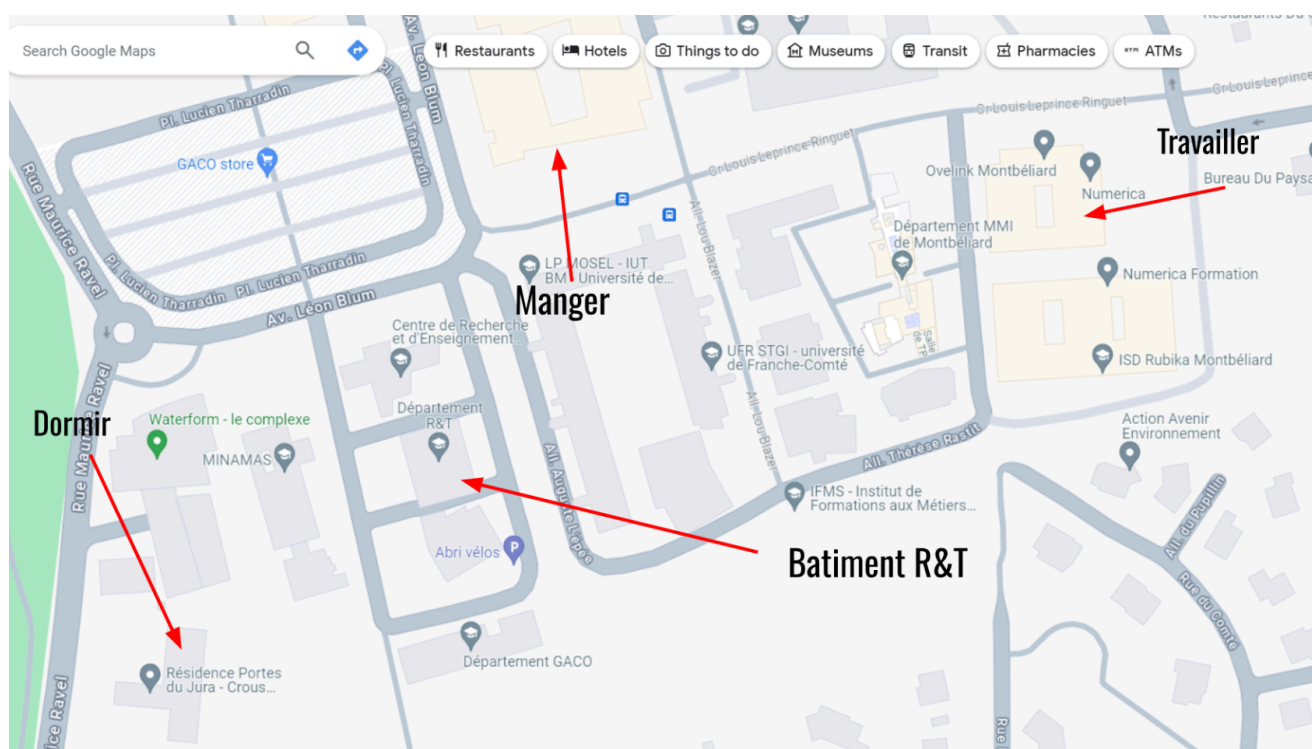


Image google maps de Numerica, Resto-U, Logement Crous et Bâtiment R&T

4. Cartographie

Notre première mission était de comprendre comment fonctionne le réseau de SEM Numerica. On nous a confié la tâche de refaire une carte du réseau central. On a commencé par regarder de près les routeurs Cisco, qui sont importants pour diriger le trafic dans le réseau. Vincent PAN, responsable technique, nous a expliqué comment tout marche lors d'une réunion. Ensuite, on a pu aller dans les salles où sont rangés les équipements pour voir comment ils sont connectés entre eux. Ça nous a aidés à comprendre comment tout est organisé.

Il est essentiel de comprendre que SEM Numerica n'est pas seulement une petite entreprise ordinaire. Elle est aussi un fournisseur d'accès. Cela signifie qu'en plus de ses activités habituelles, elle assure la connexion Internet pour d'autres entreprises. Cela peut rendre notre cartographie très grande, ce qui peut prendre toute la durée du stage à réaliser. C'est pour cela que nous nous concentrons uniquement sur le réseau interne.

On a passé environ deux semaines sur cette mission. Pendant ce temps-là, on a utilisé des logiciels comme Visio pour dessiner la carte du réseau, et Excel pour noter toutes les infos importantes sur les équipements.

Nous avons aussi commencé à utiliser le logiciel Zabbix pour surveiller activement le réseau en temps réel. Cela nous a donné la capacité de repérer rapidement les problèmes éventuels et d'avoir une meilleure compréhension du fonctionnement quotidien du réseau.

En résumé, cette première mission nous a vraiment plongés dans le monde compliqué du réseau de Numerica. Mais grâce à une approche méthodique, on a réussi à bien comprendre comment tout marche, ce qui va beaucoup nous aider pour la suite.

En raison de confidentialité, je ne peux pas mettre une image de cartographie dans ce rapport.



5. Projet du Firewall

5.1. Début du projet

Nous avons commencé un nouveau projet qui consiste à reconstruire complètement le pare-feu de SEM Numerica.

Vincent Pan, Le directeur technique nous a donné accès à un serveur dans une salle non utilisée dans la grande salle contenant plusieurs serveurs de tous types. Nous avons d'abord installé Debian 12 sur ce serveur pour avoir un système de base fiable. Ensuite, nous avons configuré un accès à distance pour pouvoir travailler dessus depuis nos bureaux, et à partir de là nous avons commencé notre projet.

Le serveur Debian que nous utilisons a deux interfaces réseau : une pour la DMZ (zone démilitarisée) et l'autre pour le WAN (réseau étendu). Pour l'interface WAN, Vincent nous a donné une adresse IP publique de SFR pour que le trafic puisse sortir vers internet. Pour la partie DMZ, nous configurons avec une adresse IP privée appartenant à la plage d'adresses de la DMZ. Cette configuration permet d'isoler les serveurs exposés à Internet dans la DMZ tout en protégeant le réseau interne.

5.2. Choix des Logiciels

Nous avons la liberté de choisir les logiciels que nous voulions utiliser pour ce projet. Après avoir discuté et rédigé un cahier des charges, nous avons décidé d'utiliser ces logiciels open source suivant :

- **Nftables** : Pare-feu
- **Grafana** : Outil de visualisation de données.
- **Loki** : Système de journalisation
- **Suricata** : Système de détection d'intrusion réseau.
- **Btrfs** : Système de snapshot
- **Ntopng** : Outil de surveillance du réseau.

Nous avons opté pour Nftables comme pare-feu, car il offre une gestion efficace et est plus moderne que iptables, le système précédent utilisé. Pour surveiller le pare-feu, nous avons opté pour Grafana, qui est excellent pour visualiser les données avec une belle interface. Loki nous permet de consulter les journaux, ce qui est essentiel pour analyser les événements.

Pour la sécurité, nous avons choisi Suricata comme système de détection d'intrusion (IDS) pour repérer toute activité suspecte sur le réseau. Pour sauvegarder notre configuration et pouvoir revenir en arrière en cas de problème, nous utilisons Btrfs, qui permet de créer des "snapshots" ou instantanés du système. Enfin, pour la surveillance et le monitoring du réseau, nous utilisons Ntopng, qui nous fournit des informations détaillées sur le trafic réseau.



5.3. Btrfs

Avec Mathis, nous avons commencé par configurer Btrfs en priorité. De cette façon, si nous rencontrons un problème ou faisons une erreur en configurant les autres logiciels, nous pourrions facilement revenir à une configuration précédente qui fonctionne.

```
manager@Plouf: ~  
GNU nano 7.2 .bin/snapshot  
btrfs subvolume snapshot / /snapshots/@root-$(date +%s)  
grub-mkconfig -o /boot/grub/grub.cfg
```

Le script "snapshot" est utilisé pour prendre un instantané d'un fichier. L'instantané est un fichier qui contient une copie exacte du fichier d'origine à un moment donné. Cela peut être utile pour sauvegarder des configurations et de le restaurer en cas de problème.

Le script utilise la commande `btrfs subvolume snapshot` pour créer l'instantané. Cette commande prend les arguments suivants :

- **source**: Le chemin d'accès au répertoire que tu veux sauvegarder .
- **destination**: Le chemin d'accès au répertoire où tu veux le stocker.

Dans l'image, la commande suivante est utilisée pour prendre un instantané du répertoire racine / et l'enregistrer dans le répertoire `/snapshots/@root-$(date +%s)` :

```
btrfs subvolume snapshot / /snapshots/@root-$(date +%s)
```

La variable `date +%s` génère un horodatage sous forme de nombre de secondes depuis l'époque Unix. Cela garantit que chaque instantané a un nom unique.

Une fois que l'instantané est créé, le script utilise la commande `grub-mkconfig` pour mettre à jour le fichier de configuration GRUB. Cela garantit que le système d'exploitation peut démarrer à partir de l'instantané depuis le bootloader GRUB.

Voici quelques exemples d'utilisation du script "snapshot" :

- Pour prendre un instantané du fichier `/etc/passwd` et l'enregistrer dans le fichier `/snapshots/passwd-$(date +%s)` :

```
snapshot /etc/passwd /snapshots/passwd-$(date +%s)
```

- Pour prendre un instantané du répertoire racine et l'enregistrer dans le répertoire `/snapshots`, tapé tout simplement `snapshot` dans le terminal en tant que `root`.

```

manager@Plouf:~$ su -
Password:
root@Plouf:~# snapshot
Create a snapshot of '/' in '/snapshots/@root-1714380925'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.1.0-20-amd64
Found initrd image: /boot/initrd.img-6.1.0-20-amd64
Found linux image: /boot/vmlinuz-6.1.0-18-amd64
Found initrd image: /boot/initrd.img-6.1.0-18-amd64
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Detecting snapshots ...
Found snapshot: 2024-04-29 10:55:25 | @root-1714380925 | N/A | N/A |
Found snapshot: 2024-04-29 10:53:16 | @root-1714380796 | N/A | N/A |
Found snapshot: 2024-04-29 10:51:06 | @root-1714380666 | N/A | N/A |
Found snapshot: 2024-04-29 10:17:48 | @root-1714378668 | N/A | N/A |
Found snapshot: 2024-04-29 10:17:37 | @root-1714378657 | N/A | N/A |
Found 5 snapshot(s)
Unmount /tmp/grub-btrfs.IaerPl80dX .. Success
done
root@Plouf:~# |

```

Capture d'écran du terminal de firewall en exécutant le script snapshot en tant que root

Pour automatiser certaines tâches, nous avons créé quelques scripts Bash. Pour cela, nous avons d'abord créé un répertoire caché nommé `.bin` dans le répertoire home de root, puis nous avons ajouté différents scripts Bash à l'intérieur.

```

[root@Plouf]-(~)
#ls .bin/
linkdown linkup snapshot

```

Nous avons créé deux scripts Bash spécifiques pour gérer l'interface WAN.

Le premier script, appelé `linkdown`, désactive l'interface WAN lorsque vous tapez "linkdown" dans le terminal en tant que root. Cela coupe l'accès à Internet pour cette interface, ce qui empêche notre adresse publique d'être exposée lorsqu'elle n'est pas nécessaire.

```

manager@Plouf: ~
GNU nano 7.2 .bin/linkdown
#!/usr/bin/sh
ip l set down enp2s0f1

```

Le deuxième script, appelé `linkup`, réactive l'interface WAN et ajoute la route par défaut. En tapant "linkup" dans le terminal en tant que root, le script configure l'interface pour qu'elle soit opérationnelle et permet au trafic de sortir vers Internet.

```

manager@Plouf: ~
GNU nano 7.2 .bin/linkup
#!/usr/bin/sh
ip l set up enp2s0f1
ip r add default via IP Publique

```



Une fois que Mathis et moi avons fini de configurer les snapshots en priorité, nous avons pu travailler en tranquillité sans avoir peur que plus rien ne marche et qu'il faudrait tout recommencer. Pour être plus efficaces, j'ai décidé de diviser les tâches avec Mathis Guesdon. J'ai pris en charge Suricata, l'IDS, et Mathis gérer Nftables, le pare-feu.

5.4. Suricata

5.4.1. C'est quoi Suricata ?

Suricata est un IDS et pour comprendre le suivant il est important de savoir c'est quoi un IDS:

Un système de détection d'intrusion (IDS) est un outil de sécurité des réseaux qui surveille le trafic réseau et les appareils pour détecter les activités malveillantes connues, les activités suspectes ou les violations des politiques de sécurité.

5.4.2. *L'installation et utilisation Suricata*

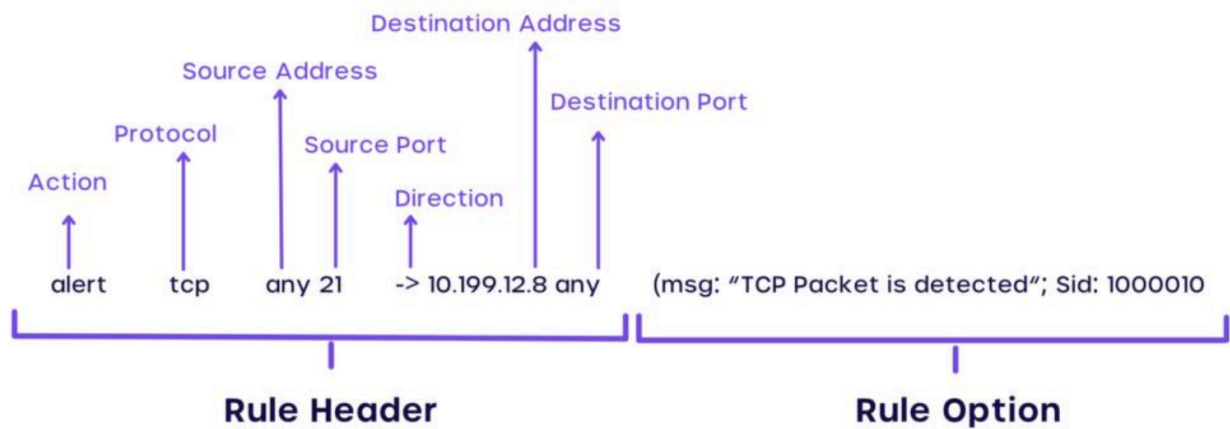
En ssh (bureau a distance) j'ai fait l'installation basique de suricata grâce au binary packages debian sur la même machine que le firewall.

```
apt-get install suricata
```

Pour la configuration, j'ai fait en sorte que l'IDS détecte le trafic sur l'interface réseau publique sortante WAN. Pour tester le fonctionnement, j'ai créé une alerte en construisant une règle manuelle qui se déclenche lorsqu'on fait une simple requête ping (ICMP) sur n'importe quel port et avec n'importe quelles adresses et ports source et destination (any).

```
alert icmp any any -> any any (msg:"ALERTE PING MARCHE!"; sid:1; )
```





Voici un exemple pour comprendre la syntaxe de Suricata pour ensuite fabriquer les règles manuellement

1. alert: Indique que cette règle doit déclencher une alerte lorsqu'elle est satisfaite, signalant ainsi un événement suspect ou potentiellement malveillant.
2. tcp: Spécifie le protocole, indiquant que la règle s'applique au trafic utilisant ce protocole.
3. any: Indique que le trafic **sortant** peut venir de n'importe quelle adresse IP (troisième valeur) ou n'importe quel port du trafic **entrant** (cinquième valeur).
4. 21: Le port source qui est utilisé par le protocole FTP cela signifie que la règle surveille le trafic FTP **sortant**.
5. ->: Utilisé pour spécifier la direction du trafic, indiquant l'expéditeur et le destinataire du paquet.
6. msg: Définit le message associé à l'alerte qui sera générée lorsque la règle est satisfaite. Il fournit une description de l'événement détecté.
7. sid: Chaque règle est associée à un identifiant de signature unique qui permet de référencer spécifiquement cette règle. Cela facilite l'identification et la gestion des alertes générées par le système de détection d'intrusion.

Une fois que j'ai confirmé que les alertes de l'IDS fonctionnaient, j'ai pris un ruleset préconstruit avec au moins 40 000 règles différentes et l'appliquer à mon IDS. C'est à partir d'ici que je rends compte de tous les bots et les requêtes qui me viennent de l'internet.

Selon le rapport Imperva Bad Bot 2024 de Thales, un leader de la cybersécurité, près de la moitié (49,6%) du trafic Internet mondial a été généré par des bots, ce qui représente une augmentation de 2% par rapport à 2022.

L'installation et la configuration de Suricata étaient assez simples pour moi à mettre en place, car j'ai fait un projet similaire à la SAE 401 dans mon parcours choisi en cybersécurité.



5.5. Ntopng

Mon prochain objectif est maintenant de mettre en place Ntopng pour la surveillance du réseau, le faire fonctionner sur le réseau et éventuellement l'intégrer avec mon IDS pour que les alertes puissent être affichées et bien documentées sur une interface web de ntopng.

Au début, nous avons testé d'autres logiciels de surveillance réseau avant de nous mettre d'accord sur ntopng. Un autre que nous avons testé s'appelle Netify. En gros, Netify est similaire à ntopng mais 10 fois plus puissant et plus simple à installer. Cependant, le problème était que son coût était de 25 euros par mois. Nous avons quand même pu le tester pendant un court laps de temps car nous disposions d'un essai gratuit de 10 minutes, durant lequel nous avons pu apprécier la puissance du logiciel.

Netify et Ntopng utilisent ce qu'on appelle le Deep Packet Inspection (DPI), ce qui permet d'analyser en profondeur le contenu des paquets de données transitant sur le réseau. Par exemple, il est capable de déterminer si vous vous promenez sur TikTok, si vous téléchargez un film piraté ou même si vous utilisez un VPN.

5.5.1. C'est quoi Ntopng ?

Ntop ntopng (Network TOP) est un outil Open Source de supervision **réseau** distribué sous GPL 3. C'est une application qui produit des informations sur le trafic réseau en temps réel (comme pourrait le faire la commande **top** avec les processus).

Il capture et analyse les trames d'une interface donnée en utilisant **Libpcap**.

Il permet d'observer une majeure partie des caractéristiques du trafic réseau en entrée et sortie (couche **2** et **3** du modèle **OSI**) à travers une interface web.

5.5.2. L'installation et utilisation de Ntopng

L'installation était assez facile comme suricata, il suffit juste d'installer le paquet deb de ntopng

```
apt install ntopng
```

Pour arrêter le daemon :

```
service ntopng stop
```

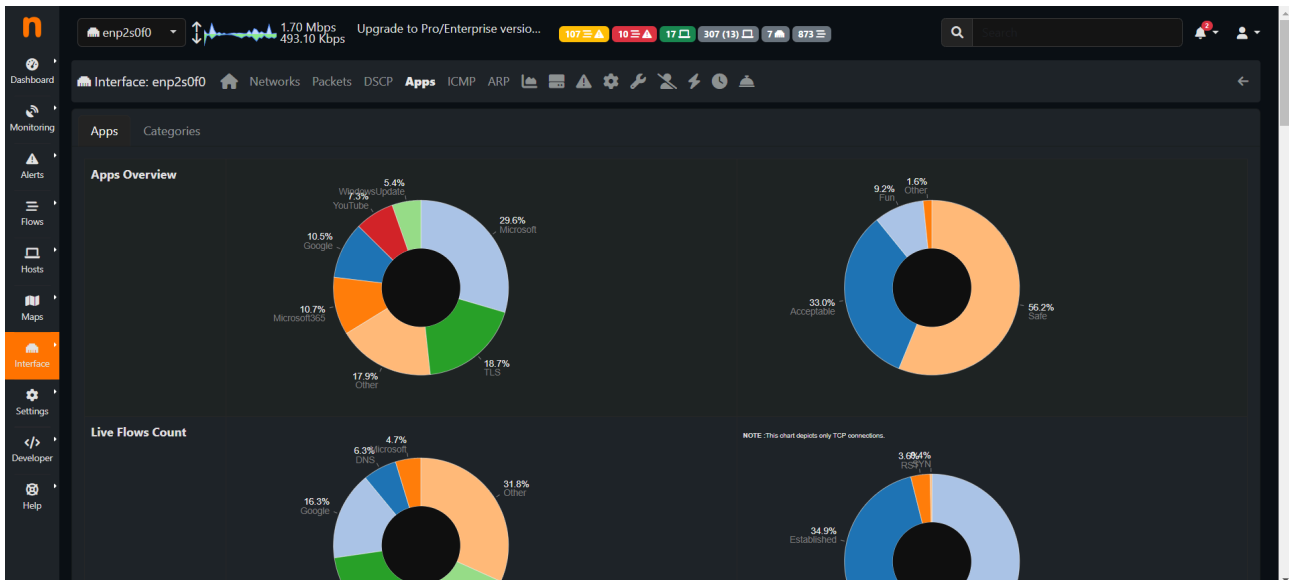
Puis pour démarrer le daemon :

```
sudo service ntopng start
```

L'accès à Ntop se fait dans un navigateur web par défaut sur le port 3000:

<http://127.0.0.1:3000>





Capture d'écran du interface web de ntopng

On a rencontré un petit problème lorsque j'ai essayé d'accéder à l'interface web ntopng. Le conflit était entre le firewall nftables de Mathis a mise en fonctionnement et mon ntopng.

Ntopng ne pouvait pas monter le serveur web car il était bloqué par le firewall. Au début, je ne savais pas pourquoi le serveur web ne fonctionnait pas, du coup j'ai passé beaucoup de temps à le trouver. J'ai demandé à Mathis d'essayer d'éteindre son firewall pour voir si c'était ça et effectivement le problème venait de son firewall.

Pour fixer cela, Mathis a ajouté un règles dans son firewall qui accepte tout trafic venant de la machine lui même donc le localhost (127.0.0.1).

```
chain firewall_in {
    type filter hook input priority filter; policy drop;
    comment "Règle de gestion d'accès au firewall"

    ip saddr $IP_TECH accept
    ip saddr $IP_SRV_LOG udp dport 161 accept
    ip saddr 127.0.0.0/24 accept
}
```

Capture d'écran des règles nftables pour résoudre ce problème.

5.6. Intégration Suricata avec Ntopng

Une fois que ntopng était fonctionnel, j'ai découvert qu'il était possible d'intégrer Suricata et ntopng. J'ai proposé l'idée à mon tuteur et il était d'accord pour que je mette cela en place.

Ntopng s'intègre avec Suricata pour importer à la fois les métadonnées des flux (Suricata agissant comme un capteur) et les alertes. L'ingestion des alertes permet à ntopng de compléter son moteur d'analyse de trafic intégré avec les capacités flexibles de détection des menaces basées sur des signatures fournies par Suricata.

ntopng implémente l'ingestion des flux et des alertes de Suricata en utilisant le format JSON Eve via syslog.

Pour configurer Suricata afin d'exporter les métadonnées des flux vers ntopng, On a dû configurer Suricata pour qui utilise syslog comme type de fichier EVE. Cela se fait depuis le fichier de configuration suricata.yaml (situé sous /etc/suricata/suricata.yaml) :

```
- eve-log:
  enabled: yes
  filetype: syslog #regular|syslog|unix_dgram|unix_stream|redis
  filename: eve.json
```

capture d'écran du fichier de configuration suricata fais pour exporter en syslog en format JSON

Enfin, je dois configurer ntopng pour ingérer les messages syslog en ajoutant l'interface Syslog syslog://<ip>:<port> au fichier de configuration pour écouter les connexions à l'<ip> et au <port> spécifiés :

```
manager@Plouf. ~
GNU nano 7.2 /etc/ntopng/ntopng.conf
#
-i=enp2s0f0
-i=syslog://*:9999
```

capture d'écran du fichier de configuration ntopng fais pour écouter les données syslog au port 9999

J'ai rencontré un problème lors des envoyées des données de suricata vers ntopng. Le problème était que lorsque j'ai activé l'option pour mettre les données en syslog de suricata, la console systemd journal envoyait les données au ntopng qui est normale mais aussi balançait tous les messages dans le console du machine. Cela rendrait le terminal inutilisable.

Pour résoudre ce problème j'ai modifié un option dans la fichier de configuration de journald qui permet de transférer les messages vers syslog et de ne pas les transférer vers le console «Wall » en mettant l'option:

```
ForwardToSyslog=Yes
ForwardToWall=No
```




```
GNU nano 7.2 /etc/systemd/journald.conf
# Use 'systemd-analyze cat-config systemd/journald.conf' to display the full config.
#
# See journald.conf(5) for details.

[Journal]
Storage=volatile
#Compress=yes
#Seal=yes
#SplitMode=uid
#SyncIntervalSec=5m
#RateLimitIntervalSec=30s
#RateLimitBurst=10000
#SystemMaxUse=
#SystemKeepFree=
#SystemMaxFileSize=100
#SystemMaxFiles=100
#RuntimeMaxUse=
#RuntimeKeepFree=
RuntimeMaxFileSize=10
#RuntimeMaxFiles=100
#MaxRetentionSec=
#MaxFileSec=1month
ForwardToSyslog=yes
#ForwardToKMsg=no
#ForwardToConsole=no
ForwardToWall=no
#TTYPath=/dev/console
```

Capture d'écran fichier configuration systemd-journald

```
<128>May 17 14:06:10 Plouf suricata[717449]:
{"timestamp":"2024-05-17T14:06:10.603706+0200","flow_id":1563551749749057,
"event_type":"flow","src_ip":"192.168.115.28","src_port":35274,
"dest_ip":"142.251.221.10","dest_port":443,"proto":"TCP","flow":
{"pkts_toserver":172,"pkts_toclient":150,"bytes_toserver":47612,
"bytes_toclient":65664,"start":"2024-05-17T13:42:25.833857+0200",
"end":"2024-05-17T14:01:48.870282+0200","age":1163,"state":"new","reason":"timeout",
"alerted":false},"tcp":{"tcp_flags":"00","tcp_flags_ts":"00","tcp_flags_tc":"00"}}
```

Exemple de message syslog transmit via Syslog-ng

5.7. Tests de fonctionnalité

Pour vérifier que notre nouveau pare-feu fonctionne correctement avant de l'utiliser en production, Vincent Pan a créé un environnement de test qui imite les conditions réelles. Cela nous permet de nous assurer que toutes les configurations et les règles de sécurité sont bien mises en place. Vincent nous a montré comment cet environnement de test fonctionne.

Prenons l'exemple où nous voulons tester un ping vers google.fr. Lorsque nous envoyons un ping depuis notre ordinateur, le trafic suit plusieurs étapes. D'abord, il passe par le routeur principal de notre réseau. Ce routeur redirige le trafic en fonction des règles ACL (Access Control List) et des VLAN (Virtual Local Area Network). Ensuite, le trafic passe par un proxy, qui agit comme un intermédiaire et gère les requêtes entre notre réseau interne et l'extérieur. Après avoir traversé le proxy, le trafic atteint notre pare-feu en développement grâce à une technique appelée "source routing". Finalement, le trafic sort par l'interface WAN pour accéder à Internet.



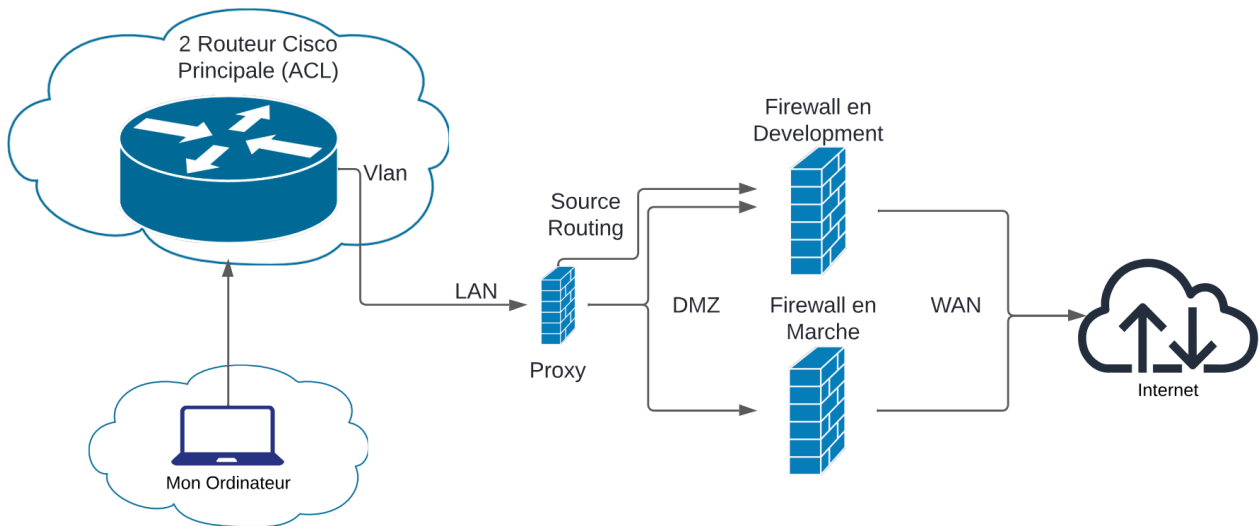


Schéma d'infrastructure réseau que j'ai fais avec lucidchart pour mieu comprendre

5.7.1. Source routing

La méthode de "source routing" utilisée pour diriger le trafic à partir du proxy vers notre pare-feu en développement implique une configuration spécifique. Voici comment nous l'avons mise en place :

Ajout d'une nouvelle table de routage :

- Dans le fichier `/etc/iproute2/rt_tables`, nous ajoutons une nouvelle table de routage appelée FWSTAGE. Cette table est identifiée par le numéro 10.

```
10 FWSTAGE
```

Ajout d'une route par défaut dans la nouvelle table :

- Nous ajoutons une route par défaut dans cette table pour diriger tout le trafic via l'adresse IP du pare-feu en production actuel.

```
ip route add default via [IP du Firewall en Marche] table FWSTAGE
```

Création d'une règle de routage basée sur l'adresse source :

- Ensuite, nous créons une règle qui utilise l'adresse IP de notre ordinateur pour cette table de routage. Cela signifie que tout le trafic provenant de notre ordinateur est traité selon les routes définies dans la table FWSTAGE.

```
ip rule add from [IP de mon ordinateur] table FWSTAGE
```



En utilisant cette configuration, nous pouvons diriger précisément le trafic de notre ordinateur à travers le proxy, vers notre pare-feu en développement, puis vers le WAN pour accéder à Internet. Cette méthode nous permet de tester le pare-feu comme s'il était en production, sans perturber le trafic réel. Grâce à cet environnement de test, nous pouvons identifier et résoudre d'éventuels problèmes avant le déploiement final du pare-feu.

5.8. Configuration syslog-ng pour Suricata

```
filter f_suricata { program(suricata); };  
destination d_ntopng { tcp("127.0.0.1" port(9999) log_fifo_size(1000)); };  
destination d_loki { syslog("192.168.114.251" transport("tcp") port(1514)); };  
log { source(s_src); filter(f_suricata); destination(d_ntopng); };  
log { source(s_src); filter(f_suricata); destination(d_loki);};
```

Le fichier de configuration présent dans l'image détaille la mise en place de la configuration syslog-ng pour le système de détection d'intrusion Suricata et l'outil de surveillance ntopng.

La section destination définit les destinations vers lesquelles les données sont envoyées. Deux destinations sont définies, `d_ntopng` et `d_loki`.

`d_ntopng`: Envoie les données filtrées par `f_suricata` au port 9999 sur localhost (127.0.0.1), utilisé par ntopng, un outil d'analyse du trafic réseau.

`d_loki`: Envoie les données filtrées par `f_suricata` au port 1514 sur l'hôte 192.168.114.251, utilisé par Loki, un système de collecte de journaux open source.

La section log définit les journaux utilisés pour envoyer les données. Deux journaux sont définis, `log1` et `log2`, chacun envoyant les données d'une source spécifique à un filtre et une destination spécifiques.

Lorsque Suricata détecte une activité suspecte sur le réseau, il génère un journal envoyé à la source `s_src`. Cette source envoie ensuite le journal au filtre `f_suricata` qui utilise Suricata pour filtrer le journal. Si le journal correspond à un modèle d'activité suspecte, il est envoyé aux destinations `d_ntopng` et `d_loki`. Ntopng et Loki peuvent alors analyser le journal pour déterminer la nature de l'activité suspecte.

Cette configuration syslog-ng illustre comment filtrer et envoyer des données de journalisation avec Suricata. En utilisant des filtres et des destinations, syslog-ng peut centraliser et analyser les données de journalisation provenant de diverses sources.



5.9. Fin du Projet

Une fois que tout était prêt, il a remplacé le pare-feu en développement par le pare-feu en production. Pour nous préparer, nous avons fait en sorte que les adresses IP dans les configurations du pare-feu de développement correspondent à celles du pare-feu en production, et vice versa. Nous avons également préparé des commandes très longues sur chaque pare-feu, qui permettaient de changer les adresses IP et les passerelles par défaut. Au décompte de trois, nous avons entré les deux commandes en même temps, et l'échange a été effectué.

Ensuite, nous avons réinstallé les mêmes configurations de Nftables sur le nouveau pare-feu en développement, dans le but de transférer notre travail sur le matériel original de l'ancien pare-feu.

Avec mon équipe et mon tuteur, nous avons conclu qu'il n'était pas nécessaire d'ajouter l'IDS et les outils de surveillance pour l'instant, l'essentiel étant que Nftables fonctionne correctement pour que le personnel et les clients de Numerica puissent avoir accès à Internet.

5.9.1. Problème rencontrée Fin du Projet

Nous avons également rencontré quelques problèmes qui pouvaient être extrêmement critiques, mais heureusement pour nous, ce n'était pas le cas. Nous avons eu un problème avec le disque qui n'était pas détecté par Debian 12 lors de l'installation sur le nouveau matériel du pare-feu. Notre tuteur a conclu que le matériel était trop ancien et que Debian 12 était trop récent, car auparavant le pare-feu était installé sur Debian 6 et utilisait Iptables, une ancienne génération de Nftables.

Avant de prendre la décision drastique de revenir à Debian 6, nous avons effectué des recherches sur internet pour trouver une solution. En fouillant dans différents forums et sites d'assistance, nous sommes tombés sur un forum datant de 10 ans où quelqu'un rencontrait exactement le même problème que nous. Ce fil de discussion détaille les difficultés liées à la détection des disques durs sur des systèmes avec des versions plus récentes de Debian, en particulier sur du matériel plus ancien.



5.9.2. Solution Fin du Projet

La solution proposée dans le forum était d'ajouter une option spécifique au démarrage via GRUB, le chargeur de démarrage utilisé par Debian. En modifiant les options de démarrage, il est possible d'activer certaines fonctionnalités ou de contourner des problèmes de compatibilité matérielle.

La commande suggérée pour résoudre notre problème était la suivante :

```
hpsa.hpsa_allow_any=1
```

Cette commande force le pilote hpsa (HP Smart Array) à permettre la reconnaissance de tous les périphériques, même ceux qui ne sont pas certifiés. Pour appliquer cette solution, nous avons dû modifier les paramètres de démarrage de GRUB. Voici les étapes détaillées que nous avons suivies :

- Démarrer l'ordinateur et accéder au menu GRUB : Juste après le démarrage, nous avons appuyé sur une touche (souvent Esc ou Shift) pour entrer dans le menu de GRUB.
- Modifier les options de démarrage : Dans le menu GRUB, nous avons sélectionné l'entrée correspondant à Debian 12 et appuyé sur la touche e pour éditer les options de démarrage.
- Ajouter l'option : Nous avons ajouté `hpsa.hpsa_allow_any=1` à la fin de la ligne qui commence par linux et décrit le noyau à charger et ses paramètres.
- Démarrer avec les nouvelles options : Nous avons appuyé sur Ctrl + X ou F10 pour démarrer Debian 12 avec les nouvelles options.

Grâce à cette modification, Debian 12 a pu détecter le disque dur, et nous avons pu continuer l'installation et la configuration du pare-feu sans avoir à revenir à une version obsolète du système d'exploitation.

5.9.3. Conclusion Fin du Projet

En conclusion, ce projet nous a permis de surmonter des défis techniques importants et d'apprendre beaucoup en matière de résolution de problèmes et d'administration système. Malgré les obstacles, nous avons réussi à mettre en place un pare-feu qui fonctionne bien et qui est sécurisé, assurant ainsi une transition sans problème pour le personnel et les clients de Numerica. Ce succès montre notre capacité à travailler ensemble, à persévérer et à utiliser les ressources disponibles pour trouver des solutions adaptées. Nous sommes fiers de ce que nous avons accompli et confiants dans la solidité et la fiabilité de la nouvelle Firewall.



6. Activités annexes

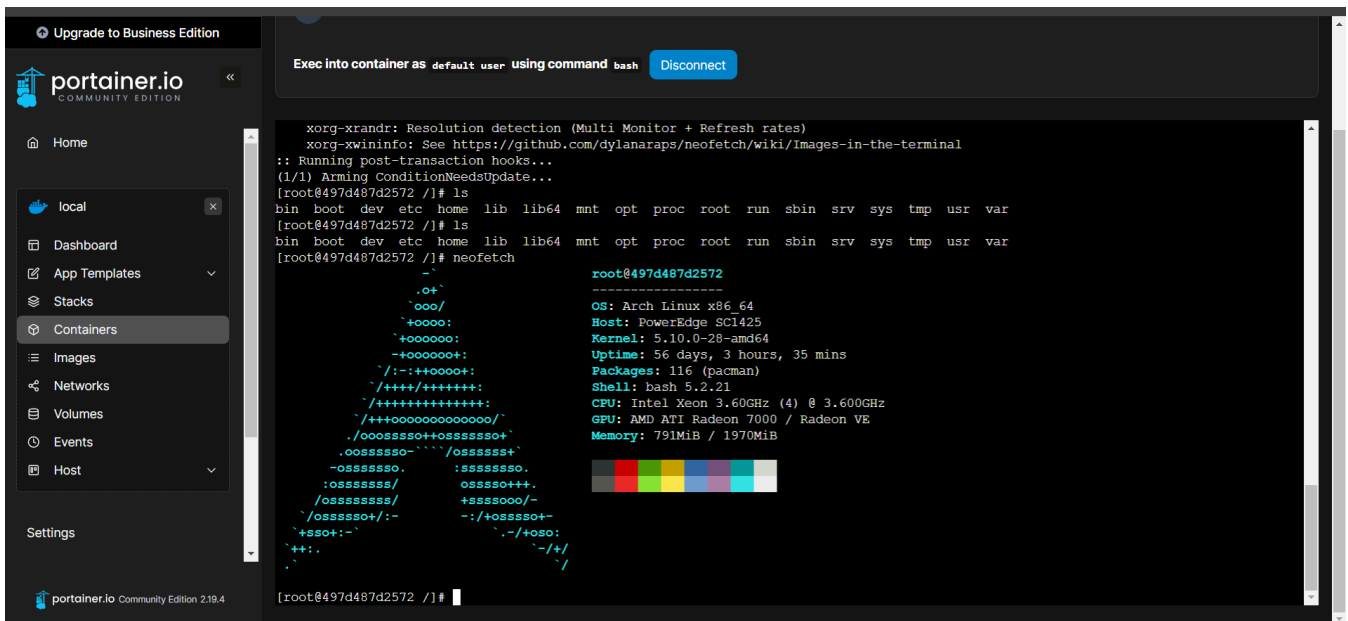
Pendant notre projet, nous avons été amenés à effectuer diverses activités typiques d'un technicien informatique au sein d'une entreprise. Cela inclut le guidage des opérateurs vers la salle serveur afin qu'ils puissent avoir accès, la mise en place des ordinateurs pour les services et les formations de Numerica, la résolution de problèmes liés à Internet ou bien la résolution de problèmes d'imprimantes.

6.1. Conteneurisation

Mathis et moi souhaitons explorer et expérimenter Kubernetes et Docker. Pour cela, M. Vincent PAN nous a accordé un accès à un serveur de conteneurisation, Portainer.

Portainer permet de gérer efficacement de nombreux aspects de Docker : conteneurs, images, volumes, réseaux, utilisateurs, etc. Il offre également la possibilité de contrôler à distance un autre serveur Docker via un agent, facilitant le déploiement d'applications dans des conteneurs en quelques clics seulement.

Pendant mes expérimentations, j'ai eu l'occasion d'installer Arch Linux, réputée pour être une distribution Linux très complexe à installer.



```
Exec into container as default user using command bash Disconnect

xorg-xrandr: Resolution detection (Multi Monitor + Refresh rates)
xorg-xwininfo: See https://github.com/dylannaraps/neofetch/wiki/Images-in-the-terminal
:: Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...
[root@497d487d2572 /]# ls
bin boot dev etc home lib lib64 mnt opt proc root run sbin srv sys tmp usr var
[root@497d487d2572 /]# ls
bin boot dev etc home lib lib64 mnt opt proc root run sbin srv sys tmp usr var
[root@497d487d2572 /]# neofetch

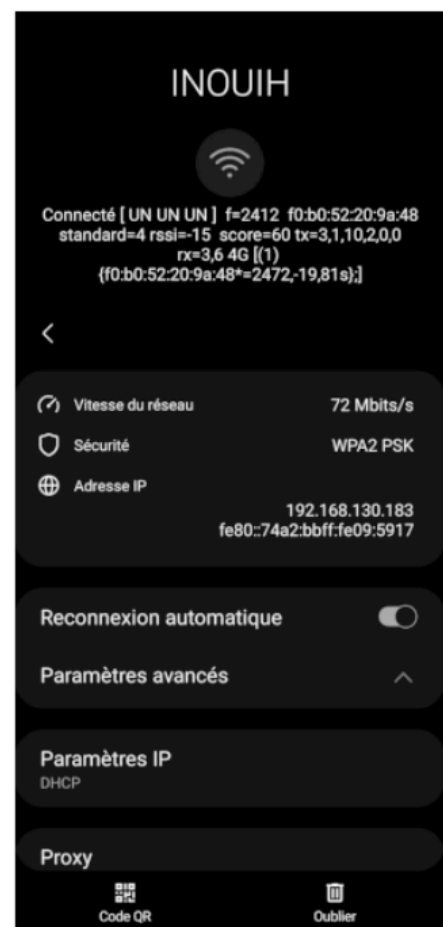
root@497d487d2572
-----
OS: Arch Linux x86_64
Host: PowerEdge_SCI425
Kernel: 5.10.0-28-amd64
Uptime: 56 days, 3 hours, 35 mins
Packages: 116 (pacman)
Shell: bash 5.2.21
CPU: Intel Xeon 3.60GHz (4) @ 3.600GHz
GPU: AMD ATI Radeon 7000 / Radeon VE
Memory: 791MiB / 1970MiB
```

Capture d'écran de portainer connecté sur la console de Arch Linux installé

6.2. Mise en place réseau pour une événement

Un exemple d'activité annexe était la mise en place du Wi-Fi pour le festival INOUIH. Pour ce faire, nous avons dû configurer un point d'accès Wi-Fi 2.4GHz et tester la connectivité depuis tous les ajouts de campus jusqu'à l'UTBM. Je mentionne cette activité car elle m'a rappelé les souvenirs de mes premières années de BUT R&T, plus particulièrement le projet de SAE13 avec M. Vanstraceele, où nous avons dû déterminer les types de mesures et les types de signaux nécessaires pour un point d'accès Wi-Fi Linksys.

En parallèle, on a également participé au festival INOUIH, qui s'est tenu le vendredi 31/05/2024. C'était un événement très enrichissant avec de nombreuses conférences et ateliers sur les nouvelles technologies. Les activités se sont déroulées principalement autour des bâtiments de Numerica.



7. Conclusion

Mon stage chez SEM Numerica a été une expérience très enrichissante à la fois professionnellement et personnellement.

Sur le plan professionnel, ce stage m'a permis d'appliquer et d'approfondir mes connaissances théoriques en informatique et en cybersécurité. J'ai notamment eu l'opportunité de travailler sur des projets concrets comme la refonte du pare-feu de l'entreprise, où j'ai utilisé des outils modernes tels que Nftables, Suricata et Ntopng. Ces expériences m'ont permis de mieux comprendre la gestion des infrastructures réseau et les systèmes de sécurité.

Sur le plan personnel, ce stage m'a apporté beaucoup en termes de relations humaines et de travail en équipe. Travailler aux côtés de Mathis Guesdon et sous la supervision de Vincent Pan m'a permis de développer des compétences en communication et en collaboration. Nous avons dû nous répartir les tâches, résoudre des problèmes ensemble et nous soutenir mutuellement dans les moments difficiles. Cette expérience m'a montré l'importance de la coopération et de la solidarité dans un environnement professionnel.

Le lien entre ce stage et ma formation en Réseaux et Télécommunications (R&T) est évident. Les compétences que j'ai développées au cours de ma formation, notamment en matière de sécurité réseau et de gestion des systèmes, ont été directement appliquées pendant ce stage. Cela m'a permis de consolider mes acquis et de les mettre en pratique dans un cadre réel, ce qui est essentiel pour une bonne compréhension de la théorie.

Enfin, ce stage a été une étape importante dans mon parcours, avec ses hauts et ses bas. J'ai grandi en tant que professionnel et en tant qu'individu, et je suis motivé à continuer à développer mes compétences et à relever de nouveaux défis dans ce domaine, en tirant des leçons de cette expérience pour m'améliorer à l'avenir.



8. Annexes

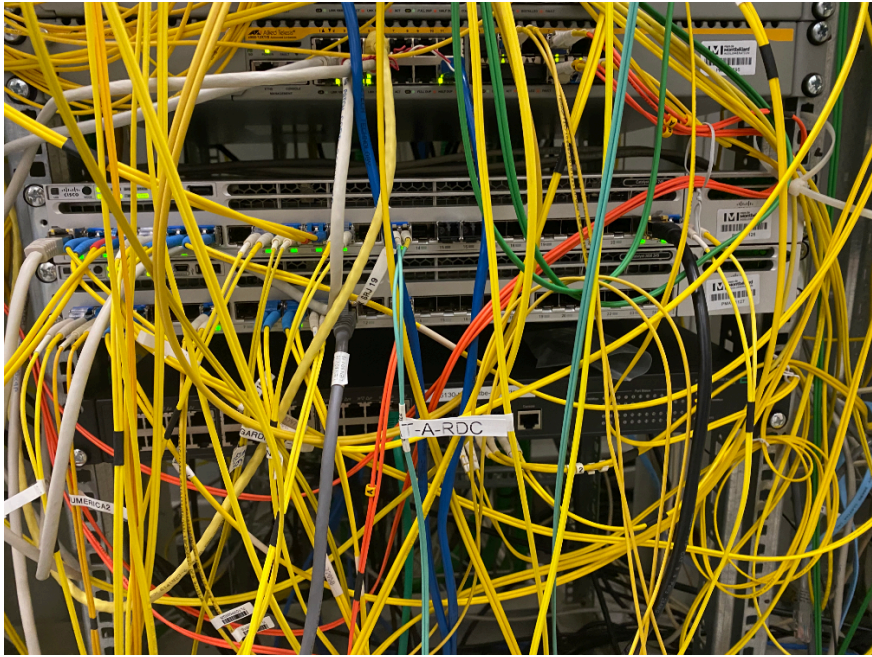
Salima INEZARENE
Présidente Directrice Générale

Frédéric MONNIER
Directriceur Général Adjoint



Organigramme de l'entreprise Numerica





Images des 2 routeur principaux lors du création du Cartographie

The screenshot displays the SURICATA interface with a list of log entries. The interface includes a search bar, a 'Query Inspector' section, and a right-hand sidebar with various configuration options like 'Panel options', 'Repeat options', and 'Logs'.

Logs reçus par Loki

Serial	Application	Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info	
	TLS.AdultCon...	DPI	TCP	192.168.115.253 :57129	ss.phncdn.com	00:16 sec	Server	0 bps	15.24 KB	ss.phncdn.com
	TLS.AdultCon...	DPI	TCP	192.168.115.253 :57079	pornhub.com	00:03 sec	Server	0 bps	155.05 KB	pornhub.com
	TLS.AdultCon...	DPI	TCP	192.168.115.253 :57108	cdn1-smallimg.phncdn.com	00:01 sec	Server	0 bps	9.7 KB	cdn1-smallimg.phncdn.com

Trafic capturé par ntopng



```

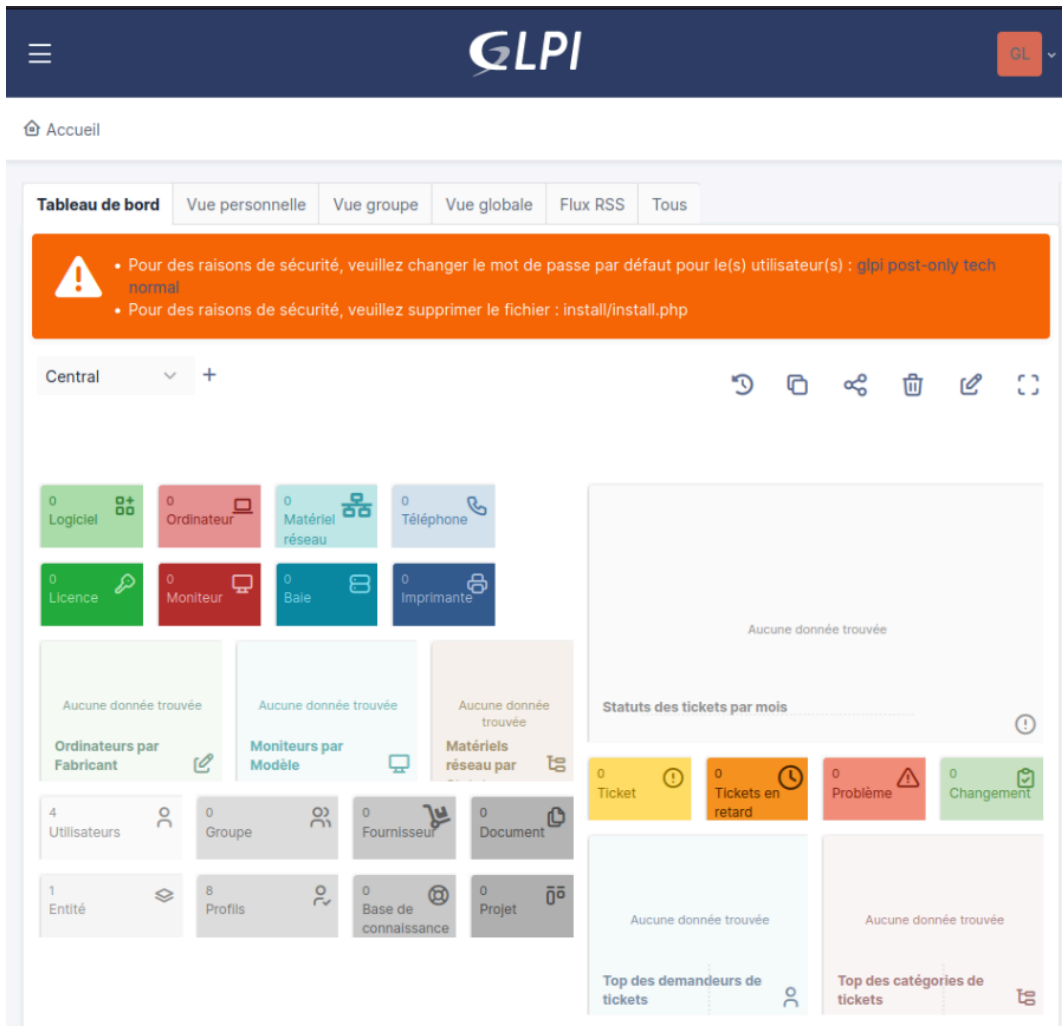
General / SURICATA
SURICATA
Common labels: suricata_logs
1.1.1.1 53 29015
media-lhr6-1.cdn.whatsapp.net A
# query dns 268335880346637 UDP
192.168.115.253 57488
2024-05-21T15:27:01.070669+0200

Log labels
┆ dest_ip 1.1.1.1
┆ dest_port 53
┆ dns_id 29015
┆ dns_rrname media-lhr6-1.cdn.whatsapp.net
┆ dns_rrtype A
┆ dns_tx_id 0
┆ dns_type query
┆ event_type dns
┆ flow_id 268335880346637
┆ job suricata_logs
┆ proto UDP
┆ src_ip 192.168.115.253
┆ src_port 57488
┆ timestamp 2024-05-21T15:27:01.070669+0200

Detected fields
┆ dest_ip "1.1.1.1"
┆ dest_port 53
┆ dns ("type":"query","id":29015,"rrname":"media-lhr6-1.cdn.whatsapp.net","rrtype":"A","tx_id":0)
┆ event_type "dns"
┆ flow_id 268335880346637
┆ proto "UDP"
┆ src_ip "192.168.115.253"
┆ src_port 57488
┆ timestamp "2024-05-21T15:27:01.070669+0200"
┆ ts 2024-05-21T15:27:01.071Z

```

Message syslog de Suricata et Loki



Interface GLPI déployé sur portainer



Contenu du fichier de configuration nftables

```
## IP sur liste blanche
define IP_ACCEPT={87.0.0.1, 87.0.0.2, 87.0.0.3, }
define IP_TECH={192.168.114.129, 192.168.0.29, 192.168.0.28, 192.168.114.1}
## IP du firewall
define IP_FW_PUB={87.0.1.1, 87.0.1.2}
define IP_FW_PRIV=192.168.0.254
define IP_PUB1=87.0.1.1
define IP_PUB2=87.0.1.2
## IP connus
define IP_SRV_LOG=192.168.0.1
define IP_SRV_METRIC=192.168.0.2
define IP_SRV1=192.168.0.3
define IP_SRV2=192.168.0.4
define IP_PROXY=192.168.0.253
## Plages réseau
define NET_SRV_DMZ={192.168.0.1-192.168.0.252}
define NET_DMZ=192.168.0.0/24
## Interfaces
define IF_INTERNET="enp2s0f1"
define IF_DMZ="enp2s0f0"
table ip6 paf {
  chain firewall_in {
    type filter hook prerouting priority 0; policy drop;
  }
}
table ip plouf {
  map tcp_services {
    comment "Liste de paires associant un port à un une adresse IP afin de laisser
passer des services TCP dans le firewall"
    flags constant;
    type ipv4_addr . inet_service : ipv4_addr;
    elements = {
      $IP_PUB1 . 80 : $IP_SRV1,
      $IP_PUB1 . 443 : $IP_SRV1,
      $IP_PUB2 . 443 : $IP_SRV2
    }
  }
  map udp_services {
    comment "Liste de paires associant un port à un une adresse IP afin de laisser
passer des services UDP dans le firewall"
    flags constant;
    type ipv4_addr . inet_service : ipv4_addr;
    elements = {
      $IP_PUB1 . 53 : $IP_SRV1,
    }
  }
  chain firewall_in {
    type filter hook input priority filter; policy drop;
    comment "Règle de gestion d'accès au firewall"
```



```

## On accepte toutes les connexion venant des admins et des serveurs de supervision
ip saddr {$IP_TECH, $IP_SRV_LOG, $IP_SRV_METRIC} accept
ip saddr 127.0.0.0/24 accept

## Fait le mort lors de ping depuis l'exterieur
20
iif $IF_INTERNET ct original protocol 1 ct state new reject with icmp type
hostunreachable

## Accepte tous les connexion déjà établit par le firewall
ct state established,related accept
}

chain firewall_out {
type filter hook output priority filter; policy accept;
comment "Sortie du firewall"
}
chain internet_dmz {
type filter hook forward priority filter; policy drop;
comment "Filtrage du traffic"
## Filtrage des accès à SRV1
ip saddr !=$IP_ACCEPT ip daddr $IP_SRV1 drop
## On accepte SANS analyse le traffic venant des admins
ip saddr $IP_TECH accept
ip daddr $IP_TECH ct state established,related accept
## On accepte AVEC analyse le traffic venant de la DMZ (proxy et serveurs)
ip saddr $NET_DMZ queue num 0 bypass
ip daddr $NET_DMZ ct state established,related queue num 0 bypass
## Accepte tout les connexion déjà DNAT
ct status dnat queue num 0 bypass
}
chain dmz_nat {
type nat hook postrouting priority srcnat;
comment "NAT sortante"

## Translation de toutes les connexions venant des admins et du proxy
ip saddr {$IP_PROXY, $IP_TECH} oif $IF_INTERNET snat to $IP_PUB1 random comment
"Proxy Lambda Users"
## Translation de toutes les NOUVELLES connexions venant de la DMZ
ip saddr $NET_SRV_DMZ oif $IF_INTERNET ct state new snat to $IP_PUB1 random
}
chain internet_nat {
type nat hook prerouting priority dstnat;
comment "NAT entrante DMZ"
## Translation des connexion entrantes uniquement sur les ports déclarés plus en
amont
dnat to ip daddr . tcp dport map @tcp_services
dnat to ip daddr . udp dport map @udp_services
}

```



configuration de promtail pour scraper un fichier de log lors de l'initialisation du système.

```
# This minimal config scrape only single log file.
# Primarily used in rpm/deb packaging where promtail service can be
# started during system init process.
# And too much scraping during init process can overload the complete
# system.
# https://github.com/grafana/loki/issues/11398

server:
  http_listen_port: 9080
  grpc_listen_port: 0

positions:
  filename: /tmp/positions.yaml

clients:
  url: http://localhost:3100/loki/api/v1/push
  scrape_configs:
    - job_name: suricata
      syslog:
        listen_address: 0.0.0.0:1514
        listen_protocol: tcp
        idle_timeout: 60s
        label_structured_data: yes
        labels:
          job: suricata_logs
```



```
version: "3.3"
services:
  mysql:
    image: ubuntu/mysql
    restart: always
    hostname: mysql
    cap_add:
      - SYS_NICE
    volumes:
      - /opt/glpi/storage/mysql:/var/lib/mysql
    env_file:
      - stack.env
    ports:
      - "3306:3306"
  glpi:
    image: elestio/glpi:${SOFTWARE_VERSION_TAG}
    restart: always
    hostname: glpi
    ports:
      - "8000:80"
    volumes:
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
      - /opt/glpi/storage/var/www/html/glpi/:/var/www/html/glpi
      - /opt/glpi/apache/php.ini:/etc/php/8.1/apache2/php.ini
    environment:
      - TIMEZONE=Europe/Paris
  pma:
    image: phpmyadmin
    restart: always
    links:
      - mysql:mysql
    ports:
      - "8008:80"
    environment:
      PMA_HOST: mysql
      PMA_PORT: 3306
      PMA_USER: root
      PMA_PASSWORD: ${ADMIN_PASSWORD}
      UPLOAD_LIMIT: 500M
      MYSQL_USERNAME: glpi
      MYSQL_ROOT_PASSWORD: ${ADMIN_PASSWORD}
    depends_on:
      - mysql
```



9. Bibliographie

Suricata:

<https://docs.suricata.io/en/latest/>

Suricata Integration Ntopng:

https://www.ntop.org/guides/ntopng/third_party_integrations/suricata.html#suricata-integration

Netify l'installation:

<https://www.netify.ai/products/netify-informatics/get-netify/debian>

Ntopng :

<https://www.ntop.org/guides/ntopng/>

Envoyer les données syslog-ng de suricata au ntopng:

https://gist.github.com/mkgin/6f10fec7f0be40b3bb53337a4052b9d3#file-syslog-ng_suricata_ntopng-conf

Solution Source routing:

https://www.tala-informatique.fr/wiki/index.php/Source_routing

Solution Detection Disque, Fin du Projet :

<https://serverfault.com/questions/611182/centos-7-x64-and-hp-proliant-dl360-g5-scsi-controller-compatibility/611210#611210>

