Sécurité dans les labos, l'Onion fait la force

Christophe Saillard, Thomas Keller

Sécurité dans les labos, l'Onion fait la force

Thomas Keller
Observatoire Astronomique de Strasbourg – UMR7550
11 rue de l'Université
67 000 Strasbourg

Christophe Saillard
Observatoire Astronomique de Strasbourg – UMR7550
11 rue de l'Université
67 000 Strasbourg

Résumé

Croyez-le ou non, question sécurité, l'outil idéal trop bien qui fait tout existe.

Comme beaucoup, nous avions besoin d'un outil magique :

- *qui surveille toutes les couches du réseau ;*
- *épluche les logs et les événements sur les serveurs et postes de travail que ce soit sous Linux, macOS ou Windows ;*
- *qui nous prévient quand un truc louche se passe ;*
- *facile à installer et à maintenir ;*
- *gratuit si possible.*

La solution c'est Security onion[1] : une suite d'outils packagés dans une distribution Linux qui embarque des fonctionnalités de NIDS[2] , HIDS[3], SIEM[4], et bien plus, le tout accessible depuis une page web unique (la « SOC[5] »).

Nous allons vous retracer notre aventure, depuis l'événement à l'origine de cette subite envie de tout surveiller, des critères qui nous ont amenés à notre choix, jusqu'aux joies de l'exploitation quotidienne. Mais l'Onion fait aussi pleurer, nous vous présenterons donc également ses inconvénients et ses défauts.

Peler l'Onion est à la portée de toutes et tous, passez voir notre poster et nous rencontrer pour partager notre recette, on vous montrera comment l'émincer sans vous couper.

Et n'oubliez pas : l'Onion fait la force !

Mots-clefs

SIEM, HIDS, NIDS, sécurité

1 Security Onion : https://securityonionsolutions.com/

2 NIDS : Network Intrusion Detection System

3 HIDS : Host Intrusion Detection Systems

4 SIEM : Security Information and Event Management

5 SOC : Security Onion Console

1 Introduction

L’ObAS et le CDS c’est quoi ?

L’Observatoire astronomique de Strasbourg (ObAS) est un Observatoire des Sciences de l’Univers, composante de l’Université de Strasbourg (Unistra) et Unité Mixte de Recherche du CNRS et de l’Unistra.

L’ObAS mène ainsi à la fois des activités :

- de service pour l’ensemble de la communauté astronomique ;
- d’enseignement ;
- de recherche dans de multiples domaines de l’astrophysique ;
- de diffusion des sciences.

L’ObAS héberge donc plusieurs services nationaux d’observation ainsi que les services du Centre de Données astronomique de Strasbourg (CDS), Infrastructure de Recherche du CNRS.

Présentation du SI

Le SI en quelques puces :

- une équipe informatique de 2 agents ;
- une équipe de développement CDS d’une dizaine d’agents ;
- environ 150 postes de travail (75 % sous Linux, 20 % sous macOS, 5 % de Windows) ;
- 2 paires de Firewall redondants Stormshield ;
- 40 switchs Juniper ;
- 40 serveurs physiques / 150 VM ;
- environ 5 Po de stockage (serveurs ZFS, baies DELL Compellent) ;
- une infrastructure authentification centralisée OpenLDAP ;
- un inventaire et une gestion du support GLPI ;
- une solution de métrologie et de supervision basée sur Zabbix/LibreNMS.

Services critiques super méga utilisés depuis l’extérieur

Les services du CDS sont massivement utilisés par la communauté mondiale des astronomes professionnels et amateurs. Ils font l’objet de plusieurs millions de requêtes par jour et doivent donc fonctionner 24/7. Nos infrastructures sont donc dimensionnées en termes de capacité, de performance et de disponibilité pour répondre à cette contrainte (hébergement redondant des services critiques dans 2 datacentres strasbourgeois).

Beaucoup de CDD et de stagiaires

Au total, l’ObAS compte une centaine de personnels permanents (enseignants-chercheurs, personnels techniques et administratifs), mais nous recevons très régulièrement des stagiaires (environ 30 par an) ainsi que des CDD (post-docs, etc.). Il faut donc gérer ce nombre important d’individus présents pour des périodes parfois très courtes.

Acte fondateur de l’évolution de la gestion de la sécurité à l’ObAS

Jusqu’en 2019, la sécurité informatique de l’ObAS se concentrait, comme dans un grand nombre d’établissements, sur les menaces extérieures.

Pour nous en prémunir, nous nous appuyions sur 2 pare-feux redondants pour filtrer les accès entrants vers notre DMZ en n’exposant que le minimum d’IP publiques et ports nécessaires. Ce dispositif de filtrage, couplé à une vraie démarche de sensibilisation des collègues (en plus des antivirus et du chiffrement des postes de travail), nous paraissait suffisant au regard du temps que nous pouvons consacrer à la sécurité et à la taille réduite de l’équipe.

Néanmoins, un jour, suite au plantage d’un de nos serveurs, nous découvrons qu’une personne de l’équipe, récemment arrivée, ne voulant pas « déranger » son collègue de bureau « admin » du serveur, choisit de faire preuve d’initiative et décide de se débrouiller seul pour obtenir les droits *root*. Il télécharge, compile et exécute des *exploits* qui, au final, font crasher le serveur. Par la suite, l’analyse du poste de travail de ce hacker maladroit a révélé une activité malveillante qui durait depuis plusieurs mois. On ne s’étendra pas plus sur cet incident pour des raisons de confidentialité, cependant cet événement a provoqué une réelle prise de conscience des enjeux de la sécurité au sein du laboratoire.

Avec le soutien de notre direction, nous avons donc pris les choses à bras le corps en élaborant un plan d’amélioration pour élever notre niveau de sécurité face aux menaces internes en nous appuyant sur une solution de gestion de la sécurité… aux petits oignons…

2 Phase d’étude et de maquettage

2.1 Organisation

L’équipe informatique encadre régulièrement des stagiaires pour avancer sur certains projets structurants. Ainsi, grâce à la prise de conscience de la direction de l’importance de l’élévation de notre niveau de sécurité, nous avons choisi de nous appuyer sur un étudiant alternant de licence professionnelle spécialisée en administration réseau pour renforcer l’équipe sur un temps long. Nous l’avons donc encadré tout au long de ce projet sur fond de COVID et de travail en distanciel.

2.2 Définition des besoins et cahier des charges

Partant du principe du « plus jamais ça », nous avons aussi identifié **3 besoins principaux** :

1. Gagner en visibilité entre nos réseaux internes et vers internet (NIDS);
2. Gagner en visibilité directement sur les équipements terminaux, Linux, macOS et Windows, serveurs et postes de travail (HIDS). En effet, dans le même VLAN, les flux « poste à poste » ne sont pas directement visibles, nous avions donc besoin d’avoir une vue directe sur le poste en cas de tentative d’intrusion ou de comportement suspect (exécution de commandes inhabituelles ou accès à certains fichiers hors du contexte d’utilisation standard de l’usager)
1. Avoir une vue globale sur l’ensemble des alertes et des informations liées à la sécurité de notre SI (SIEM).

En plus de ces 3 besoins, nous avons établi une liste d’actions malveillantes contre lesquelles nous prémunir. Cette liste a ensuite été complétée lors de notre phase d’étude par des outils open source pour traiter potentiellement la détection, le blocage et la remontée d’alerte.

En voici la synthèse :

Action malveillante	Détection	Blocage	Remontée d'alerte
Activité suspecte inter VLAN, depuis et vers Internet (NIDS)	Surricata		
Activité suspecte sur les équipements terminaux (HIDS)	Wazuh		Logs
IP spoofing	Arpwatch		E-mail
Brute force SSH	Fail2ban, Crowdsec	iptables	
Scan de port	PSA	firewall (UFW)	

Figure 1 - Typologie des actions malveillantes et outils associés

2.3 À la recherche de la bonne solution

Lors de la phase d'étude nous permettant de déterminer la liste des solutions open source adaptées à notre cahier des charges, nous découvrons l'existence de *Security Onion* (SO) qui intègre nativement déjà une partie des applicatifs recensés (*Surricata*, *Wazuh*). *À priori*, SO répondait donc à l'ensemble de nos besoins (NIDS, HIDS et SIEM).

En plus des outils open source que nous souhaitons tester, nous prenons contact avec la société Darktrace[6] qui équipe l'Université de Lyon 2 avec sa solution de gestion centralisée de la sécurité. Nous avons donc eu l'opportunité de tester cette solution dans le cadre d'un « POV »[7] de plusieurs semaines (durée plus longue que d'habitude pour cause de confinement).

Darktrace est une *appliance* NIDS qui, placée au bon endroit sur le réseau voit passer tout le trafic et qui, après une période d'apprentissage (15 jours dans notre cas) est capable de générer des alertes comportementales qui mettent en lumière les actions et menaces potentielles qui sortent de la « norme ». L'interface « Threat Visualizer » est assez attrayante et l'éditeur de la solution promet de ne pas passer plus de 15 minutes par jour pour faire le tri des alertes. Enfin, pour quelques euros en plus une technologie baptisée « Antigena » assure une réponse autonome. Le rêve quoi…

Au final, malgré ses qualités, et son indéniable intérêt, cette solution boostée à l'IA (oui déjà à l'époque) n'a pas été retenue, car hors budget pour notre structure malgré des ambitions à la hausse concernant la sécurité. Par ailleurs, la partie HIDS également ne permettait pas de superviser notre parc Linux.

Nous misons donc tout sur SO et nous lançons dans une phase de tests et de maquettage pour finalement l'adopter au quotidien. Parallèlement, nous continuons à mettre en place l'implémentation de l'ensemble des autres outils identifiés (Arpwatch, Fail2ban, UFW, PSA).

3 Présentation de Security Onion

« Security Onion is a free and open platform built by defenders for defenders »

Vous l'aurez deviné, ce n'est pas du *made in France*, il n'y a que les Américains pour présenter un produit de cette façon. Créé en 2008, SO est une suite cohérente de nombreux logiciels libres de sécurité. Initialement basée sur une distribution Ubuntu, SO s'installe aujourd'hui uniquement depuis une ISO basée sur Oracle Linux 9.

6 https://darktrace.com/fr/products/network

7 POV : Proof Of Value

On peut classer SO dans la catégorie des SIEM : il intègre notamment des fonctions de NIDS et de HIDS et agrège les informations collectées par toutes les briques applicatives dans un tableau de bord unique, la Security Onion Console (SOC), qui permet de les exploiter de façon efficace.

3.1 Où dois-je mettre l'Onion ?

Pour être efficace, une solution de type NIDS doit pouvoir analyser l'ensemble des flux en transit sur le réseau. Il faut donc placer une ou plusieurs sondes au(x) bon(s) endroit(s) et avoir la capacité de lui envoyer des flux réseau à analyser. Chez nous le bon endroit s'est avéré être la patte interne de nos pare-feux (redondants actif/passif) par laquelle transitent tous les flux inter VLAN qui sont routés directement sur les pare-feux. En effet, c'est via ces interfaces que transite l'ensemble des flux entrants et sortants entre nos VLAN internes mais aussi vers Internet (via Osiris, notre MAN, et Renater).

Reste à trouver ensuite le bon moyen de faire transiter ce trafic vers notre sonde. Pour ce faire, la solution la plus élégante consiste en l'utilisation d'un « TAP » (Traffic Access Point) qu'on place de façon transparente en coupure au milieu des flux à analyser. Un tel dispositif n'est pas gratuit et nous avons donc au départ utilisé les moyens à notre disposition pour nous en passer.

Vous trouverez en annexe, une présentation détaillée de la mise en œuvre de ce TAP et de sa version « low cost ».

3.2 Installation et administration

Il suffit de télécharger un fichier ISO[8] sur le dépôt de Security Onion et de l'installer sur votre serveur. Il y a plusieurs types d'installation en fonction du contexte d'utilisation :

- *standalone* où toutes les fonctionnalités, la collecte et le stockage des données sont hébergés sur un seul serveur ;
- multisites, si vous avez besoin de plusieurs points de collecte réseau.

Initialement, l'administration de la solution se faisait uniquement en CLI via des scripts inclus basés sur Salt[9]. Pour la gestion des mises à jour (système hôte et ensemble applicatif SO), l'outil SOUP (Security Onion Updater) est également fourni. Dans les dernières versions, l'administration peut se faire de plus en plus directement via la SOC.

3.3 Prérequis matériel

En mode *standalone* : 4 CPU, RAM 16 Go, stockage 200 Go, 2 interfaces réseau. Il s'agit vraiment d'un minium. Dans notre cas, nous utilisons un serveur avec 64 Go RAM, un CPU 12 cœurs, 2 cartes réseau Ethernet SFP+ 10 Gb/s pour la collecte, une carte Ethernet 1 Gb/s pour le management et 88 To d'espace de stockage utile. La durée de rétention dépend directement de la quantité de trafic collectée. Ainsi, avec cette capacité de stockage, nous conservons un historique des flux de 2 semaines pour un débit moyen de trafic observé montant et descendant d'environ 400 Mb/s . Il est également possible d'appliquer des filtres pour réduire le volume de collecte.

3.4 Les différentes couches de l'Onion

Pour présenter une vue synthétique par l'intermédiaire de la SOC, SO s'appuie sur plusieurs briques logicielles installées automatiquement via l'ISO et exécutées dans des conteneurs Docker.

8 https://github.com/Security-Onion-Solutions/securityonion/blob/2.4/main/DOWNLOAD_AND_VERIFY_ISO.md

9 https://saltproject.io/

Fidèle au modèle en couche, en voici un résumé pour la fonction NIDS :

1. Collecte des flux réseau : *Steganographer*[10]
2. Analyse de flux réseau (IDS, Intrusion Detection System) : *Suricata*[11] (basé sur des signatures) et *Zeek*[12] (inspection de paquet)
3. Parsing et stockage des logs : *Elasticsearch*[13]

La fonction HIDS est assurée par l'*Elastic Agent*[14] qui doit être installé sur chaque équipement terminal (Linux, macOS, Windows) et qui pour l'instant se révèle relativement gourmand en ressources. Les logs des agents sont ensuite transmis vers l'*Elasticsearch* du SO. À noter qu'il est également possible de faire remonter directement les syslogs d'un équipement terminal directement à SO.

La présentation et exploitation de l'ensemble des données collectées par SO se fait par l'intermédiaire de l'interface Web appelée SOC.

3.5 La face visible de l'Onion

La SOC propose le menu suivant :

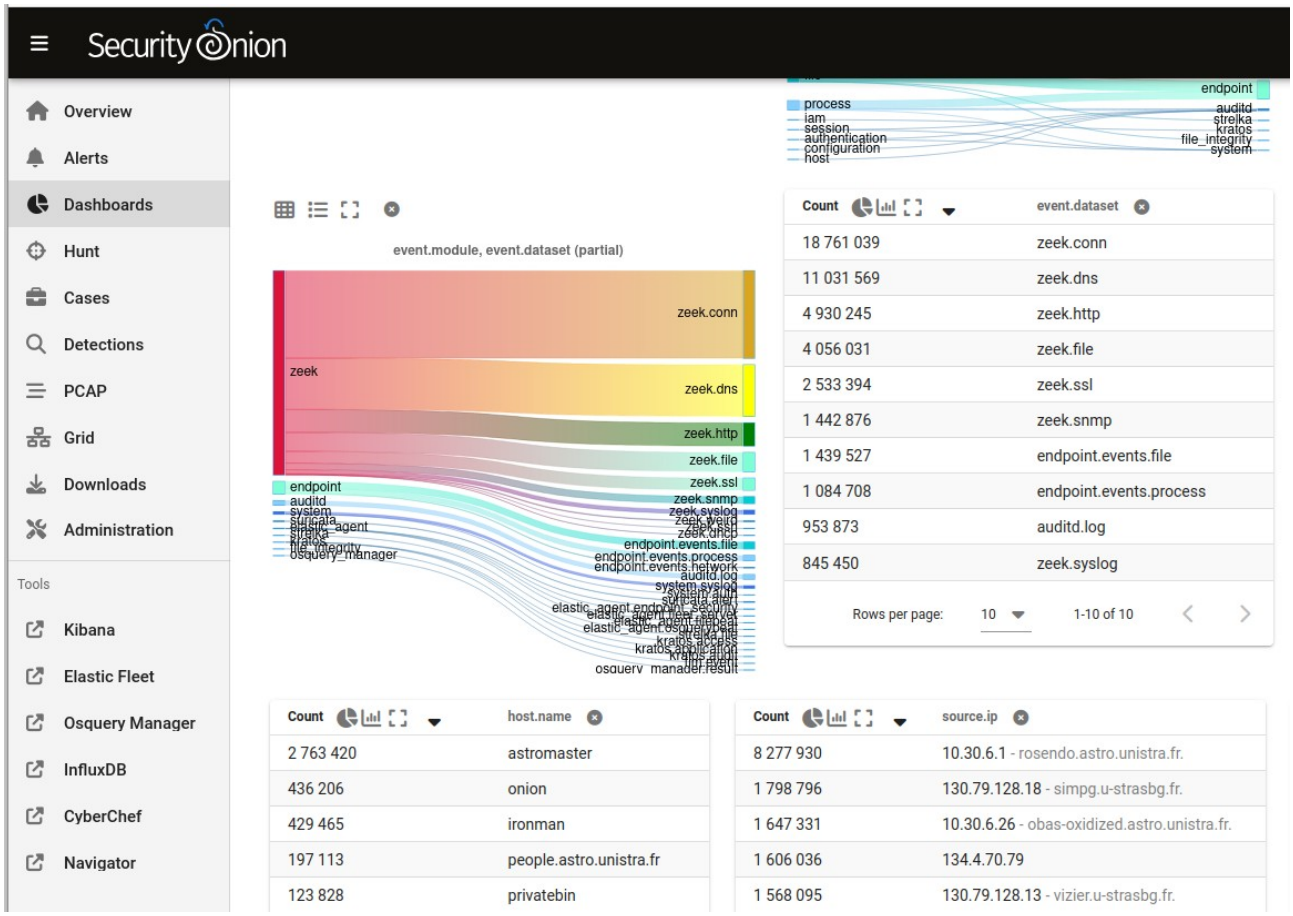

Figure 2 - Vue de la Security Onion Console

10 https://github.com/google/stenographer

11 https://suricata.io/

12 https://zeek.org/

13 https://www.elastic.co/fr/elasticsearch

14 https://www.elastic.co/fr/elastic-agent

Elle donne accès à plusieurs fonctionnalités :

- **Alerts :** centralise toutes les remontées d'alertes avec des outils permettant de les « trier » selon un grand nombre de critères (niveau de criticité, IP source, port destination, plage horaire, etc.). Un menu contextuel permet d'effectuer des actions sur les éléments sélectionnés (dump réseau, inclusion ou exclusion de la valeur pointée, requête sur VirusTotal, etc.) ;
- **Dashboards :** accès à des tableaux de bord, entièrement personnalisables via des requêtes en langage OQL (Onion Query Langage, similaire à Elasticsearch) ;
- **Hunts :** similaire au *dashboard* mais plus précis pour la recherche de menaces ;
- **Cases :** pour réaliser le suivi d'incidents ;
- **Detections** : personnalisation des règles de détection ;
- **PCAP** : permet de récupérer un flux détecté au format PCAP[15], pour une analyse avec Wireshark par exemple ;
- **GRID** : vue globale des composants de SO, pertinent notamment pour la gestion du mode multisite avec plusieurs nœuds de collecte ;
- **Downloads** : accès au téléchargement des paquets *Elastic Agent* préconfigurés ;
- **Administration** : paramétrage de SO.

La partie « Tools » donne un accès direct à des outils intégrés :

- **Kibana** : permet de faire des requêtes *Elasticsearch* « natives » ;
- **Elastic Fleet** : configuration des *Elastic Agents* ;
- **Osquery Manager** : interrogation des *Elastic Agents* ;
- **InfluxDB** : supervision et monitoring du serveur SO ;
- **CyberChef** : véritable couteau suisse cyber qui regroupe plein d'outils (conversion, calcul de hash, etc.) ;
- **Navigator** : base de connaissance « MITRE ATT&CK »[16].

4 Comment on cuisine l'Onion au quotidien ?

4.1 Onion unboxing

Sorti du carton, SO génère de base une énorme quantité d'alertes à éplucher, à mettre en rapport avec le trafic « astronomique » généré par les services du CDS (5 millions de requêtes/jour). **Un important travail de paramétrage et d'adaptation est donc nécessaire pour réduire le bruit et faire ressortir les alertes pertinentes** (par l'intermédiaire de la fonction « Detections » de la SOC).

En termes de méthodologie, nous sommes partis d'une fenêtre temporelle très courte (par exemple 3 h), nous avons fait un premier tri et ensuite incrémenté la fenêtre (6 h, 12 h, 24 h, etc.) et refait le travail de tri pour ne conserver que les alertes pertinentes.

Pour mémoire, notre capacité de stockage (88 To) nous permet de conserver environ 3 semaines d'historique de trafic.

15 https://en.wikipedia.org/wiki/Pcap

16 https://attack.mitre.org/

Cette phase de tuning des alertes a été fastidieuse : les alertes remontées sont souvent très pointues et il faut vraiment les analyser avant de les ignorer en prenant des risques inconsidérés.

Néanmoins, cette opération a été très enrichissante sur la compréhension des différents flux externes, mais aussi internes qui transitent sur notre réseau. Cela a mis en évidence des potentielles failles de sécurité (liste non exhaustive) souvent liées à des oublis ou des erreurs de configuration au niveau de nos pare-feux :

- alertes *brute force* SSH, sur des IP normalement pas exposées ;
- interception de mots de passe en clair sur certains serveurs ;
- page d'administration de serveurs WordPress exposée.

4.2 L'Onion au quotidien

Nous consultons la SOC 3 à 4 fois par semaine brièvement pour gérer les urgences. Idéalement nous aimerions y consacrer plus de temps pour un vrai travail de fond (environ 30 minutes par jour). En effet, dans une équipe restreinte où la sécurité n'est pas une tâche à plein temps avec une personne dédiée, celle-ci est malheureusement très vite reléguée au second plan.

Dans la SOC, les alertes sont classées par niveau de criticité (low, medium, high). On utilise donc le *Dashboard* en excluant les alertes peu critiques (low). On prend ensuite quelques minutes pour traiter les alertes qui nous semblent les plus critiques.

Depuis 4 ans, nous n'avons pas identifié de façon proactive d'incident de sécurité, néanmoins, la routine quotidienne de consultation de la SOC nous permet d'identifier régulièrement des failles de sécurité relatives à des versions de packages vulnérables sur certains de nos services (grâce à une remontée des CVE[17] dans l'analyse des flux). Nous sommes maintenant beaucoup plus vigilants quant à l'application des mises à jour automatisées de sécurité sur les serveurs (utilisation des *unattended upgrade* Ubuntu[18]).

4.3 L'Onion pour lever les doutes

Cependant, suite à des demandes assez régulières de levée de doute remontée par le CERT-OSIRIS[1], nous avons été en mesure de fournir des éléments précis (preuve d'établissement de connexion entre hôtes, identification d'une IP interne natée) collectés grâce à *Security Onion.* On ne travaille plus à l'aveugle et la recherche d'information est vraiment très performante.

4.4 L'Onion, les équipements terminaux et les autres outils de sécurité

Nous utilisons *Ansible*[19] pour gérer l'automatisation du déploiement de configuration et de paquets sur notre parc de serveurs et de postes de travail sous Linux. Nous avons donc créé un *playbook* dédié à la sécurisation des équipements terminaux qui installent les outils suivants :

- *Elastic Agent* (pour la fonction HIDS) en remplacement de l'agent *Wazuh*[20] sur lequel SO s'appuyait jusqu'à la version 2.3 (version actuelle : 2.4) ;
- *Fail2ban*[21], pour bloquer les attaques par force brute ;

17 https://www.cve.org/

18 https://guide.ubuntu-fr.org/server/automatic-updates.html

19 https://www.ansible.com/

20 https://wazuh.com/

21 https://github.com/fail2ban/fail2ban

- *Crowdsec*[22], pour bloquer les tentatives de connexion depuis des IP recensées dans des listes communautaires ;
- *UFW*[23], pour définir une politique de filtrage locale minimale.

Fail2ban, *Crowdsec* et *UFW* ont une action directe sur l'équipement terminal (blocage d'IP) alors que l'*Elastic Agent* remonte des informations à la SOC. Il faut donc la consulter régulièrement pour constater les éventuelles alertes. Nous avons pour l'instant peu de recul sur l'*Elastic Agent*, car nous utilisions auparavant *Wazuh*.

À noter que pour les postes sous Windows, l'installation de l'Elastic Agent est manuelle au regard du petit parc à gérer (moins de 10 postes). Pour les postes sous macOS, nous venons de déployer une solution de déploiement centralisée basée sur *Munki*[24]

5 Notre avis sur l'Onion

5.1 L'Onion qui fait rire…

5.1.1 Une solution clé en main

SO intègre une collection d'outils sous un même environnement et évite de devoir déployer et d'administrer une multitude de serveurs dédiés. C'est la vraie valeur ajoutée de cette solution. Par ailleurs, au-delà des fonctionnalités qui nous intéressaient au départ, nous avons découvert d'autres outils pertinents directement intégrés : règles YARA[25], *CyberChef*, etc.

La maintenance est également simplifiée : les mises à jour et configuration se font en un même point.

5.1.2 Une solution gratuite

SO existe dans une version gratuite dont le support est communautaire. Le jeu de règles *Suricata* est donc basé sur sa version « open », il y a donc un petit délai avant la mise à disposition des règles les plus à jour par rapport à la version « pro » qui est plus réactive.

SO fonctionne très bien sur du matériel « recyclé » ou disponible dans le cadre du marché MATINFO[26], mais là encore il est possible de faire l'acquisition d'*appliance* proposées par SO.

5.1.3 Une prise en main relativement simple

Il existe de nombreux tutoriels officiels pour appréhender la solution et la documentation est de qualité.

5.1.4 Un projet très actif

Le projet SO est très dynamique et les mises à jour sont fréquentes, la SOC s'enrichit très régulièrement en fonctionnalités et intègre directement de plus en plus de paramétrages pour éviter de mettre trop les mains dans le cambouis.

22 https://www.crowdsec.net/

23 https://doc.ubuntu-fr.org/ufw

24 https://www.munki.org/munki/

25 https://yara.readthedocs.io/en/latest/

26 https://www.matinfo.fr/

5.2 L'Onion qui fait pleurer…

5.2.1 Un projet trop actif ?

Les mises à jour sont **très** voire **trop** fréquentes. Il est parfois difficile de suivre toutes les évolutions. Heureusement, les modifications sont souvent sous-jacentes et la console reste relativement stable. Au-delà des mises à jour, certains outils sont remplacés par d'autres. Cela a été notamment le cas pour *Wazuh*, dont l'intégration native à SO faisait partie des points forts de la solution et qui a été remplacé il y a peu par l'*Elastic Agent*. S'agissant d'un agent à déployer sur tous nos postes, le changement est plus complexe à gérer.

5.2.2 Un projet de moins en moins « open » ?

La solution s'éloigne un peu de sa philosophie initiale et certaines fonctionnalités sont réservées à la version bénéficiant d'un support, par exemple l'envoi de notifications. SO tente d'orienter ses usagers vers ses différentes *appliances* avec contrat de support associé. Néanmoins nous continuons à l'utiliser sur notre ancien serveur DELL sans aucun souci.

5.2.3 Du temps pour faire bien les choses

SO n'est pas une solution magique, son paramétrage initial et son utilisation quotidienne nécessitent un temps assez important, mais il est clair que ce temps serait bien plus considérable si nous avions dû déployer chacune des briques par nos propres moyens.

Conclusion et perspectives

La gestion de la sécurité de nos systèmes d'information devrait être une préoccupation quotidienne. Dans les petites structures, il est quasi impossible d'y consacrer le temps nécessaire et nous manquons aussi souvent de connaissances et de compétences pour traiter efficacement cette problématique. Nous considérons aujourd'hui qu'une solution comme *Security Onion* apporte un vrai plus pour élever notre niveau de sécurité, car elle nous a permis de gagner en visibilité sur les flux réseau et sur nos équipements terminaux. Par ailleurs, sa mise en œuvre a mis en évidence des incohérences dans notre politique de filtrage et sur les vulnérabilités potentielles de nos systèmes. Dans les prochaines semaines, nous allons faire évoluer notre architecture SO actuellement en *standalone* vers une installation multisites pour répondre à un changement de topologie de notre infrastructure réseau dont le routage sera très prochainement réalisé sur plusieurs sites.

La question de la sécurité ne saurait être abordée uniquement que par un angle technique et il est important de noter que, suite à l'incident réseau à l'origine de notre prise de conscience de nos faiblesses internes, 2 actions « administratives » importantes ont été réalisées :

- notre règlement intérieur a été modifié et il intègre maintenant des parties consacrées à la sécurité informatique (usages, durée de validité des comptes, conservation des données…) ;
- le rôle de CSSI (Correspondant Sécurité du Système d'Information) est clairement identifié dans notre organigramme.

Enfin, l'Université de Strasbourg lance une démarche d'homologation de l'ensemble des services informatiques hébergés dans ses laboratoires et composantes en s'appuyant sur les outils de l'ANSSI[27]. Cette tâche qui s'annonce assez fastidieuse permettra toutefois d'élever encore notre niveau de sécurité et nous nous inscrivons dans cette démarche avec enthousiasme.

Mais n'oubliez pas le plus important : l'Onion fait la force !

27 https://monservicesecurise.cyber.gouv.fr/

Annexe

Comment brancher un NIDS dans son réseau ?

Mirroring low-cost

Pour commencer nos tests, nous avons donc utilisé la fonction de *port mirroring* de nos switches de concentration (stackés en un « virtual-chassis », placés en tête de réseau) sur lesquels sont connectés les pare-feux. En l'occurrence, il s'agit d'un switch *Juniper EX*, chez qui cette fonctionnalité est baptisée « trafic analyzer ». Cette fonctionnalité peut impacter les performances du switch et il est recommandé de ne pas l'utiliser en permanence. Nous ne l'avons donc utilisée au départ que lors des tests en gardant en tête qu'il faudrait trouver une solution pérenne à l'avenir.

Autre limitation : on ne peut définir qu'un seul port de sortie vers lequel diriger les flux « mirrorés ». En cas de panne du switch sur lequel se trouve le port de sortie, on perd donc le trafic vers la sonde. C'est un moindre mal et ça n'est jamais arrivé.

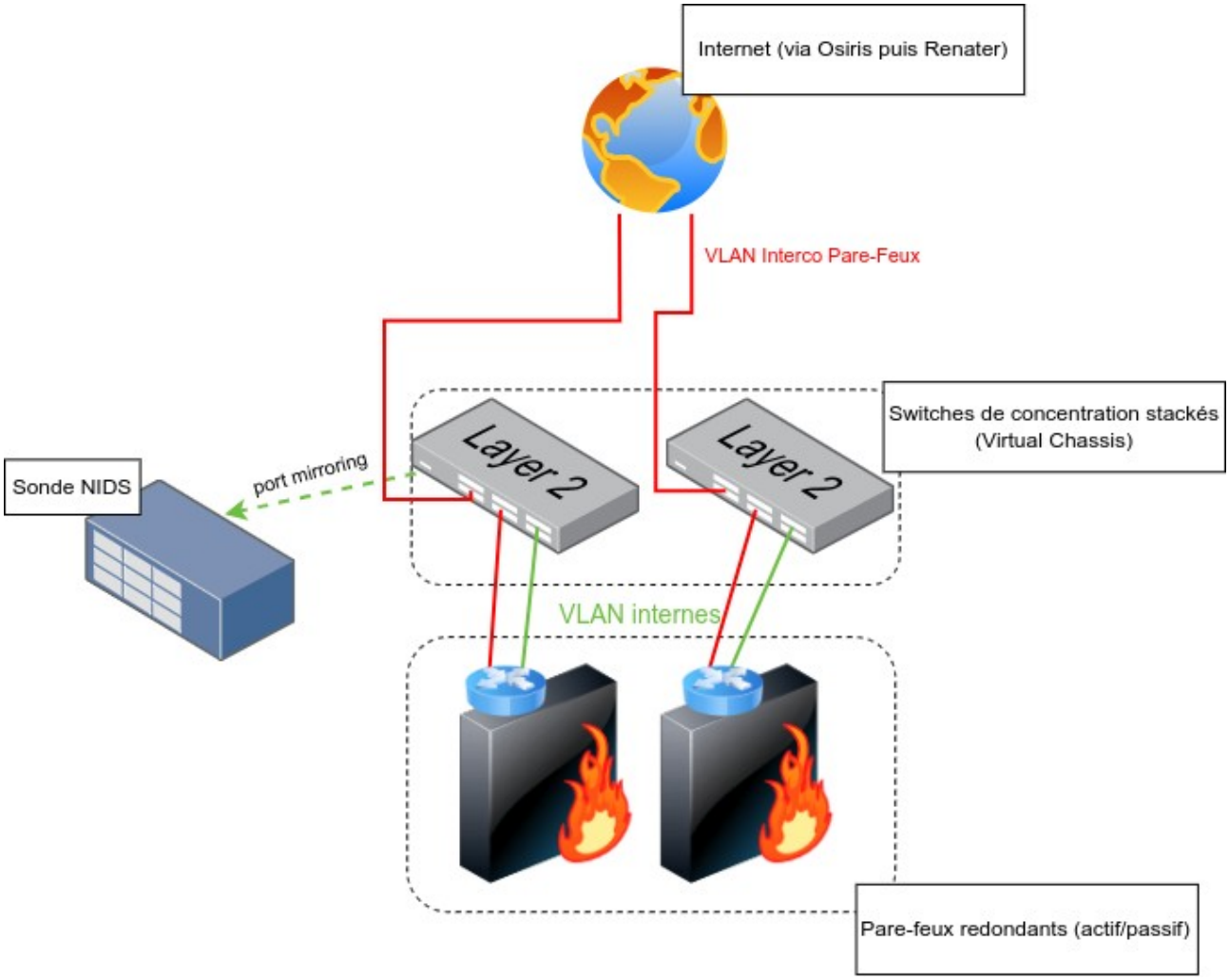

Figure 3 - Interconnexion de la sonde NIDS dans notre réseau, version low-cost

Configuration d'un « trafic analyzer » Juniper

Principe de base : on « mirrore » (donc en input) le flux *egress* et *ingress* des interfaces qui relient notre pare-feu et notre switch et faire ressortir le tout (output) vers une interface connectée au à la sonde.

```
forwarding-options {

analyzer {
       tap-onion {
           input {
               ingress {
                   interface xe-0/0/12.0;
                   interface xe-1/0/12.0;
               }
               egress {
                   interface xe-0/0/12.0;
                   interface xe-1/0/12.0;
               }
           }
           output {
               interface xe-0/0/22.0;
           }
       }
   }
}
```

Figure 4 - configuration d'un trafic analyser sur un switch Juniper

Analyse du trafic avec un vrai TAP

Un TAP est un équipement passif qu'on va mettre en coupure sur un lien optique réseau pour « recopier » le signal montant et descendant. Il est impératif de garantir le bon fonctionnement de ce dispositif qui peut entraîner une coupure totale de notre réseau en cas de dysfonctionnement. Par ailleurs, il faut également s'assurer que la confidentialité du trafic est conservée. Nous avons donc choisi un équipement certifié par l'ANSSI[28] pour être totalement serein.

En l'occurrence, nous avons fait l'acquisition d'un TAP optique de chez *Allot*, modèle TAMOD-OWL-850[29] pour recopier un seul lien optique. Nous y avons consacré un budget d'environ 1 600 € HT. Par économie, nous ne supervisons donc pas le lien optique de notre pare-feu de secours qui dans les faits n'est utilisé que lors des opérations de mises à jour pour éviter les interruptions de services.

Nous avons dû également prévoir deux interfaces SFP+ sur notre serveur sonde, car le TAP transmet le trafic qu'il voit en *ingress* et en *egress* sur deux fibres séparées qu'il faut ensuite brancher sur le canal entrant du module SFP+ de la sonde.

Le schéma ci-après détaille l'interconnexion de la sonde par l'intermédiaire du TAP.

28 https://cyber.gouv.fr/produits-certifies/tap-optique-version-tamod-owl-850-tamod-owl-1310-et-tamod-owl-1550

29 https://www.allentis.eu/tap-optiques

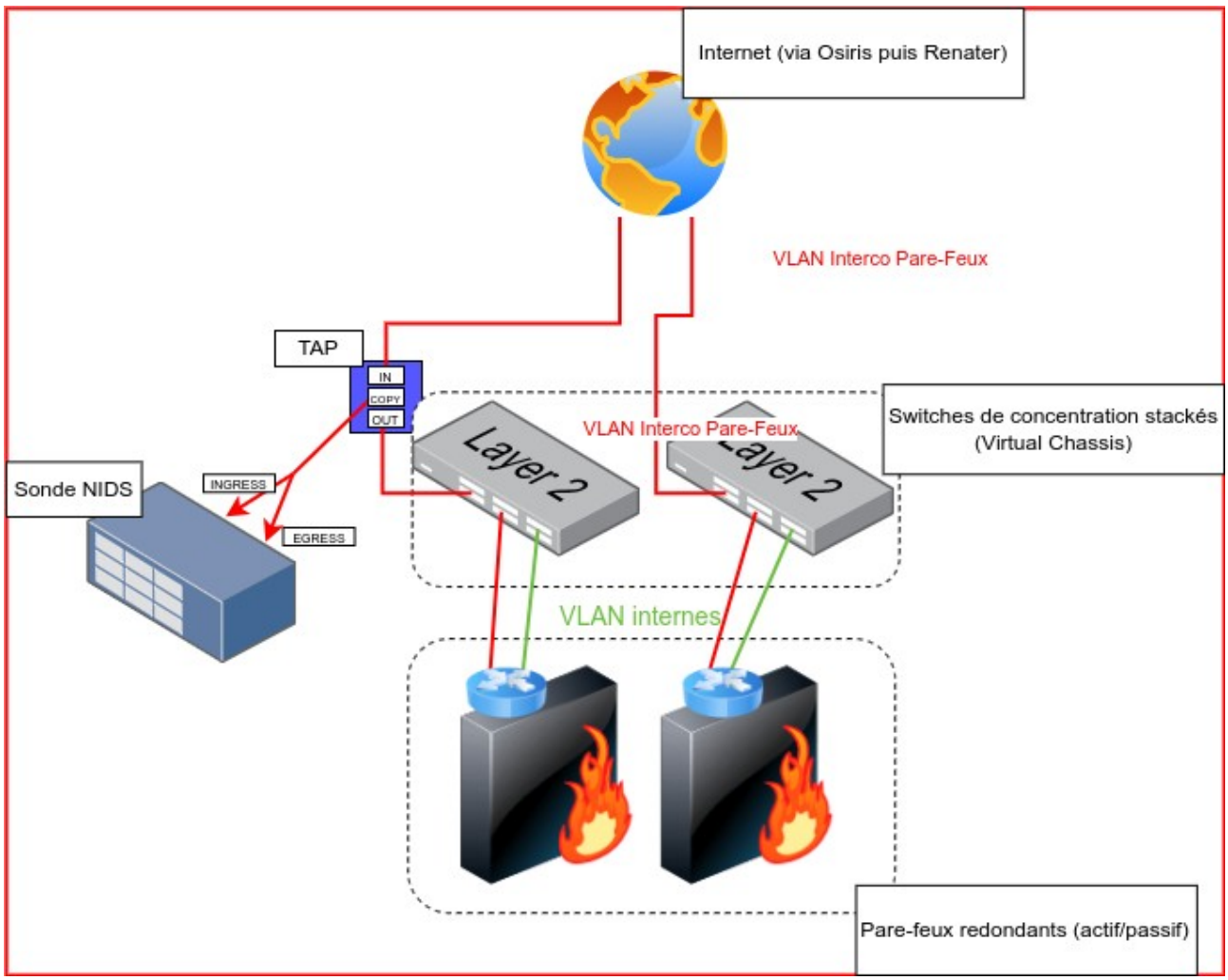

Figure 5 - Interconnexion de la sonde NIDS dans notre réseau via un TAP

La figure 6 est une photo du TAP. En haut on voit la jarretière en « Y » qui part vers la sonde avec une couleur spécifique en rouge et en bleu.

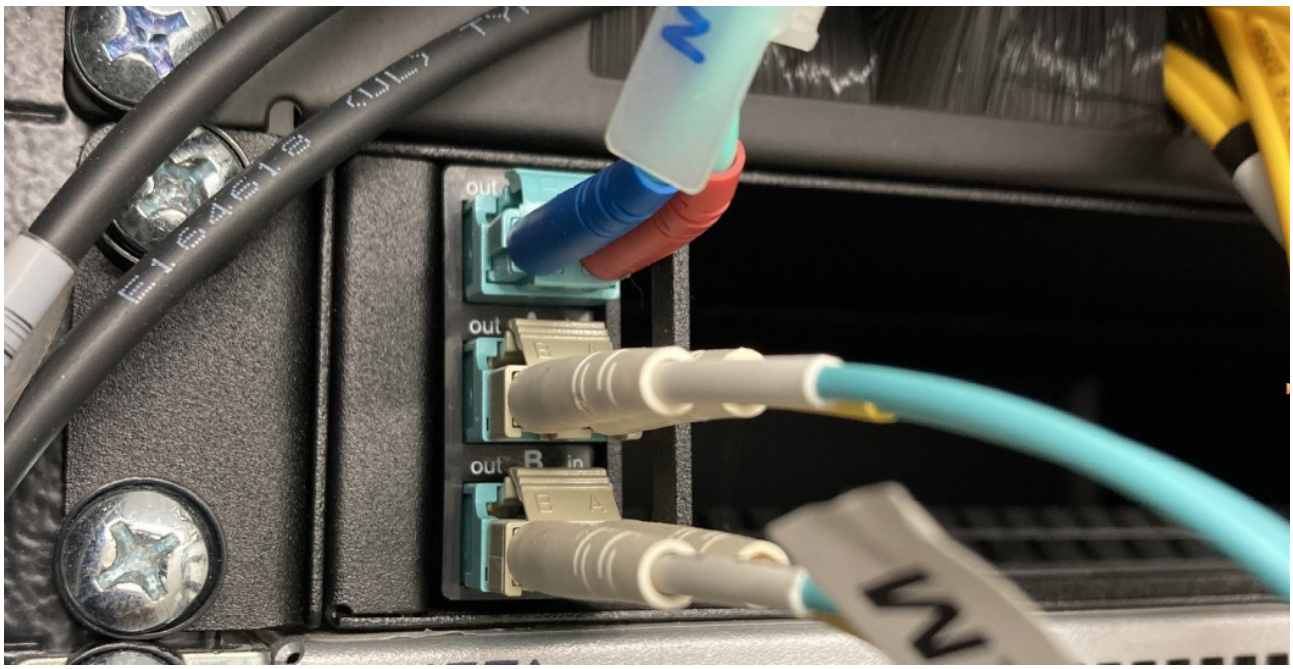

Figure 6 - Photo du TAP

Sur la figure 7, on voit une photo de l'extrémité de la jarretière en « Y » se sépare donc en deux connecteurs pour faire remonter les flux ingress et egress vers le côté « input » de nos 2 interfaces SFP+ de la sonde SO.

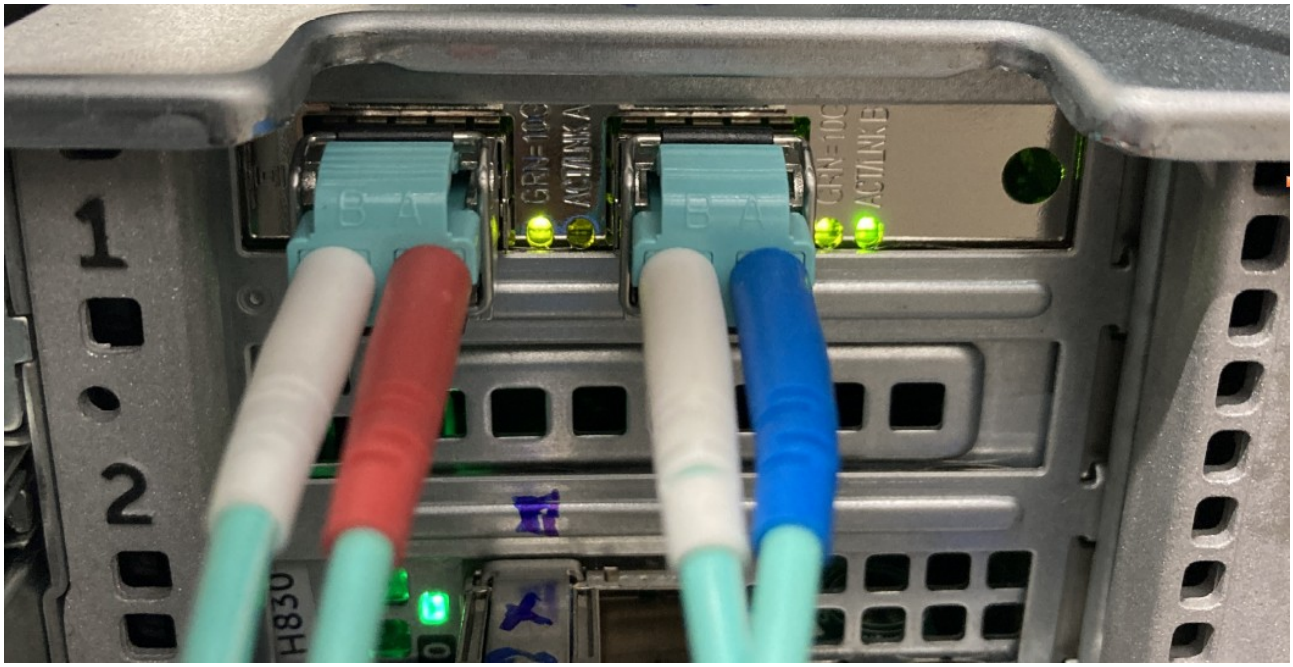

Figure 7 - Interconnexion des interfaces de la sonde sur le TAP

Bibliographie

[1] Guilhem BORGHESI, Magali DAUJAT, Marc HERRMANN. Présentation du CERT OSIRIS. Dans Actes de la conférence JRES2013, décembre 2013, https://2013.jres.org/archives/20/index.htm