Surveiller un réseau de laboratoire avec Suricata

Cyril Bras
CNRS/CERMAV
Service système d'information
Domaine universitaire
601, rue de la chimie
38400 St Martin d'Hères

Marek Bertovič
CNRS/CERMAV
Service système d'information
Domaine universitaire
601, rue de la chimie
38400 St Martin d'Hères

Résumé

Si vis pacem, para bellum[1]*…*

Dans le contexte actuel, où les cyberattaques sont de plus en plus fréquentes, il est illusoire pour toute structure reliée au réseau Internet de penser pouvoir y échapper. Courant novembre 2014, le CERMAV subissait une attaque réussie sur son serveur web institutionnel provoquant une interruption de service de plusieurs heures. Les méthodes utilisées par les pirates auraient sans doute pu être détectées plus rapidement si nous étions équipés d'un système de détection d'intrusion (IDS). D'autant plus que quelques semaines auparavant, au cours des journées sécurité C&esar 2014 à Rennes, nous avions assisté à la présentation d'un IDS, il s'agissait de SURICATA.

Il n'en fallait pas moins pour se décider à franchir le pas et équiper les installations informatiques du laboratoire de ce système. Cependant, nous avons rapidement constaté que le flux de données était difficile à traiter sans une interface visuelle, c'est pourquoi nous avons testé plusieurs solutions d'affichage qui vous serons présentées (SELKS &SNORBY).

La mise en place de cet outil a rapidement été rentabilisée par la détection de machines compromises par des malware ou encore de sites Internet institutionnels utilisant un codage en base 64 pour la protection de leurs mots de passe…

Mots-clefs

IDS, sonde, probe, snorby, selk, kibana, défense active, suricata

1 Introduction

Le CERMAV est un laboratoire de recherche fondamentale sur les glycosciences, unité propre du CNRS (UPR5301), situé sur le campus universitaire de Grenoble. Il comporte un effectif moyen d'environ 120 personnes dont 60 permanents. A la différence des autres laboratoires du campus (qui sont gérés par les DSI[2] des universités grenobloises), le service Système d'Information (composé de deux informaticiens et

[1] « Qui veut la paix prépare la guerre » en français
[2] Direction des systèmes d'information

d'un stagiaire pour le projet IDS[3]) doit, entres autres missions, superviser le trafic réseau et assurer la sécurité des connexions. Comme toute structure reliée au réseau Internet (1), il est illusoire de penser que le CERMAV ne soit pas victime de cyberattaques. La meilleure des solutions que l'on puisse appliquer est de s'y préparer afin de pouvoir réagir de façon adéquate et protéger au mieux le patrimoine scientifique.

2 Contexte

Une note du CNRS (2) sur le bienfondé de la mise en œuvre d'éléments de défense active sur les réseaux des laboratoires, nous a conduits à réfléchir sur l'opportunité de mettre en place ce type d'élément. Nous avions jusqu'en 2012 une sonde utilisant le logiciel SNORT, outre une configuration fastidieuse et une machine incapable de faire face au trafic généré, nous avions abandonné le projet.

C'est au cours des journées sécurité C&esar 2014 à Rennes, où nous avons assisté à la présentation (3) d'un autre logiciel de détection d'intrusion, SURICATA. L'idée de remettre en place des sondes de détection d'intrusion est venue. De plus, courant novembre 2014, le serveur web institutionnel du laboratoire était victime d'une attaque qui eut pu être détectée plus rapidement avec une sonde de détection d'intrusion. Ces éléments nous motivent donc à tester cet IDS sur le réseau de notre laboratoire afin de détecter d'éventuelles intrusions ou anomalies.

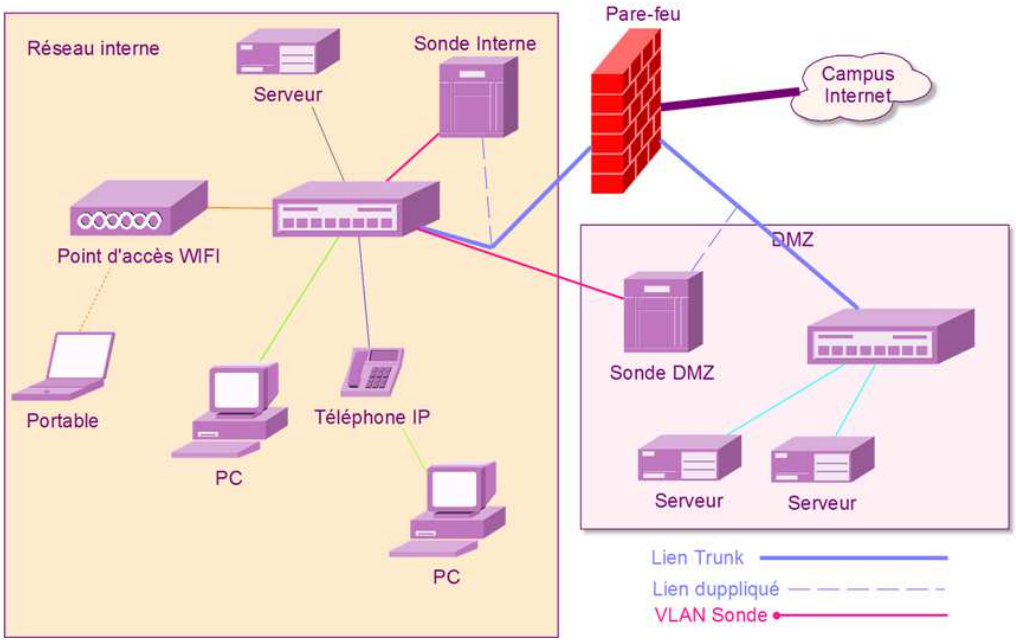

Figure 1 - Schéma de principe du réseau du CERMAV

Pour un premier essai, nous décidons de placer la sonde à l'interface entre notre pare-feu et le commutateur principal. En effet, sur ce lien sont visibles l'ensemble des VLANS utilisés au CERMAV. La sonde pourra donc analyser tout le trafic interne du laboratoire. Ensuite, nous déploierons une nouvelle sonde au niveau des DMZ.

[3] *Intrusion detection system*

3 Mise en œuvre

3.1 Matérielle

Pour nos deux sondes, nous décidons de réutiliser deux ordinateurs *rackable* qui été dédiés au filtrage anti-spam et équipés de processeurs Intel deux cœurs et de 4Go de RAM. Ces ordinateurs sont également pourvus de deux cartes réseaux chacun, ce qui est un minimum pour pouvoir faire fonctionner SURICATA. En effet, une carte sera dédiée à la supervision de la sonde et une autre à la réception du trafic réseau à analyser.

```
monitor session 1 source interface Gi3/0/22
monitor session 1 destination interface Gi3/0/17 encapsulation replicate
monitor session 2 source interface Gi1/0/26
monitor session 2 destination interface Gi3/0/4 encapsulation replicate
```

Figure 2 - Commandes sur un commutateur Cisco pour mettre en œuvre le mirroir de port

Au niveau des équipements réseau, il suffit de disposer de commutateurs compatibles avec le miroir de port. L'ensemble du trafic d'une interface du commutateur est alors dupliqué sur l'interface de supervision de la sonde, sans conséquences sur les performances du réseau du laboratoire y compris en cas de panne de cette dernière.

3.2 Logicielle

3.2.1 Installation

Le premier objectif étant l'installation du logiciel Suricata, nous préparons la machine avec une distribution Linux CentOS et procédons à l'installation et à la configuration de la sonde[4]. Rapidement, nous constatons qu'il serait intéressant de disposer d'une interface graphique afin d'afficher et de classer les messages d'alertes générés par la sonde.

```
11/04/2012-11:16:36.282266  [**] [1:2012648:3] ET POLICY Dropbox Client Broadcasting [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {UDP}
255.255.255.255:17500
11/04/2012-11:16:39.094685  [**] [1:2210039:1] SURICATA STREAM Last ACK with wrong seq [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.12:53516 -> 192.168
11/04/2012-11:16:39.102791  [**] [1:2210039:1] SURICATA STREAM Last ACK with wrong seq [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.12:53517 -> 192.168
11/04/2012-11:16:40.853592  [**] [1:2200074:1] SURICATA TCPv4 invalid checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.24:22 -> 192.168.1.54:50447
11/04/2012-11:16:41.522184  [**] [1:2200074:1] SURICATA TCPv4 invalid checksum [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.1.24:22 -> 192.168.1.54:50447
```

Figure 3 - Exemple de messages provenant de Suricata

Plusieurs solutions s'offrent à nous. La première consiste à rajouter sur la machine hébergeant la sonde un logiciel traitant les messages générés. Nous avons tout d'abord testé Kibana qui combiné avec Logstash et Elastic Search permet d'obtenir des graphes ou de manière plus générale de donner du sens aux montagnes de logs générés par Suricata.

[4] https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Installation

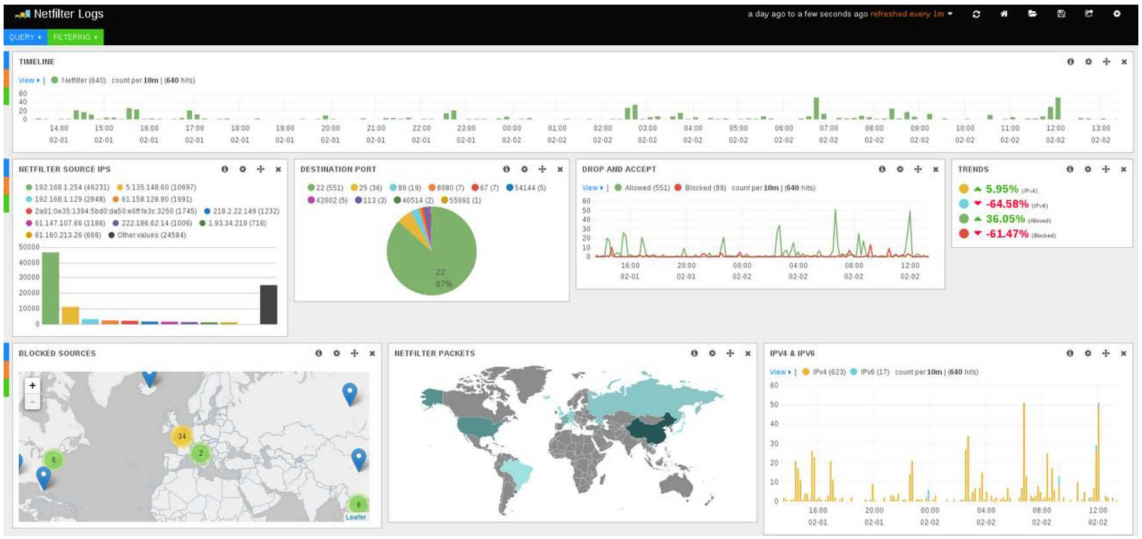

Figure 4 - Interface de gestion du logiciel Kibana

Au fur et à mesure de nos tests, nous avons découvert qu'il existait une distribution Linux Debian intégrant Suricata et des outils d'analyse. Il s'agit de SELKS[5] (Suricata, Elasticsearch, Logstash, Kibana, Scirius). Outre le fait de fournir une solution clef en main et immédiatement opérationnelle, cela permet aussi de s'affranchir des éventuels problèmes de librairies ou de compatibilité de versions logicielles que nous avions rencontrés sous CentOS mais aussi de faciliter le déploiement d'autres sondes.

Maintenant que nous avons vu à quel point il est simple de déployer des sondes, il nous est apparu nécessaire de pouvoir centraliser l'ensemble des informations collectées dans une seule et même interface.

3.2.2 Centralisation des données collectées

La centralisation est effectuée en utilisant le logiciel Snorby[6] qui permet l'affichage des résultats de logiciels de détection tels que Snort, Saga et bien entendu Suricata. Il s'agit d'un logiciel Open Source maintenu par la société Threat Stack reposant sur une base de données de type MySQL. Cette dernière est remplie au fur et à mesure par les sondes et l'interpréteur de log Barnyard2.

Figure 5 - Interface de gestion du logiciel Snorby

[5] http://linuxfr.org/news/selks-1-0-une-distribution
[6] https://github.com/Snorby/Snorby

Snorby offre une interface agréable et un classement des alertes par niveau de sévérité. Il est ensuite possible de classer chacune des alarmes dans 9 catégories (Faux positifs, virus, violation de politique de sécurité, déni de service, reconnaissance, tentative d'intrusion, accès non autorisé utilisateur, accès non autorisé administrateur & spécial). Autre élément intéressant, le moteur de recherche intégré, qui permet de retrouver un évènement à partir de mots clefs, d'adresses IP, de signatures…

4 Retour d'expérience

4.1 Constat

Après plusieurs mois d'utilisation sur le réseau du laboratoire (entre mars et septembre 2015 cela représente environ 320 000 incidents de niveau haut, 105 000 de niveau moyen et 220 000 de niveau bas), nous avons été en mesure de détecter plusieurs types de problèmes de sécurité qui seront détaillés dans la suite de ce document. Cependant, il est nécessaire de consacrer quotidiennement du temps pour procéder à l'analyse et au traitement des évènements générés qui sont de plusieurs milliers par jour. In fine, cela ne représente que quelques dizaine d'incidents nécessitant une intervention. Bien que l'application Snorby adresse par courriel un rapport journalier, il est difficile d'être très réactif face à une attaque.

4.2 Détections

Comme nous l'avons vu précédemment, Snorby regroupe par niveau de gravité les alertes générées par les sondes.

4.2.1 Incidents de niveau haut

Nous analysons plusieurs fois par jour ce type d'incident. Il a ainsi été possible de relever :

- Des ordinateurs de bureau compromis par des malware
- Des sites institutionnels utilisant un codage en base 64 pour la protection des identifiants
- Des mots de passes compromis lors d'authentification sur des sites internet ne proposant pas de connexion utilisant le protocole https (webmail, sites d'éditeurs, commerce en ligne…)
- Des tentatives d'intrusion sur notre serveur web
- …

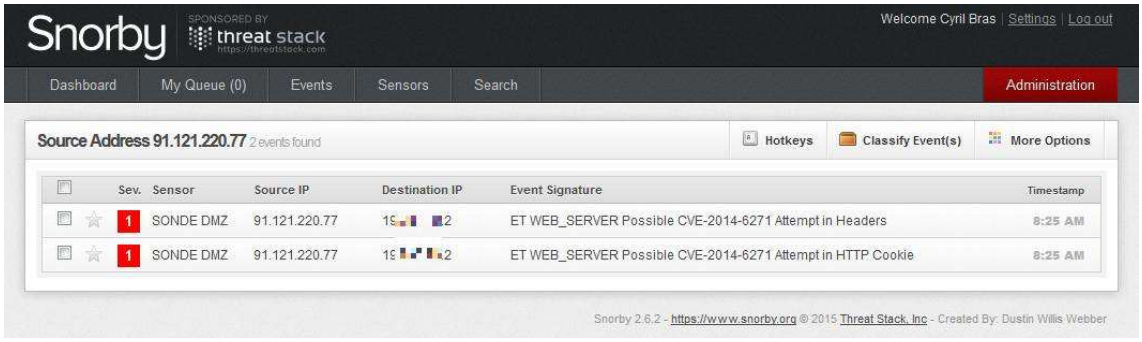

Figure 6 - Exemple de tentative d'attaque sur le serveur web du labo

Pour chacun des incidents détectés nous mettons en œuvre des solutions (analyse antivirale, modification des mots de passe, blocage d'adresses IP au niveau de notre pare-feu…)

4.2.2 Incidents de niveau moyen

Nous analysons ce type d'incident une fois par jour sauf si un pic est constaté au cours de la journée. Il a ainsi été possible de relever :

- Des tentatives de reconnaissances (connexion à des pages d'erreur sur nos sites web)
- Des versions de plug-in obsolètes (version de Java, Flashplayer…)
- Présence de logiciels de P2P
- …

4.2.3 Incidents de niveau bas

Nous analysons ce type d'incident une fois par semaine sauf si un pic est constaté au cours de la journée. Il a ainsi été possible de relever :

- Des requêtes web mal formées
- Des échecs de téléchargement
- Beaucoup de faux positifs

Des adaptations de filtres ont été nécessaires afin de limiter le nombre de faux positifs (par exemple : l'utilisation du logiciel de sauvegarde Bacula générait plusieurs milliers d'alertes quotidiennes).

4.3 Personnalisation

Afin d'éviter les faux positifs récurrents (par exemple : des alertes générées tous les jours par notre solution de sauvegarde) ou au contraire pour surveiller un flux particulier (par exemple : les connexions de nos serveurs en DMZ vers des sites web sur les ports tcp/80 et tcp/443), un paramétrage de certains fichiers est nécessaire.

- Le fichier /etc/suricata/suricata.yaml contient la déclaration des fichiers de règles utilisés. Nous allons rajouter une référence vers notre fichier de règles personnalisées.
- Le fichier /etc/suricata/classification.config qui contient les définitions de classes d'alerte.
- Le fichier /etc/suricata/rules/cermav.rules qui contient les définitions de nos règles. L'explicitation de la syntaxe du fichier se trouve sur le site Internet de openinfosecfoundation.org[7]

Figure 7 - Extrait des règles spécifiques pour le CERMAV

4.4 Perspectives

Suricata n'étant pas seulement un IDS mais intégrant aussi un IPS[8], cela laisse des perspectives intéressantes de protection pour nos serveurs. Une réaction automatique de protection en cas d'attaque par exemple mais que nous n'avons pas encore eut le temps de tester… Il pourrait être intéressant de déployer cela au niveau de nos serveurs hébergeant les sites Internet du laboratoire. En effet, ils sont soumis à des tentatives régulières d'intrusion que Suricata a permis de déceler plus facilement. Auparavant, ces tentatives n'étaient décelables qu'en réalisant une analyse approfondie des fichiers de journalisation.

[7] https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Rules
[8] *Intrusion Prevention System*, système de prévention d'intrusion

5 Conclusion

Depuis la mise en service de nos sondes au cours du mois de mars 2015, nous avons détecté de nombreux problèmes de sécurité sur nos réseaux. Ces problèmes n'auraient pas forcément été détectés sans la présence des sondes. Le bilan est donc très positif d'autant plus que l'analyse quotidienne des évènements et leur traitement n'entraine pas une surcharge considérable de travail.

Bibliographie

1. **Davadie, Philippe.** *L'entreprise - Nouveaux défis Cyber.* Paris : Economica, 2014.

2. **Di Benedetto, Louis.** *RSSIC-01.* Paris : CNRS, 2014.

3. *Détection d'intrusion dans les systèmes industriels. Suricata et le cas de Modbus.* **Mathieu Feuillet, David Diallo.** Rennes : s.n., 2014. C&esar 2014.

4. *Architecture système sécurisée de sondes IDS réseau.* **Pierre Chifflier, Arnaud Fontaine.** Rennes : s.n., 2014. C&esar 2014.

5. *Suricata | Open Source IDS / IPS / NSM engine.* [Online] 2015. http://suricata-ids.org/.

6. **Stamus Networks.** Open Source | Stamus Networks. [Online] 2015. https://www.stamus-networks.com/open-source/.