ENSTA Bretagne
2, rue François Verny
29806 BREST cedex
FRANCE
Tel +33 (0)2 98 34 88 00
www.ensta-bretagne.fr

Status Report

promo 2017

March 8, 2017

Sniff Hynesim

RIGAUD MICHAËL

Contents

Introduction

The cyber security is one of the major thread of the 21th century, and attackers use techniques more and more sophisticated. So one of the most important aim for cyber security engineer is to find a way to detect and stop attacks. To do that effectively cyber engineer need to analyze cyber attack to find a way to detect them. A solution is use simulators of network and information system to reproduce as much as they want, without injury, and huge agility scenario of cyber attack. To do that, the ENSTA Bretagne has decided as many company like Thales or the DGA to use Hynesim[1].

The aim of this project is elaborate a solution to alert when a strategy of attack is spotted out. To do that, we have to create pattern of attack and use an IDS[2] to alert us. To verify the solution and create pattern as exhaustive as possible, we will use Hynesim.

To begin, we will present Hynesim and the advantages of this software. Then, we are going to present the aim of an IDS and the most popular IDS. And to finish we will present the aim of this project and the organization of the project.

[1] This software is presented in the chapter 1

[2] Intrusion detection system

Part I

Bibliographic study

CHAPTER 1

Hynesim

Firstly, I will present Hynesim. In fact, I will use this tool to create our network and test our solution so it is important to introduce it.

1.1 Presentation

Figure 1.1: Hynesim logo

Définition 1.1 : *Hynesim*
Means HYbrid NEtwork SIMulation, is a distribution platform of simulation of information systems developed by Diateam. [3]

The platform was initially developed by Diateam for DGA[1] MI (Maitrise de l'information) to create virtual networks. But now is a major project to develop information systems and automatize cyber security attacks. This project has two version, an open source version and a professional version. The open source version has less options, but I will use this version for this project.

1.2 Architecture

In order to work, Hynesim needs a server with on it the main software. This software is the virtualization part. It manages virtual machines and networks.

Moreover, to see virtual machine and interact with them, users needed to have a client interface. This interface can be installed on a simple computer.

To add a virtual machine to Hynesim, I need to create it on Virtual Box[2] or VMWare[3] and then import them on Hynesim.

[1] French Procurement Agency

[2] More information here: `https://www.virtualbox.org/`

[3] More information here: `https://www.vmware.com/`

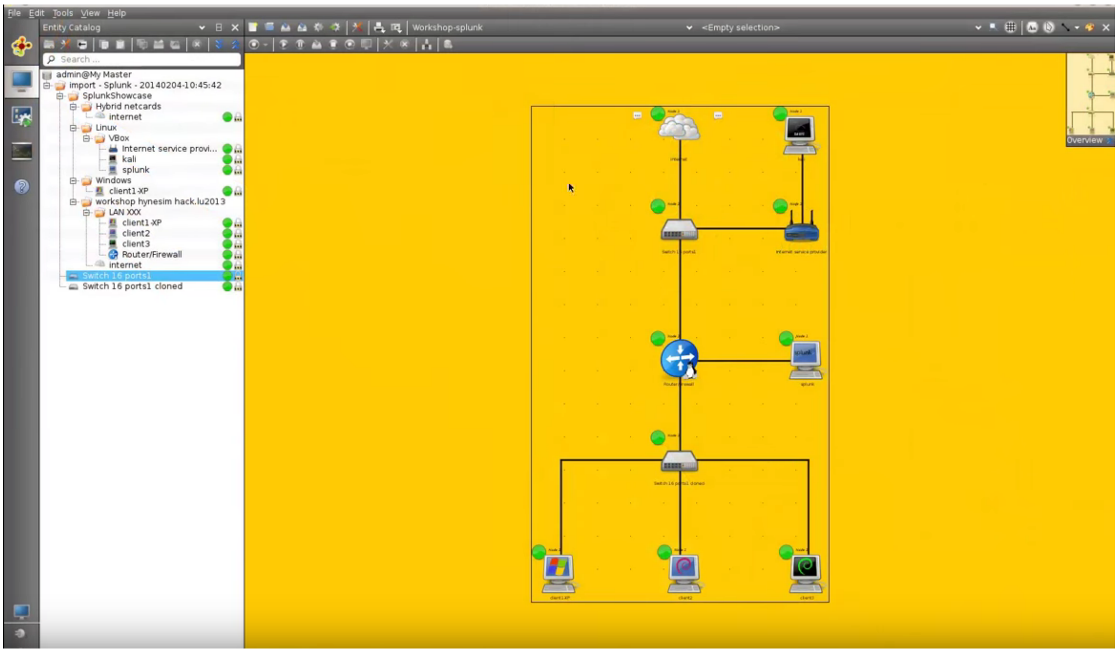


Figure 1.2: An example of a client Hynesim interface

CHAPTER 2

IDS

In this subject, a probe will be created to detect attacks. It is the aim of IDS that I are going to introduce.

2.1 Presentation

Définition 2.1 : *IDS*
An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. [2]

There are many type of IDS:

- NIDS, network IDS. They listen and analyze the network and detect attacks from network packets. They are the most interesting for our subject, so this document will focus primarily on this type of IDS.
- HIDS, host IDS. These IDS are on a system and they detect intrusion within it.
- Hybrid IDS. They are composed of NIDS and HIDS.
- IPS. They are NIDS with active functions which help to stop attackers.
- KIDS/KIPS, kernel IDS. They are types of HIDS. They are on the system kernel. They are more effective and slower than HIDS.

In the following document I will talk about NIDS. But to detect attacks there is also many methods.

To be efficient IDS should have a good balance between some features.

- **Speed:** In fact an IDS should analyze packets as fast as possible, otherwise, it will behind the network traffic.
- **False alarm:** An IDS raises an alert when it detect attacks. But it could raise alarm during a normal utilization, a false alarm. It is one of the most important features, because at every alarm a system administrator needs to analyze the alert. So in company, every alarm cost time and money.
- **Probability of detection:** For an IDS, the capacity of detecting attacks. The higher the probability the more the IDS will raise false alarms. In some sensitive systems, I prefer to detect every attack and raise many false alarms. That depends on the system.

2.2 Detection methods

2.2.1 Misuse detection

This technique is the simplest. It uses attack signature to raise alerts. In fact, all attacks have a particularity, if I detect this particularity I can detect the attackers. There are three sub methods.

Pattern matching

In this technique I have a base of signature and the IDS looks for the pattern. If the pattern matches perfectly, this IDS raises an alert.

The problem is, only attacks which are in our base can be detected. So if there is a new attack (zero day), or if the attack is not perfectly the same, it cannot be detected.

However, this method is much used because it is high-performance, and with this method the IDS don't raise a lot of false alerts

Dynamic pattern matching

In this techniques the IDS is also based on signature but this data base is dynamic. In other words, the IDS has the faculty of adaptation and learning. The IDS improve its data base of signature automatically.

Protocol analysis

The last sub method I will present is the protocol analysis. This technique is based on the verification of protocol. The IDS will check if flows are compliant with RFC[1] standards. It will verify parameters of packets and fields. An IDS can check many protocols such as FTP, HTTP, ICMP, . . .

The advantages of this method is that I can detect unknown attacks contrary to pattern matching. However, software publishers don't often respect RFC so this technique is not always very efficient.

2.2.2 Anomaly detection

This technique consists in detecting an intrusion through the analysis of the user's past behavior. So the IDS should create a profile of users from his use and raises alerts when there is an event outside this profile. To create a profile the IDS could use machine learning.

Advantages	Drawbacks
The IDS should be able to detect every type of attack even unknown (zero days) attacks.	This method is not reliable. Every alteration of the use creates an alert
The IDS is autonomous	This method needs a learning period. In fact, this method needs to learn the habit of users, so I need a period without attacks
	Hackers can need only time. In fact, if the attacker creates a new profile after many month with his attacks, he could attack silently

[1] Requests for Comments, is a type of publication from the Internet Engineering Task Force (IETF) and the Internet Society (ISOC), the principal technical development and standards-setting bodies for the Internet.

Probabilistic method

Bayésien network is a learning machine based on probability. The IDS will create a probabilistic model and will raise an alert if the user don't respect this model.

For example, I know that in 90% of cases in an HTTP request the first parameter is GET after a connection to the port 80.

2.3 Main IDS

There are many IDS on the market. The most popular open source solutions are Snort, Suricata, Bro, Fail2Ban, ACARM ... There are also some tools «all in one». It is usually OS[2] with an IDS, a tool to analyze alert, a tool to create rules,... The most popular are SELKS and ELKS. A description of SELKS is available page 26.

[2] Operating system

CHAPTER 3

SIEM

After some new discussion with my tutor, we decide to had a new component in our architecture. So we will add a SIEM to analyze all alert and take care of server's log. So we will add in this status report a part about SIEM which was not initially.

3.1 Description

Définition 3.1 : *SIEM*

In the field of computer security, security information and event management (SIEM) software products and services combine security information management (SIM) and security event management (SEM). They provide real-time analysis of security alerts generated by network hardware and applications.

Part II

Presentation of the subject

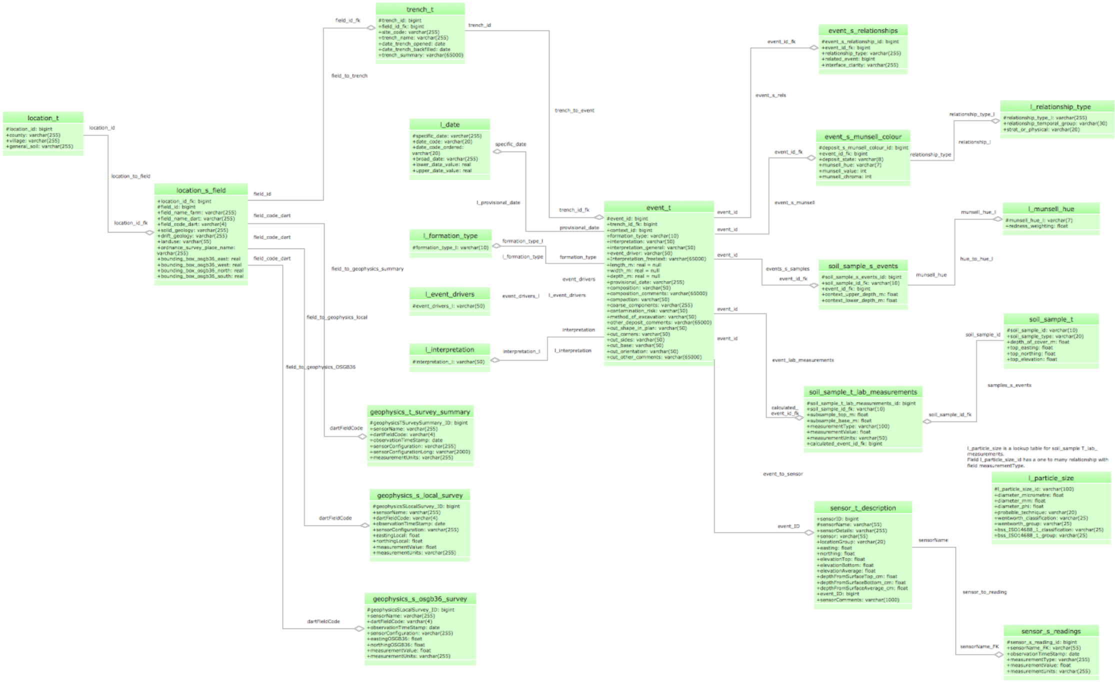
CHAPTER **4**

Presentation of the subject

After this bibliographic study, I will present the subject of my work and the aim of it. My work supposed that a model checking was done, so I will firstly explain that.

4.1 Model checking

Since few years, we are able to realize model to represent systems. These models permit to simulate systems. To do these models, there is many languages UML, SysML, Fiacre,. . . The figure 4.1 represents an example of a complex system.



trench_t
#trench_id: bigint
+field_id_fk: bigint
+site_code: varchar(255)
+trench_name: varchar(255)
+date_trench_opened: date
+date_trench_backfilled: date
+trench_summary: varchar(65000)
field_id_fk
trench_id
field_to_trench
location_t
#location_id: bigint
+county: varchar(255)
+village: varchar(255)
+general_soil: varchar(255)
location_id
location_to_field
location_id_fk
location_s_field
+location_id_fk: bigint
#field_id: bigint
+field_name_farm: varchar(255)
+field_name_dart: varchar(255)
+field_code_dart: varchar(4)
+solid_geology: varchar(255)
+drift_geology: varchar(255)
+landuse: varchar(55)
+ordnance_survey_place_name: varchar(255)
+bounding_box_osgb36_east: real
+bounding_box_osgb36_west: real
+bounding_box_osgb36_north: real
+bounding_box_osgb36_south: real
field_id
field_code_dart
field_code_dart
field_code_dart
field_to_geophysics_summary
field_to_geophysics_local
field_to_geophysics_OSGB36
l_date
#specific_date: varchar(255)
+date_code: varchar(20)
+date_code_ordered: varchar(20)
+broad_date: varchar(255)
+lower_date_value: real
+upper_date_value: real
specific_date
l_provisional_date
l_formation_type
#formation_type_l: varchar(10)
formation_type_l
l_formation_type
l_event_drivers
#event_drivers_l: varchar(50)
event_drivers_l
l_event_drivers
l_interpretation
#interpretation_l: varchar(50)
interpretation_l
l_interpretation
trench_to_event
trench_id_fk
provisional_date
formation_type
event_drivers
interpretation
event_t
#event_id: bigint
+trench_id_fk: bigint
+context_id: bigint
+formation_type: varchar(10)
+interpretation: varchar(50)
+interpretation_general: varchar(50)
+event_driver: varchar(50)
+interpretation_freetext: varchar(65000)
+length_m: real = null
+width_m: real = null
+depth_m: real = null
+provisional_date: varchar(255)
+composition: varchar(50)
+composition_comments: varchar(65000)
+compaction: varchar(50)
+coarse_components: varchar(255)
+contamination_risk: varchar(50)
+method_of_excavation: varchar(50)
+other_deposit_comments: varchar(65000)
+cut_shape_in_plan: varchar(50)
+cut_corners: varchar(50)
+cut_sides: varchar(50)
+cut_base: varchar(50)
+cut_orientation: varchar(50)
+cut_other_comments: varchar(65000)
event_id
event_id
event_id
event_id
event_id
event_s_rels
event_id_fk
event_s_relationships
#event_s_relationship_id: bigint
+event_id_fk: bigint
+relationship_type: varchar(255)
+related_event: bigint
+interface_clarity: varchar(255)
event_id_fk
event_s_munsell_colour
#deposit_s_munsell_colour_id: bigint
+event_id_fk: bigint
+deposit_state: varchar(8)
+munsell_hue: varchar(7)
+munsell_value: int
+munsell_chroma: int
relationship_type
relationship_type_l
relationship_l
l_relationship_type
#relationship_type_l: varchar(255)
+relationship_temporal_group: varchar(30)
+strat_or_physical: varchar(20)
event_s_munsell
munsell_hue_l
l_munsell_hue
#munsell_hue_l: varchar(7)
+redness_weighting: float
hue_to_hue_l
munsell_hue
events_s_samples
event_id_fk
soil_sample_s_events
#soil_sample_s_events_id: bigint
+soil_sample_id_fk: varchar(10)
+event_id_fk: bigint
+context_upper_depth_m: float
+context_lower_depth_m: float
soil_sample_id
soil_sample_t
#soil_sample_id: varchar(10)
+soil_sample_type: varchar(20)
+depth_of_cover_m: float
+top_easting: float
+top_northing: float
+top_elevation: float
samples_s_events
event_lab_measurements
calculated_event_id_fk
soil_sample_t_lab_measurements
#soil_sample_t_lab_measurements_id: bigint
+soil_sample_id_fk: varchar(10)
+subsample_top_m: float
+subsample_base_m: float
+measurementType: varchar(100)
+measurementValue: float
+measurementUnits: varchar(50)
+calculated_event_id_fk: bigint
soil_sample_id_fk
l_particle_size is a lookup table for soil_sample T_lab_ measurements.
Field l_particle_size_id has a one to many relationship with field measurementType.
l_particle_size
#l_particle_size_id: varchar(100)
+diameter_micrometre: float
+diameter_mm: float
+diameter_phi: float
+probable_technique: varchar(20)
+wentworth_classification: varchar(25)
+wentworth_group: varchar(25)
+bss_ISO14688_1_classification: varchar(25)
+bss_ISO14688_1_group: varchar(25)
event_to_sensor
event_ID
sensor_t_description
+sensorID: bigint
#sensorName: varchar(55)
+sensorDetails: varchar(255)
+sensor: varchar(55)
+locationGroup: varchar(20)
+easting: float
+northing: float
+elevationTop: float
+elevationBottom: float
+elevationAverage: float
+depthFromSurfaceTop_cm: float
+depthFromSurfaceBottom_cm: float
+depthFromSurfaceAverage_cm: float
+event_ID: bigint
+sensorComments: varchar(1000)
sensorName
sensor_to_reading
sensorName_FK
sensor_s_readings
#sensor_s_reading_id: bigint
+sensorName_FK: varchar(55)
+observationTimeStamp: date
+measurementType: varchar(255)
+measurementValue: float
+measurementUnits: varchar(255)
dartFieldCode
geophysics_t_survey_summary
#geophysicsTSurveySummary_ID: bigint
+sensorName: varchar(255)
+dartFieldCode: varchar(4)
+observationTimeStamp: date
+sensorConfiguration: varchar(255)
+sensorConfigurationLong: varchar(2000)
+measurementUnits: varchar(255)
dartFieldCode
geophysics_s_local_survey
#geophysicsSLocalSurvey_ID: bigint
+sensorName: varchar(255)
+dartFieldCode: varchar(4)
+observationTimeStamp: date
+sensorConfiguration: varchar(255)
+eastingLocal: float
+northingLocal: float
+measurementValue: float
+measurementUnits: varchar(255)
dartFieldCode
geophysics_s_osgb36_survey
#geophysicsSLocalSurvey_ID: bigint
+sensorName: varchar(255)
+dartFieldCode: varchar(4)
+observationTimeStamp: date
+sensorConfiguration: varchar(255)
+eastingOSGB36: float
+northingOSGB36: float
+measurementValue: float
+measurementUnits: varchar(255)

Figure 4.1: Complex model in UML

Then with the simulation of these systems, we can imagine that we are able to find all possible states of our system. So we know that if we are not in a valid state of the system, the system was potentially hacked.

4.2 Position of my project

So for my project, I guess I have all possible state for my system and I will check these possible states with the actual state of the system. To do so, I have to find the state of applications and state of the network, and analyze them.

I choose to use an IDS to have the status of the network. In fact, an IDS can raise alert when there is particular message, so I will raise event when I sniff particular message. To have a status of an application, I choose to implement probe in applications, or if it is possible use the log of these applications. Then, the SIEM will correlate all these messages and alerts to find the status of our system and it will compare with all previous possibility.

The figure 4.2 represent the position of our project.

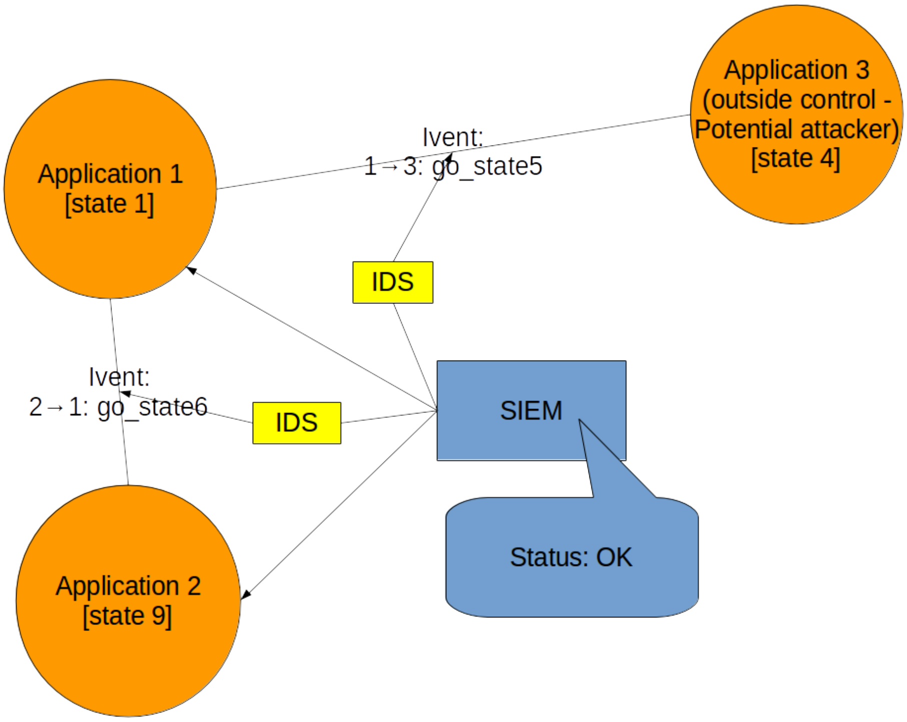


Figure 4.2: Position of my project

CHAPTER 5

Technical choice

5.1 Application created for test

As it was explain before, I have to protect a specific application. To do my tests, and prove the concept of my work, I choose to create a simple application. I plan to add some security issue to be more realistic.

This application have three states, and two part. Firstly, there is an admin interface which contain the state admin. This interface is connected on the port 9000. Only one specific known computer can access to this interface. Of course, the aim of attackers is to arrive in this state.

Then there is the main application. It is connected to the port 9124, and everybody can connect to it. There is two state ping and pong. If the application is in the state ping and the user send «topong» the application go to state pong, and if the application is in the state pong and the user send «toping» the application go to the state ping. There is also an unknown backdoor, if the user is in the state ping for the third time and he send «admin», the application go to the admin interface.

The figure 5.1 represent a model of our application.

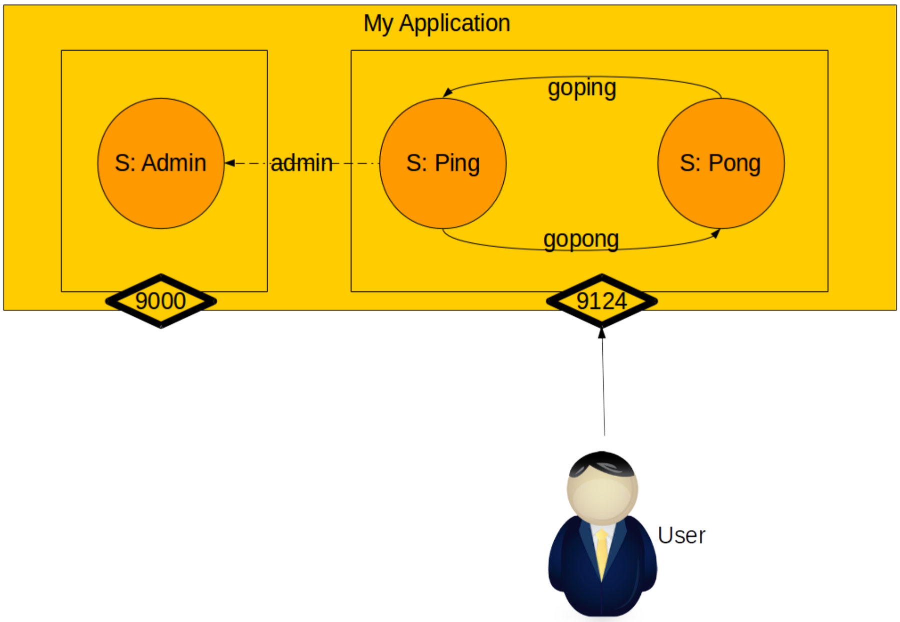


Figure 5.1: Model of the application

5.2 IDS

Choice of SELKS

5.3 SIEM

Implement my SIEM

CHAPTER 6

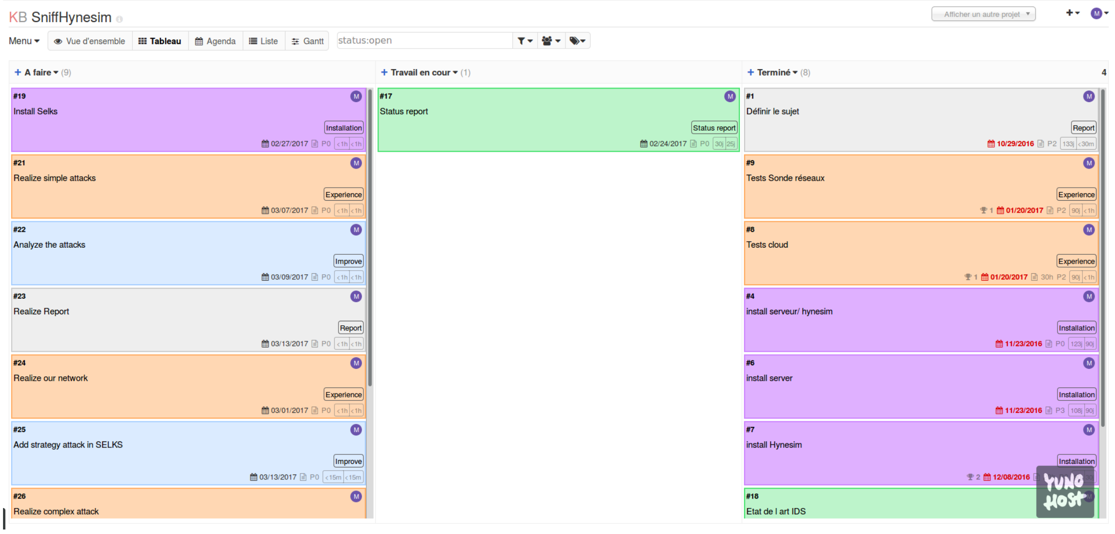
Forecasting organization

6.1 Kanban

To achieve this project, we decided to use some tools to arrange our work. First of all, we decided to use an agile technique of management which name Kanban.

Définition 6.1 : *Kanban*
Kanban is a new technique for managing a software development process in a highly efficient way. Kanban underpins Toyota's "just-in-time" (JIT) production system. The kanban system consists of a big board on the wall with cards or sticky notes placed in columns with numbers at the top [9]

Kanban is an inventory-control system to control the supply chain. It uses a board with columns. Each column represents a status, for example: to do, doing, done. In each column we put «notes» which represent a task. Moreover, each column has a maximum number of notes authorized.



A faire (9)

#19 Install Selks — Installation — 02/27/2017 P0 <1h <1h

#21 Realize simple attacks — Experience — 03/07/2017 P0 <1h <1h

#22 Analyze the attacks — Improve — 03/09/2017 P0 <1h <1h

#23 Realize Report — Report — 03/13/2017 P0 <1h <1h

#24 Realize our network — Experience — 03/01/2017 P0 <1h <1h

#25 Add strategy attack in SELKS — Improve — 03/13/2017 P0 <15m <15m

#26 Realize complex attack

Travail en cour (1)

#17 Status report — Status report — 02/24/2017 P0 30j 25j

Terminé (8) 4

#1 Définir le sujet — Report — 10/29/2016 P2 133j <30m

#9 Tests Sonde réseaux — Experience — 1 01/20/2017 P2 90j <1h

#8 Tests cloud — Experience — 1 01/20/2017 30h P2 90j <1h

#4 install serveur/ hynesim — Installation — 11/23/2016 P0 123j 90j

#6 install server — Installation — 11/23/2016 P3 108j 90j

#7 install Hynesim — Installation — 2 12/08/2016

#18 Etat de l art IDS

Figure 6.1: Kanboard of this project

Limiting the amount of tasks, at each step in the process, prevents overproduction and reveals bottlenecks dynamically. In fact, with this technique it is possible to have a better overview of the project and control it dynamically.

For this project, we use Kanboard[5] self-hosted on our own Yunohost server.[1]. You can see in the figure 6.1 the kanboard of this project. Each color represent a category of tasks: blue: improvement , purple: installation, red : experience, green: status report, and grey: report.

With this tool, it is also possible to see tasks as a Gantt diagram. The figure 6.2 presents our Gantt diagram.

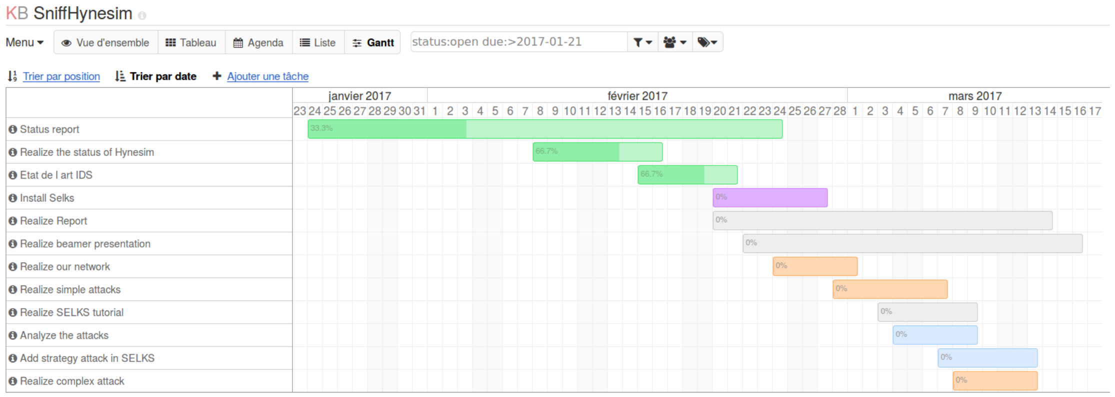


Figure 6.2: Gantt diagram of the project

6.2 Github

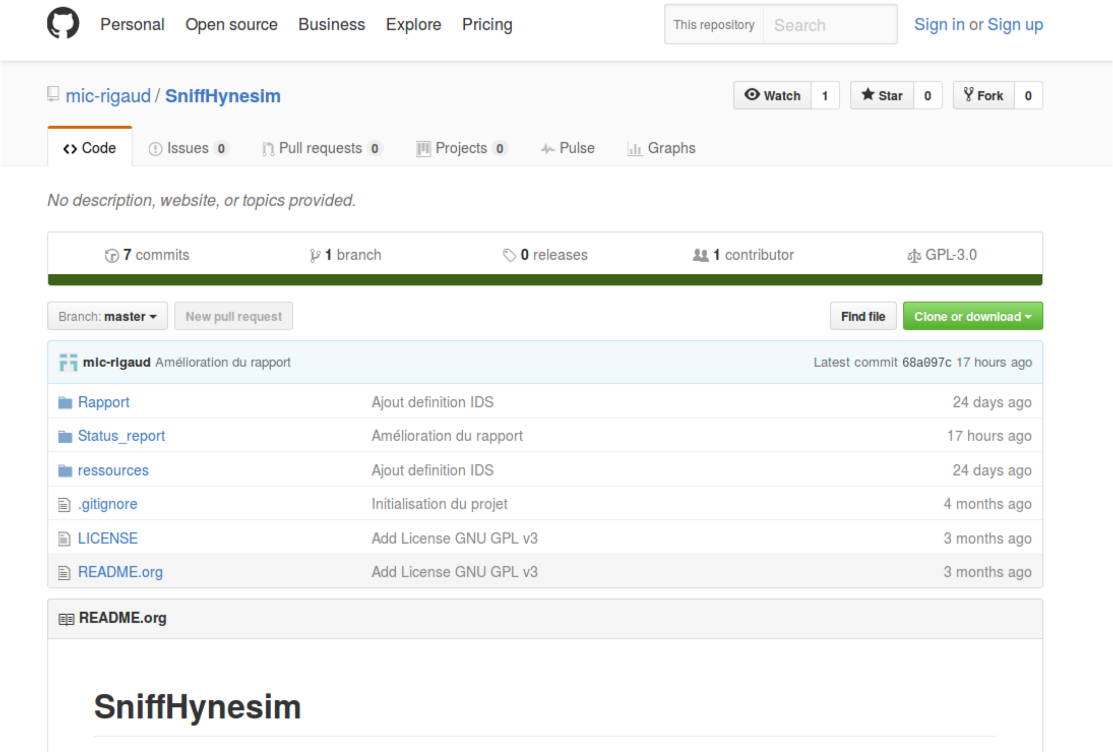
To achieve this project, we also decided to use Git and Github as a Git server.

Définition 6.2 : *Git*

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people.

Git enables me to control version of our work and obtain a real showcase of it for our tutor. The figure 6.3 is a screenshot of our Github server.

[1] It is possible to see our kanboard at this link: `https://mic-rigaud.fr/kanboard/?controller=BoardViewController&action=readonly&token=10ea65eca908023dbcd8bc8dce75791c7a14d67912627dafaa5b71033222`



mic-rigaud / SniffHynesim

No description, website, or topics provided.

7 commits | 1 branch | 0 releases | 1 contributor | GPL-3.0

mic-rigaud Amélioration du rapport — Latest commit 68a097c 17 hours ago

Rapport	Ajout definition IDS	24 days ago
Status_report	Amélioration du rapport	17 hours ago
ressources	Ajout definition IDS	24 days ago
.gitignore	Initialisation du projet	4 months ago
LICENSE	Add License GNU GPL v3	3 months ago
README.org	Add License GNU GPL v3	3 months ago

README.org

SniffHynesim

Figure 6.3: Github server

Part III

Technical study

Chapter 7

Hynesim installation

7.1 Hynesim

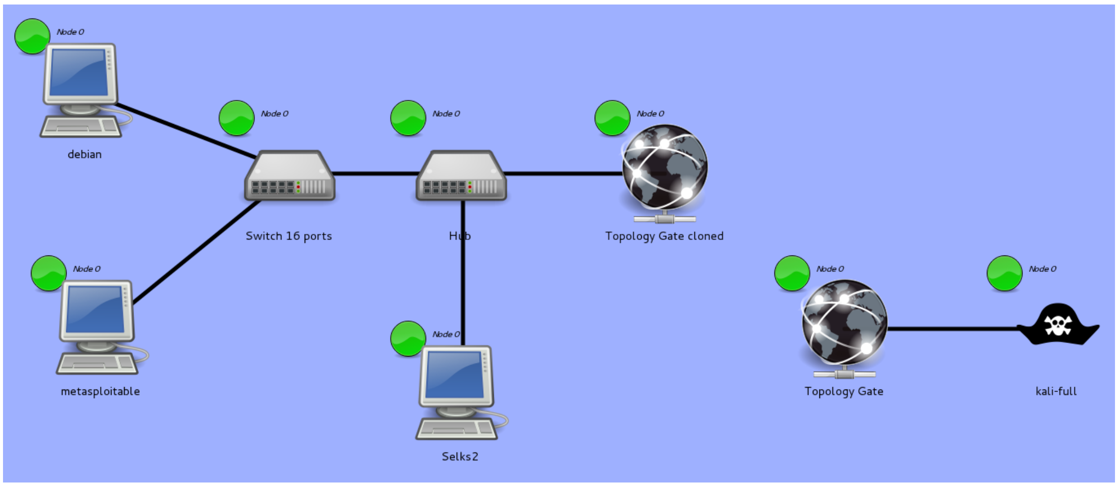
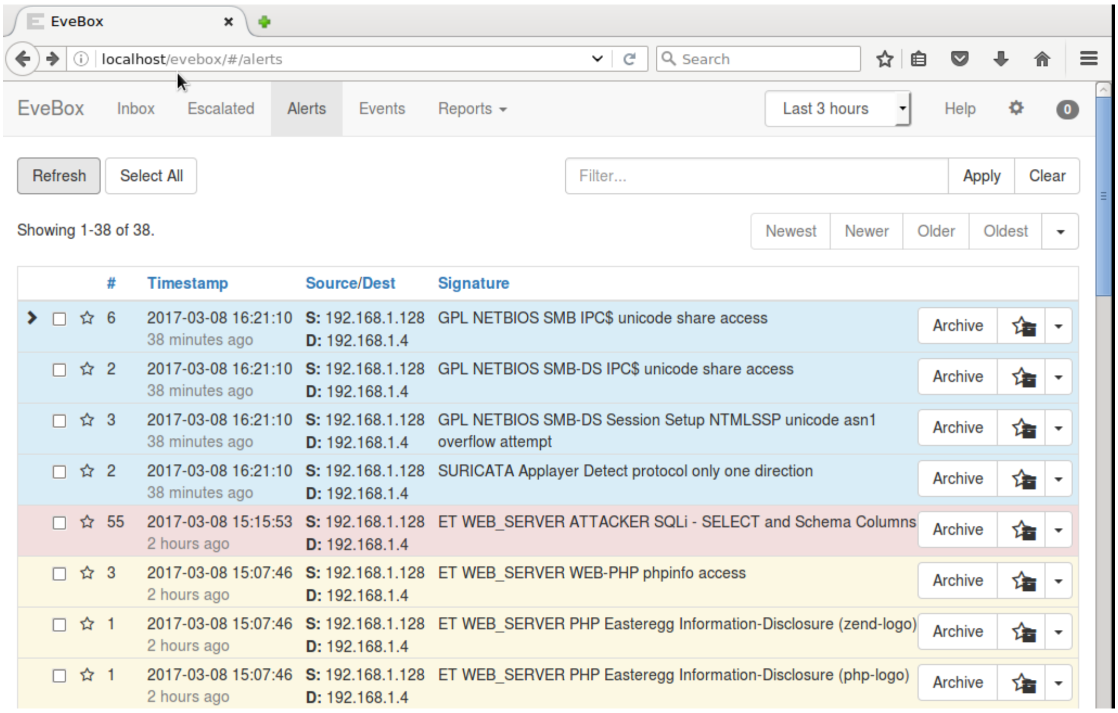


Figure 7.1: Network infrastructure

7.2 Selks



Showing 1-38 of 38.

		#	Timestamp	Source/Dest	Signature	
›	□ ☆	6	2017-03-08 16:21:10 38 minutes ago	**S:** 192.168.1.128 **D:** 192.168.1.4	GPL NETBIOS SMB IPC$ unicode share access	Archive
	□ ☆	2	2017-03-08 16:21:10 38 minutes ago	**S:** 192.168.1.128 **D:** 192.168.1.4	GPL NETBIOS SMB-DS IPC$ unicode share access	Archive
	□ ☆	3	2017-03-08 16:21:10 38 minutes ago	**S:** 192.168.1.128 **D:** 192.168.1.4	GPL NETBIOS SMB-DS Session Setup NTMLSSP unicode asn1 overflow attempt	Archive
	□ ☆	2	2017-03-08 16:21:10 38 minutes ago	**S:** 192.168.1.128 **D:** 192.168.1.4	SURICATA Applayer Detect protocol only one direction	Archive
	□ ☆	55	2017-03-08 15:15:53 2 hours ago	**S:** 192.168.1.128 **D:** 192.168.1.4	ET WEB_SERVER ATTACKER SQLi - SELECT and Schema Columns	Archive
	□ ☆	3	2017-03-08 15:07:46 2 hours ago	**S:** 192.168.1.128 **D:** 192.168.1.4	ET WEB_SERVER WEB-PHP phpinfo access	Archive
	□ ☆	1	2017-03-08 15:07:46 2 hours ago	**S:** 192.168.1.128 **D:** 192.168.1.4	ET WEB_SERVER PHP Easteregg Information-Disclosure (zend-logo)	Archive
	□ ☆	1	2017-03-08 15:07:46 2 hours ago	**S:** 192.168.1.128 **D:** 192.168.1.4	ET WEB_SERVER PHP Easteregg Information-Disclosure (php-logo)	Archive

Figure 7.2: Example of alerts

CHAPTER 8

Implementation

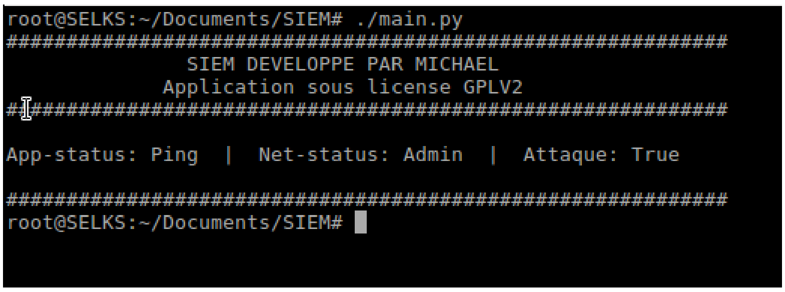
8.1 Application

8.2 Probe

8.2.1 Network probe

8.2.2 Application probe

8.3 SIEM



```
root@SELKS:~/Documents/SIEM# ./main.py
############################################################
               SIEM DEVELOPPE PAR MICHAEL
             Application sous license GPLV2
############################################################

App-status: Ping  |  Net-status: Admin  |  Attaque: True

############################################################
root@SELKS:~/Documents/SIEM# 
```

Figure 8.1: My siem interface

CHAPTER 9

Analysis of results

9.1 Results

9.2 Way of improve

Conclusion

After this bibliographic study, to implement our network sniffer which detect strategy of attack we have to improve an IDS. In fact, IDS have the ability to detect attack with many methods. Here, we have to use an anomaly detection method to find attackers. However, as we see, this method has many disadvantages so we have to improve it.

Moreover, as it was advise by the subject we will use Hynesim. In fact, Hynesim is a very interesting tool to virtualize and simulate network. By this way, we could test under many attacks our solution to secure network.

After this study, we have only a partial view of how implement this detection method. So one of the most important work for the next two weeks will be find the way to detect strategy of attacks. It is a difficult objective but also one of the most interesting for the security of our networks.

Annexe

Appendix A

SELKS

Définition A.1 : *SELKS*

SELKS is a free and open source Debian (with LXDE X-window manager) based IDS/IPS platform released under GPLv3 from Stamus Networks (https://www.stamus-networks.com/).[10]

The SELKS ISO is both Live and Installable ISO in one. Once installed it is ready to use out of the box solution.

SELKS is comprised of the following major components:

S Suricata IDPS - http://suricata-ids.org/

E Elasticsearch - http://www.elasticsearch.org/overview/

L Logstash - http://www.elasticsearch.org/overview/

K Kibana - http://www.elasticsearch.org/overview/

S Scirius - https://github.com/StamusNetworks/scirius

So SELKS is an OS which contains many softwares. Firstly there is Suricata which is the network analyzer which raise alerts. Alerts produce by Suricata are written in a EVE file. Then Logstash traduce this file into a JSON file readable by Elasticsearch. Elasticsearch is a data base. It keep all event and permit a navigation into many millions of events almost instantaneous. And to finish Kibana propose a User interface for this data base. The figure A.2 represents this architecture.[1]

You can see on figure A.3, an example of the interface of kibana which give an overview of alerts. This image was extract from the Twitter of Stamus Networks.

Figure A.1: screenshot of SELKS desktop

[1] SIEM (security information and event management) which are not developed here, are software to analyze and display event of every security tools.

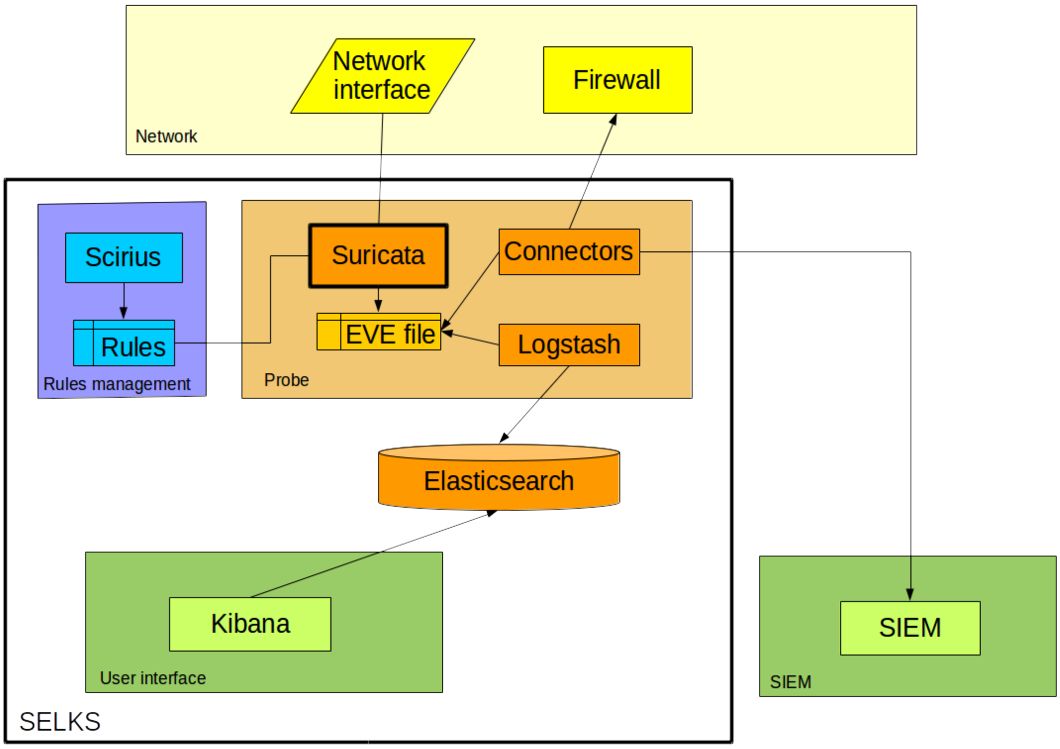
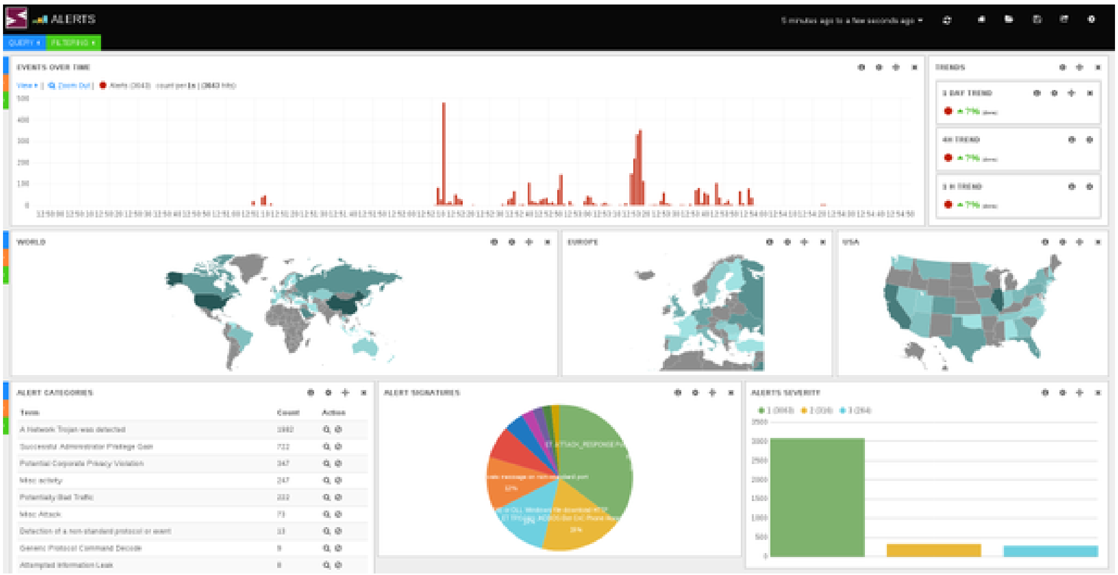


Figure A.2: Selks architecture



ALERT CATEGORIES

Term	Count	Action
A Network Trojan was detected	1982	
Successful Administrator Privilege Gain	722	
Potential Corporate Privacy Violation	347	
Misc activity	247	
Potentially Bad Traffic	222	
Misc Attack	73	
Detection of a non-standard protocol or event	13	
Generic Protocol Command Decode	9	
Attempted Information Leak	8	

Figure A.3: Kibana interface

List of Figures

Bibliography

[1] 3ilson.org. Selks + esxi installation guide. Youtube, October 2016.

[2] Vangie Beal. intrusion detection system. Webopedia.

[3] Diateam. Hynesim. https://www.hynesim.org/.

[4] Solange Ghernaouti. *Sécurité informatique et réseaux*, volume 4, chapter 8.4. Dunod, 2013. At Ensta Bretagne code: I11 GHE.

[5] Frédéric Guillot. Kanboard. https://kanboard.net/.

[6] Jonathan Krier. Les systèmes de détection d'intrusions. Technical report, developpez.com, july 2006.

[7] Eric Leblond. Let's talk about selks. In *SSTIC conference*, June 2014.

[8] Eric Leblond. Suricata, dévoilez la face sécurité de votre réseau. *Gnu Linux Magazine France Hors-série*, 76:9, 2015.

[9] David Peterson. What is kanban. Technical report, KanbanBlog, 2009.

[10] StamusNetworks. Selks. `https://github.com/StamusNetworks/SELKS`, 2014.