Smart Intrusion Detection Model for the Cloud Computing

Mostapha Derfouf, Mohsine Eleuldj, Saad Enniari and Ouafaa Diouri

Abstract Nowadays, Cloud computing is turning into a major trend in the field of computer science. It is referred to as a new data hosting technology that became very popular lately thanks to the repayment of costs induced to companies. However, and since this concept is still in its first stages of use, many new security risks start to appear, making its huge benefits fade compared to the security risks it brings with it. This paper proposes a smart intrusion detection model that is based on the principle of collaboration between many IDSs (Intrusion detection systems). These IDSs are host based intrusion detection systems (HIDS) that are customized in order to support data mining and machine learning and we call them SHIDS (Smart host based intrusion detection systems). The SHIDS are deployed on the different virtual machines in the cloud to detect and protect against attacks targeting applications running on virtual machines by exchanging encrypted IDMEF alerts. This approach provides many benefits in terms of portability and costs. That being said, this paper will set forth the different architectures of intrusion detection systems in the cloud, provide a comparison between the major architectures, propose our intrusion detection architecture and validate it with an experiment using Open stack.

Keywords IDS · HIDS · NIDS · Cloud computing · Machine learning · Open stack

M. Derfouf (✉) · M. Eleuldj · S. Enniari · O. Diouri
Department of Computer Science, Mohammadia School of Engineers,
Mohammed V University in Rabat, Rabat, Morocco
e-mail: mostaphaderfouf@research.emi.ac.ma

M. Eleuldj
e-mail: saadenniari@research.emi.ac.ma

S. Enniari
e-mail: eleuldj@emi.ac.ma

O. Diouri
e-mail: diouri@emi.ac.ma

Á. Rocha et al. (eds.), *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Advances in Intelligent Systems and Computing 520, DOI 10.1007/978-3-319-46568-5_42

1 Introduction

Cloud computing is a new data hosting technology that stands today as a satisfactory answer to the problem of data storage and computing. It can provide processing and accommodation of digital information via a fully outsourced infrastructure allowing users to benefit from many online services without having to worry about the technical aspects of their uses. Cloud computing appears as a tremendous opportunity for companies but logically raises the question of the security of data when they are hosted by a third party.

We have already dealt with the problem of data storage security on Cloud computing and proposed a solution based on encryption to secure data [1], in this paper, we propose a smart intrusion detection model designed to secure the cloud. First we will give an overview about the different intrusion detection models in the cloud environments then we provide a comparison between the different IDS models finally we propose our own intrusion detection model for the Cloud and provide the experiment.

2 Related Work

In the “Advanced IDS Management Architecture” [2] the authors proposed an IDS which uses an Event Gatherer combined with the Virtual Machine Monitor (VMM). This IDS is composed of many sensors and a central management unit. The Event Gatherer plugin plays the role of Handler, sender, and receiver in order to provide an integration of different sensors. This architecture uses the IDMEF (Intrusion Detection Message Exchange Format). An interface is designed to expose the result reports for users.

The multilevel IDS concept is proposed by Kuzhalisai and Gayathri [3] which deals with effective use of system of resources. The proposed system binds user in different security groups based on degree of anomaly called anomaly level. It consists of AAA module which is responsible for authentication, authorization and accounting. When user tries to access the cloud the AAA checks the authentication of the user and based on it, it gets the recently updated anomaly level. Security is divided into three levels: high, medium and low.

Gul and Hussain [4] have proposed a multi-threaded NIDS designed to work in distributed cloud environment. This multi-threaded NIDS contains three modules: capture and queuing module, analysis/processing module and reporting module. The capture module is responsible of reading the network packets and sending them to the shared queue for analysis.

In [5] the authors proposed a framework that integrates a network intrusion detection system (NIDS) in the Cloud. The proposed NIDS module consists of Snort and signature apriori algorithm that is capable of generating new rules from captured packets. The new rules are added to the Snort configuration file.

Table 1 Synthesis of the related work

IDS/Features	Advanced IDS management architecture	Cloud intrusion detection system	Cloud detection and prevention system	Improved hybrid IDS
Type	Collaborative	Collaborative	Intelligent	Intelligent
The ability to detect unknown attacks	No	No	Could be	Could be
The ability to analyze the content of encrypted streams	Yes	Yes	No	Yes
Encrypting exchanged alerts	No	No	No	No
Diffusion of detected attacks	No	Using alert system message	No	No

In [6] the authors proposed Ajit Kumar Gautam, Vidushi Sharma, Shiv Prakash and Manak Gupta proposed an improved Hybrid IDS. The Improved hybrid IDS is combination of anomaly based detection and honey pot technology with KFSensor and Flowmatrix. The Honey pot is used to attract more and more attackers, the detection obtained can be used to create new signatures and update the database. Finally anomaly can be used to detect unknown attack in the whole network.

The Cloud Detection and Prevention System (CIDPS) architecture [7] is illustrated and presented as a workflow scenario. Sensor inputs or alerts generated while monitoring network, host, platform and applications together with the latest CIDPS challenges and enterprise CIDPS policies and their updates, drive through the CIDPS Trust Management system to be analyzed. Inference Engine (IE) is the logical and main part of IDE.

In CIDS architecture [8] each node has its own analyzer and detector components that are connected to the behavior and knowledge based databases. The individual analysis reduces the complexity and the volume of exchanged data, but at the expense of the node processing overhead. This framework contains CIDS components, cloud system components and NIDS components.

Table 1 presents the different architectures of intrusion detection systems in a comparative table.

3 Virtual Machine with Smart Intrusion Detection Model

The Cloud Computing is based on the principle of virtualization in which the different services and applications are deployed in virtual machines. The virtual machines use various operating systems (OS) and expose different application to the cloud customers, thus any vulnerability in these systems and applications can be remotely exploited by hackers hence the importance of implementing intrusion detection systems in the Cloud at the virtual machine level.

We propose a smart host based intrusion detection system (SHIDS) as security tool to monitor the hypervisor and virtual machines on that hypervisor, detect malicious activities at the VM level and protect against attacks targeting applications running on virtual machines, this approach provides many benefits in terms of portability and costs. The virtual machine is equipped with a SHIDS (Smart host based intrusion detection system) it is a customized HIDS (Host based intrusion detection system) designed to support data mining in order to detect unknown attacks, the proposed SHIDS behaves like a HIDS and controls the state of the virtual machine, since it has access to its stored information, whether in RAM, in the file system, log files or audit trails. In addition the SHIDS can analyze all the activities on the virtual machine hosting the cloud services.

The SHIDS has the ability to analyze the content of encrypted streams since SHIDS has access to the encryption keys and certificates on the machine where it is installed while the NIDS (Network based intrusion detection system) cannot analyze encrypted traffic. The SHIDS can detect easily the “Trojan” attacks while this type of attack is difficult to detect by a NIDS. The HIDS is the suitable solution to secure our virtual machines in the Cloud environment.

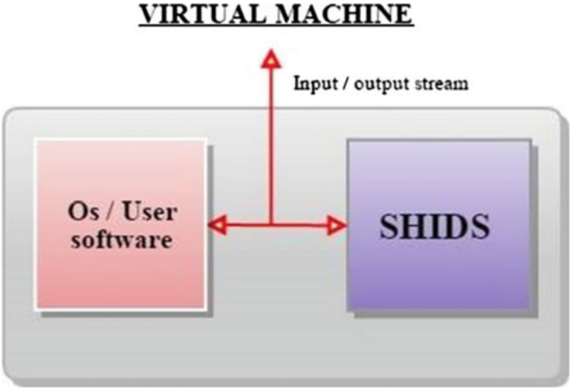
3.1 Virtual Machine Components

Figure 1 shows the virtual machines equipped with a SHIDS (Smart Host based IDS).

The proposed virtual machine contains the following components:

- Os software: Is the system software that manages computer hardware and software resources, it can be Linux, Microsoft windows etc.
- User software: It is the application provided by the cloud and designed to be used by the customers of the cloud.
- SHIDS (Smart Host Based Intrusion Detection System): it refers to an intrusion detection system that is placed on a single host system. It is a customized HIDS whose detection engine is modified and improved in order to support and integrate data mining and machine learning modules to detect unknown and new attacks that are not stored in the IDS database.

Fig. 1 Virtual machine integrating a smart host based IDS



3.2 SHIDS Architecture

The different components (Fig. 2) composing the SHIDS are:

- Logging and alert system: is the module responsible for logging alerts, their storage in the database and the exchange of alerts with the other intrusion detection systems.
- IDS database: It is a MySQL database that stores the signatures of the various attacks known previously.
- Behavior database: It is a database used to store the past behaviors of individual users.
- Configuration interface: It is the interface used for the configuration and the parameterization of the SHIDS.
- Data Mining Engine: Is a program that allows using data mining techniques such as statistics, frequent pattern mining, clustering and classification to detect the new attacks that are unknown previously.
- knowledge base: It is a database that stores the attacks detected by the data mining engine after the application of different data mining techniques.

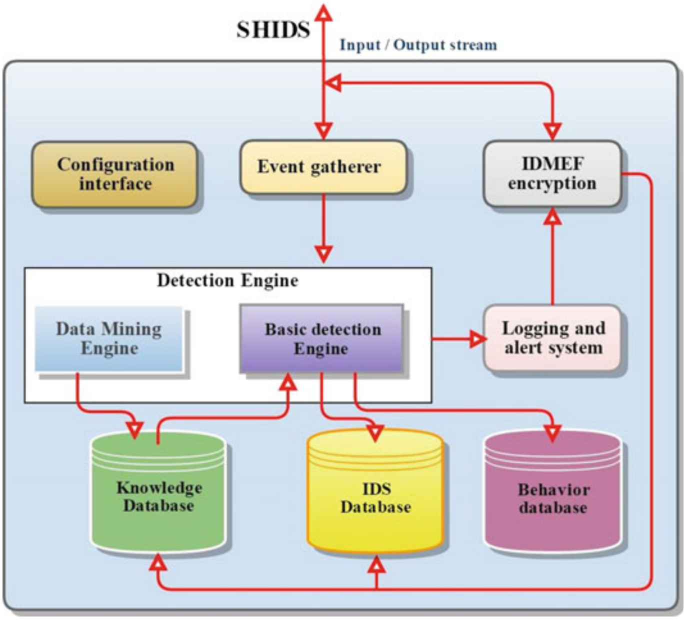


Fig. 2 SHIDS architecture

The SHIDS behaves like a demon or a standard service on a host system that detects suspicious activities. The SHIDS can monitor the machine activities, user activities and malicious activities like worms, virus and Trojans. The event gatherer which collects all the events coming from outside to the virtual machine, send the coming streams to the basic detection engine which is the customized program responsible of analyzing the events and detecting attacks based on the signatures stored in the IDS database. When an input is received the basic detection engine solicits the database to find a matching and in order to improve the SHIDS we added a behavior database using anomaly detection technique based on statistical measures, it focuses on characterizing the past behavior of individual users or groups of users to detect significant deviations. Many metrics are taken in consideration such as the connection time of the user, the number of password failures and CPU and memory usage. If no matching is found this means that the signature is not stored in the database so the basic detection engine will call the data mining engine, this latter applies the data mining techniques such as the clustering, association and fuzzy logic to check if the received signature matches an intrusion, in this case if an intrusion is detected it will be stored in the knowledge base and the basic detection engine will be notified. Finally the logging and alert system sends the detected attack in an IDMEF format to the central ids in order to update the other SHIDS as shown in Fig. 3.

3.3 Communication Between SHIDS in Virtual Machines

The HIDSs in each virtual machine can exchange information about intrusions using the IDMEF (intrusion detection messages exchange format) [9] format.

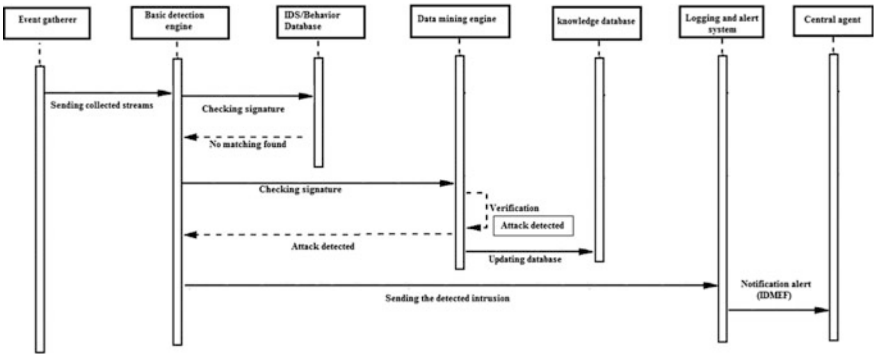
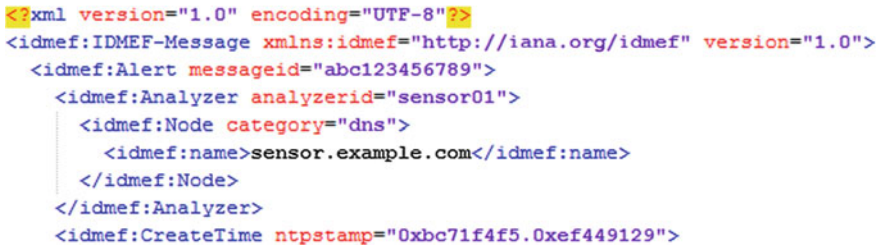


Fig. 3 UML sequence diagram of the SHIDS



```
<?xml version="1.0" encoding="UTF-8"?>
<idmef:IDMEF-Message xmlns:idmef="http://iana.org/idmef" version="1.0">
  <idmef:Alert messageid="abc123456789">
    <idmef:Analyzer analyzerid="sensor01">
      <idmef:Node category="dns">
        <idmef:name>sensor.example.com</idmef:name>
      </idmef:Node>
    </idmef:Analyzer>
    <idmef:CreateTime ntpstamp="0xbc71f4f5.0xef449129">
```

Fig. 4 Example of IDMEF message

The IDMEF data model is an object-oriented representation of the alert sent to central intrusion detection managers by intrusion detection analyzers [9]. IDMEF (intrusion detection messages exchange format) is a data format used to exchange incident reports between intrusion detection systems, intrusion prevention systems and software that must interact with them. IDMEF messages are designed to be easily handled. The details of the format are described in RFC 2007. The RFC 4765 [10] presents an implementation of the XML data model and associated DTD. An example of an IDMEF message is shown in Fig. 4.

The IDMEF format was chosen for its openness and extensibility. IDMEF message can be either an alert or a heartbeat. The IDMEF library is generated by JAXB based on the IDMEF XML schema provided by the RFC [10]. JAXB generates a class and their members based on a specific XML Java mapping. To protect the exchange of IDMEF alerts we used RSA encryption to secure the exchanges between the central agent and the other SHIDSs so that nobody can read the IDMEF alerts on the network, the rng-tools [11] (Hardware Random Generator) was used to prevent users from sniffing traffic from the insiders.

4 The Proposed Architecture

Two scenarios are possible to secure the cloud environment, the first one is to set up a distributed intrusion detection architecture in which each HIDS communicate by exchanging IDMEF alerts, in this case if an intrusion is detected by a SHIDS, it will be communicated to all other SHIDS to update their database as shown in Fig. 5.

The drawback of this architecture is that it will burden the network traffic because there are a lot of exchanges of alerts between SHIDS in addition to that if the number of servers is very large it will generate a lot of alerts which degrades the performance of the SHIDS since the alerts come from several sources.

The second scenario is based on a centralized IDS architecture (Fig. 6) that is based on the principle of collaboration between many SHIDS deployed on the different virtual machines (assuming that we have n virtual machines VM1, VM2 …. VMn and m Physical machines PM1, PM2…. PMm with m # n) in the cloud to detect and protect against attacks targeting applications running on these virtual

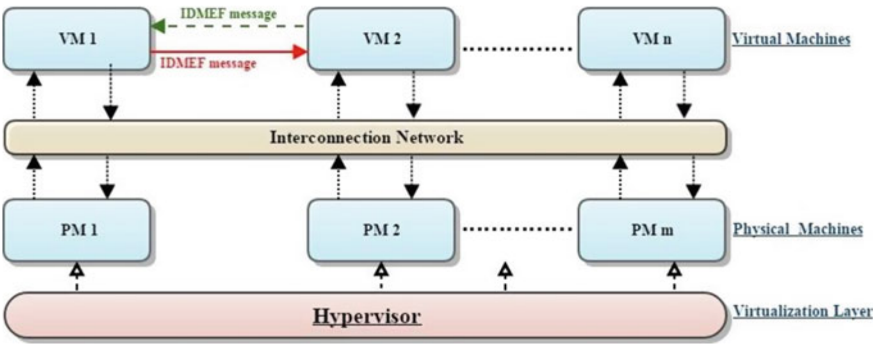


Fig. 5 Distributed intrusion detection architecture

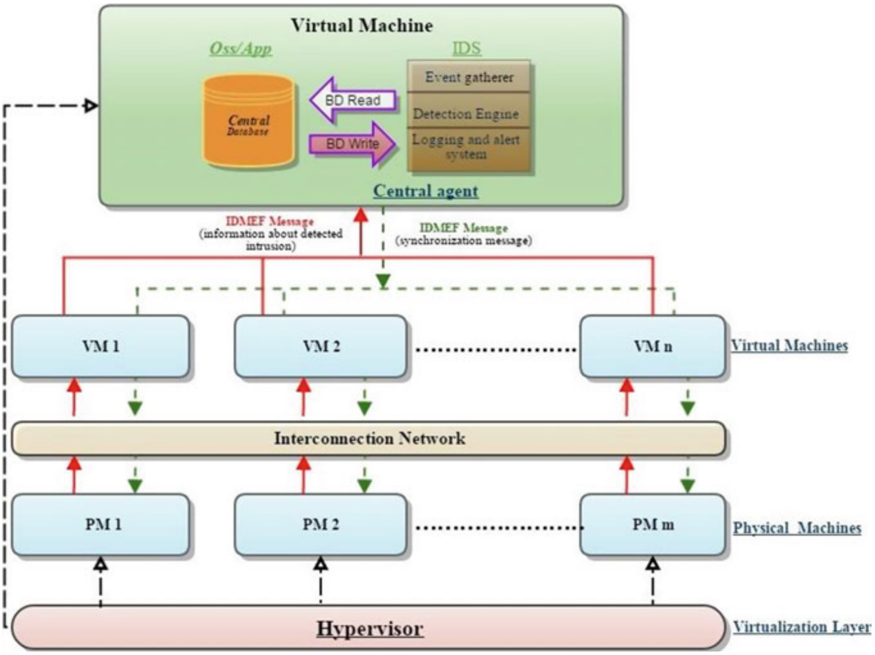


Fig. 6 Centralized intrusion detection architecture

machines, this approach provides many benefits in terms of portability and costs. The concept of this model is that the different SHIDSs are placed in each VM and cooperate with each other by exchanging alerts about detected intrusions. In our model there is a central agent that is responsible for the reception of notifications from the other SHIDS as well as writing and reading from a central database in order to synchronize the local database of each SHIDS. Thanks to this concept the different SHIDS are synchronized and each detected attack is communicated to other neighbors by the central agent. The model is improved by using the data mining and machine learning techniques to detect unknown attacks.

The model proposed in Scenario 2 offers more benefits than the one proposed in Scenario 1, so we adopted the architecture of the second scenario.

5 Experiment

The objective of this experiment is to valid the proposed architecture and test the detection of intrusions that are already stored in the database, in this case we will simulate four types of attacks (remote login with a wrong password, accessing a protected resource, trying to do administrative tasks such as changing the root password, and trying to log in without password) on a server in a virtual machine and verify that the attacks are detected by the SHIDS and sent in an IDMEF format to the central agent finally check that this later will notify all other SHIDS to update their databases. The data mining module of our architecture will be tested in the next paper taking into consideration the performance and the validity of the results.

The experiment was performed by using OpenStack [12] installed on an Ubuntu machine [Intel(R) Core(TM) i3-2370 M CPU @ 2,70 GHz 2,70 GHz, 8 Go de RAM]. We created three virtual machines (VM1, VM2 and VM3). During this experiment we put the focus on the following modules of OpenStack:

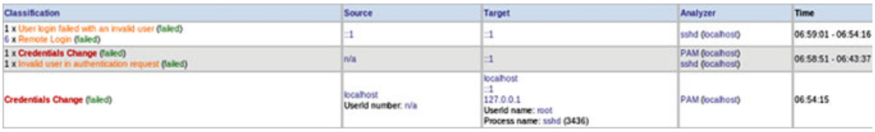
- Glance: It is the image Service of OpenStack that we used to perform registration and discovery of virtual machine images.
- Nova: It is the compute service of OpenStack that we used to automate and manage pools of computer resources.
- Horizon: It is the OpenStack Dashboard that provides a graphical interface to access and automate cloud resources.

We used Prelude [13] as a hybrid IDS (IDS) compound of heterogeneous types of sensors:

- NIDS: Network Intrusion Detection System (Snort).
- HIDS: Host based Intrusion Detection System (Samhain) that we customized to support data mining and machine learning techniques.

Prelude is designed in a modular way so as to adapt to any environment, the Manager is the central component that receives events from the different sensors and processes them, the Libdb is the library that provides an abstraction layer for the storage of IDMEF alerts in a database, the LML is the logs processing module and Prewikka is a web based graphical user interface (GUI) for Prelude.

On VM1 we installed Prelude Manager which is the module that processes and centralizes the system alerts, on VM2 and VM3 we installed the SHIDSs, the idea is that if one of these machines (for example VM2) detects an intrusion, it will use the IDEMF format to inform the central IDS (installed on VM1) and this later will



Classification	Source	Target	Analyzer	Time
1 x User login failed with an invalid user (failed) 6 x Remote Login (failed)	::1	::1	sshd (localhost)	06:59:01 - 06:54:16
1 x Credentials Change (failed) 1 x Invalid user in authentication request (failed)	n/a	::1	PAM (localhost) sshd (localhost)	06:58:51 - 06:43:37
Credentials Change (failed)	localhost Userid number: n/a	localhost ::1 127.0.0.1 Userid name: root Process name: sshd (3436)	PAM (localhost)	06:54:15

Fig. 7 Alerts provided by the Prelude sensors displayed on the Prewikka

update the VM3 with the new intrusion detected. The different exchanged alerts are encrypted using RSA standard. In order to be able to persist IDMEF alerts describing the intrusion detected we used Libpreludedb [14] Library that provides an abstraction layer for storing IDMEF alerts. In the experiment we simulated attacks on the VM2 and check if these attacks are transmitted in IDMEF format to the central IDS (VM1). In the central IDS the alerts are stored using Mysql database. To display the different alerts sent by the different sensors we use a web graphical interface called Prewikka. The Fig. 7 shows the alerts provided by the Prelude sensors and displayed on the web interface of the central IDS.

6 Conclusion

To sum up Cloud computing can be considered as a new concept that has brought many benefits but several security threats and vulnerabilities have appeared with this new concept that is why the cloud providers must identify these security issues and try to protect against them.

In this paper we have outlined the different architectures of intrusion detection systems in the cloud and provide a comparative study between them finally we implemented the proposed architecture using Openstack. In the next paper we will improve the architecture through the use of data mining techniques to detect new and unknown intrusions, improve accuracy and speed of intrusion detection and ensure a good adaptive capacity and scalability.

References

1. Derfouf, M., Mimouni, A., Eleuldj, M.: Vulnerabilities and storage security in cloud computing. In: International Conference on Cloud Computing Technologies and Applications —CLOUDTECH 2015, Marrakech, Maroc, 2–4 June 2015
2. Roschke, S., Cheng, F., Meinel, C.: An advanced ids management architecture. J. Inf. Assur. Secur. **5**, 246–255 (2010)
3. Kuzhalisai, M., Gayathri, G.: Enhanced security in cloud with multi-level intrusion detection system. In: IJCCT, vol. 3, Issue 3 (2012)
4. Gul, I., Hussain, M.: Distributed cloud intrusion detection model. Int. J. Adv. Sci. Technol. **34**, 71–82 (2011)

5. Modi, C.N., Patel, D.R., Patel, A., Rajarajan, M.: Integrating signature apriori based network intrusion detection system (nids) in cloud computing. Proc. Technol. **6**, 905–912 (2012)
6. Gautam, A.K., Sharma, V., Prakash, S., Gupta, M.: Improved hybrid intrusion detection system (HIDS): mitigating false alarm in cloud computing. JCT (2012)
7. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J.C.: Taxonomy and proposed architecture of intrusion detection and prevention systems for cloud computing. In: Cyberspace Safety and Security, pp. 441–458. Springer (2012)
8. Kholidy, H.A., Baiardi, F.: CIDS: a framework for intrusion detection in cloud systems. In: 2012 Ninth International Conference on Information Technology: New Generations (ITNG), pp. 379–385. IEEE (2012)
9. Debar, H., Curry, D., Feinstein, B.: The intrusion detection message exchange format, internet draft. Technical report, IETF Intrusion Detection Exchange Format Working Group, Jul 2004
10. Debar, H., Curry, D., Feinstein, B.: The intrusion detection message exchange format IDMEF, RFC 4765, IETF, 1 Mar 2007. http://tools.ietf.org/html/rfc4765
11. https://www.gnu.org/software/hurd/user/tlecarrour/rng-tools.html
12. https://www.openstack.org/
13. https://www.prelude-siem.org/
14. https://www.prelude-siem.org/projects/prelude/wiki/InstallingPreludeDbLibrary