# A Global Intrusion Detection System using PcapSockS Sniffer and Multilayer Perceptron Classifier

Azidine Guezzaz[1], Ahmed Asimi[2], Younes Asimi[2,3], Zakariae Tbatous[2] and Yassine Sadqi[4]

*(Corresponding author: Ahmed Asimi)*

M2SC Team, Technology High School Essaouira, Cadi Ayyad University Marrakech[1]
Km9, Road Agadir, Essaouira Aljadida BP.383, Morocco
LaB SiV Laboratory, SCCAM Team, Faculty of Sciences, IbnZohr University[2]
Agadir, Morocco
Technology High School Guelmim ,IbnZohr University[3]
Agadir, Morocco
Km9, Road Agadir, Essaouira Aljadida BP.383, Morocco
Polydisciplinairy Faculty, Sultan Moulay Slimane University[4]
Beni Mellal, Morocco
(Email: asimiahmed2008@gmail.com)

*(Received Aug. 1, 2017; Revised and Accepted Apr. 12, 2018; First Online Feb. 10, 2019)*

## Abstract

The evolution of networks requires a high monitoring of their resources and a reliable security of exchanges to obtain a faithful communication between their systems. The automatic detection of intrusions has become an active discipline due to the increased needs of computer security and large malicious traffic with attacks that can infect systems. Intrusion detection and prevention systems are the recent technologies used to monitor data activities. Thus, their assessment is very useful. The main goal of this paper is to analyze some sniffers tools and to assess the performances of certain intrusion detection and prevention systems. The analysis measures assess the authenticity, availability, integrity and confidentiality but also certain parameters related to security, such as: Detection type, filtering detection method, real time reaction, updating, alerting, logging. A novel detection approach is designed to perform the monitoring of networks. It is based on PcapSockS sniffer that collects data and on multilayer perceptron to analyze and make the appropriate decisions. This approach makes a reliable detection by minimization number of false positives and elimination of false negatives.

*Keywords: Classification; Intrusion Detection; Performances; Security; Sniffing*

## 1 Introduction and Notations

As long as the intrusions detection makes a network safer, prevention aims to make appropriate decisions by reacting in real time. The IDPSs (Intrusion Detection and prevention Systems) are designed for networks security needs. The sniffers tools are used to capture the circulated packets within network interfaces; they decode certain packets of a specific interest. The IDPS are used to control exchanged events through networks, to inform the existence of an intrusion, and then to take a concise action and bring systems into a safe state. The current IDPS are oriented towards automatic responses to intrusions in real time with alerts. They can be classified according to the type of detection approach, level of monitoring, frequency of use or nature of reaction. False positives are generated when a detection system identifies normal activity as an intrusion, while false negatives correspond to undetected intrusions, so no alert is generated. It is impossible to find a standard detection tool that can overcome all limitations. The second section presents a state of art on intrusion detection, sniffing and multilayer perceptron. The performances analysis is cited in the third part, based on security objectives and on parameters related to security. For the fourth section, the proposed solutions are described. The article is accomplished by a conclusion and the future works. In this paper, we use the following notations (Table 1):

Table 1: Notations

| | |
|---|---|
| $f$ | : Sigmoid Function. |
| $(X_i)_{i=1..n}$ | : The presented inputs. |
| $X_i = (x_{i,j})_{j=1..m}$ | : The presented occurrences to input $X_i$. |
| $W^{(0)} = (w_{i,0})_{i=1..n}$ | : The initialized weights. |
| $W_i = (w_{i,j})_{j=1..m}$ | : The associated weights to input $X_i$. |
| $w_{0,i}$ | : Initialized Bias to 1 and associated to input $X_i$. |
| $a_i$ | : Weighted sum associated to input $X_i$. |
| $y(a_i) = f(a_i)$ | : Calculated output associated to input $X_i$. |
| $\epsilon_i$ | : Calculated error associated to an entry $X_i$. |
| $W_i^{op} = (w_{i,j}^{(op)})_{j=1..m}$ | : Optimal system solution (Training Algorithm) for $X_i$. |
| $W_{0,i}^{op}$ | : Optimal System Bias (Training Algorithm) for $X_i$. |
| $a_i^{(op)}$ | : Optimal weighted sum associated to an input $X_i$. |
| $a_i^{(max)}$ | : Maximum weighted sum associated to an input $X_i$. |
| $W^{max} = (w_j^{(max)})_{j=1..m}$ | : Maximum weights. |
| $w_0^{(max)}$ | : Maximum bias. |
| $d = +1$ | : Normal Output. |
| $d = -1$ | : Anormal Output. |

## 2 State of Art

This section gives a state of art of IDPS, the sniffing techniques and multilayer perceptron.

### 2.1 Intrusion Detection and Prevention

Intrusion detection is a set of techniques used to detect undesirable activities. An intrusion attempts to violate one of security objectives [2, 11]. An IDPS can be software or hardware which can detect malicious events that attempt to infect a security policy. IPS (Intrusion Prevention Systems) are considered as second generation detection systems, designed to make necessary decisions to stop the detected intrusions accurately. There are two fundamental detection methods [2, 7, 11, 12, 17, 18, 24]:

- Scenario approach that identifies an intrusion using a configuration known for malicious activity.
- Behavioral approach that attempts to identify malicious based on a deviation from normal activity. It is proposed by J. P. Anderson (1980) and extended by D. E. Denning (1987).

An IDPS can control and detect accurately the abnormal activities by blocking them quickly. It is characterized by following properties:

- Real time takes into account time constraints and delays related to the results.
- Response time which determines the duration between activation and time of the results.
- Blocking is used to interrupt the passage of suspicious activities.
- Alert is a message generated after detection to inform the manager about the existence of an intrusion.

The IDPS architecture is composed by [10,12,15,16,24,28] (Figure 1):

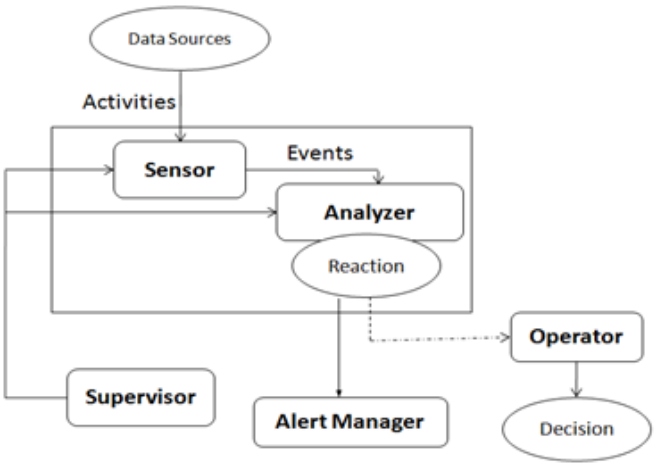


Figure 1: Classical architecture of IDPS

- Data sources contain the data that reflects what is happening on the hosts and the traffic of packets that is intercepted by a network monitor.
- Activities are collected within data sources and stored in database.
- Sensor observes the system activities through data sources and provides a sequence of events that inform evolution of the system state.

- Events represent the preprocessed activities presented to analyzer.
- Analyzer determines if the events contain malicious activities.
- Reaction is guaranteed by activating the countermeasures to end the detected attack.
- Supervisor is responsible to analyze alerts and has a global vision toward system.
- Alert manager is used to generate alerts after detection.
- Operator is a part of IDPS that make a final decision.

The majority of current IDPSs integrate heterogeneous technologies like, VPN (Virtual Private Network), antivirus, antispam, *etc.*

## 2.2 Sniffing Techniques

The sniffing is a process used to intercept and analyze the network traffic. It listens to public conversations in computer networks [6, 14, 27]. The sniffers may be used as a hardware or software solution or as software only to manage and ensure the network security. They can also be used by unauthorized uses. An intruder can learn network configuration information by sniffing. There are different types of sniffing packets [21, 22]:

- IP (Internet Protocol) sniffing collects all IP packets traveled through a network.
- MAC (Medium Access Control) sniffing captures the corresponding frames to supervised interfaces MAC addresses.
- ARP (Address Resolution Protocol) sniffing intercepts ARP packets used to query the ARP cache.

The sniffers are constituted by the components described by Clincy & Abi Halaweh in [3, 27]:

- Hardware is represented by Network Interface Cards and activated in sniffing mode.
- Driver captures data from the network cards, applies a number of filters and stores it in a memory.
- Buffer stores the captured traffic or transfers it to permanent storage.
- Analyzer is responsible to analyze the traffic in real time taking into account the criteria needs.
- Decoder receives a stream of bits and interprets them to finally build a descriptive texts format.
- Editor changes the traffic using a unified format and then converts it and retransmits it in the network.

The filtering is an essential operation to classify packets that are captured using filters according to the needs of capture. The simulation with sniffing tools is used in learning of computer networking, allows a good understanding of network concepts and topologies. The study carried on [14] highlights the difference between these two principles libraries. The main capture libraries are libnet and libpcap. The Airpcap adapter is used on hosts running to listen in to wireless traffic in monitor mode.

## 2.3 Multilayer Percepron

The birth of artificial neural discipline dates back to the 1940s with W. McCulloch and W. Pitts who showed that with such networks, we could in principle calculate any arithmetic or logical function [23]. The training is a dynamic and iterative process [29] used to modify the parameters of network in reaction with stimuli that receives from its environment. The supervised training adjusts network parameters by a direct comparison between the actual network output and the desired output. The unsupervised training involves no target values. It tries to associate information from the inputs with an intrinsic reduction of data dimensionality or total amount of input data. The type of training is determined by how the parameter changes occur [16, 26]. The MLP (Multilayer perceptron) (Rosenblatt 1957) is a neural network that composed of successive feedforward layers connecting neurons by weighted links [15, 16, 29]. The input layer is used to collect the input signals and the output layer provides responses. One or more hidden layers are added for transfer. The training of MLP is performed by the error gradient propagation. In the 1980s, an error propagation algorithm was invented [29]:

**Algorithm 1:** Back propagation training

1) DBA : Training Base.
   $X_i = (x_{i,j})_{j=1...m}$ : Inputs.
   $C_i = (c_{i,j})_{j=1...m}$ : Desired Results for $X_i$.
   $W_i = (w_{i,j})_{j=1...m}$ :Weights for $X_i$.
   $\theta_i$: Calculated Results.
   $\lambda_i$:Training rate.

2) BEGIN : Calculate $W_i$ for the input $X_i$

   For $i$ from 1 to $n$ do
   Initialize the weights randomly
   Optimization of weigts:
   For $j$ from 1 to $m$ do
   $w_{i,j} = w_{i,j-1} + \lambda_i(c_{i,j} - \theta_i)x_{i,j}$
   EndFor
   *EndFor*

3) END

The examples of the training basis are shown successively in order to adjust the weights by accumulating the

calculated gradients. The training is stopped when the calculated error is less than a certain threshold.

# 3 Our Contribution

In this section, we describe the results of performances analysis carried on some network sniffers and certain IDPS. It proposes a novel model of IDPS based on PcapSockS sniffer and multilayer perceptron.

## 3.1 Results of Performances Analysis

The sniffers analyze data from all the network layers. If the application level analysis fails to identify the problem and find a solution, sniffers can dig into lower level details. Based on various criteria and referring to the detailed study in [13, 14, 21, 27], we arrive at a classification of the following systems (Table 2):

After this assessment, the majority of sniffers above use libpcap library to intercept traffic and include a filtering system. They are highly available to monitor wired and wireless networks with a high flows supporting a large number of protocols. The study helps us to discover certain limitations. The actual sniffers are more efficient, allowing real time analysis. They capture packets from the network and decode them into human readable format. To be able to choose a better detection system before installing it on the affected network, it is useful to test and evaluate the operational efficiency of these systems.

- Snort is an open source network IDPS, developed by Sourcefire. It is a scenario and anomaly system [8, 9, 25].
- Suricata is an open source IDPS that uses the snort rules, its important advantage is multithreading that means reduction of time and gives also a high performances. David and Benjamin analyzed Snort and Suricata and conclude that Suricata is relevant and exact than Snort [8, 11, 25].
- Mc Afee Host Intrusion Prevention is aiming to protect systems, resources and applications. It establishes reporting and gives an exact management, progressed and easy to use [11, 16].
- Net ASQ is an engine integrating intrusion prevention and eliminates intrusions in real time. Its hardware alternative arranges a Watchdog which realizes regularly tests of activities [11, 16].

To satisfy this assessment, we propose, the degree of guarantee of the safety objectives: authenticity, confidentiality, integrity, availability [11, 16, 20] (Table 3):

Most of the existing solutions concerning intrusion detection are related to the setting up of NIDPS in association with some HIDPS and other software types of management. It has been observed that NIDS become less effective even when presented with a bandwidth of a few hundred megabits per second.

## 3.2 Our Proposed Approach

This proposition is based to avoid some vulnerabilities and limits. The structure of system is (Figure 2):

Our IDPS system is constituted by the different components bellow:

- Data sources: The circulated data flow within the network are intercepted and processed to monitor and make an effective decision.
  - ∗ High level means the monitoring of various activities within the high layers.
  - ∗ Low level means the monitoring of various activities within the low layers.
- Sensor observes the data and provides the analyzer a sequence of activities that inform the evolution of the system state.
  - ∗ PcapSockS Sniffer intercepts traffic from the low and high level.
  - ∗ Activities: the collected data are stored in a collection base in the form of activities.
- Analyzer is made to take a decision by exploiting the implemented detection methods.
  - ∗ Normalization: is located directly after sniffing which is used to eliminate the potential ambiguities and to have a uniform structure of activities.
  - ∗ Comparator is a component that compares an event with the contents of the intrusion basis.
  - ∗ Events: the normalized activities become events and presented to analyzer.
  - ∗ Multilayer Perceptron Classifier is able to distinguish the normal behavior from the new data.
  - ∗ Notification: after detection of intrusion the analyzer sends notification to manager.
  - ∗ Updating: the intrusions basis is updated in order to increase the possibilities of the new detections and to facilitate the next analysis.
- Manager is responsible for the management and analysis of the alerts generated by the analyzer. It contains:
  - ∗ Management: the manager is responsible to analyze alerts and take action to prevent the damage of intrusion.
  - ∗ Real time blocking means the realtime response to block intrusion and anticipate connection.
  - ∗ Automatic reaction provides reaction mechanisms to cope with detected intrusion or reduce their effect.
- Supervisor is the person who administers the various components of that system. He has a global vision on the system.

Table 2: Performances assessment of some network sniffers

| Network sniffers | S/H | Library | Filtering | Flow | Availability | Alert | Real time |
|---|---|---|---|---|---|---|---|
| Tcpdump | S | Libpcap (Winpcap) | ++ | Flow of Ethernet networks | Very economical installation file size: 484 KB | - - | - - |
| Wireshark | S | Libpcap (Winpcap) | ++ | Flow of Ethernet and wireless networks. | 81 MB after installation. | - - | ++ |
| PACKETYZER | S | Libpcap (Winpcap) | ++ | Flow of Ethernet, FDDI, PPP, Token Ring and wireless networks. | -supports 483 protocols.<br>-Decodes and edits packets. | - - | ++ |
| Netflow CISCO | S<br>H | Libpcap | ++ | High flow networks (Gigabit). | Very high (provides valuable information about users, network applications, peak hours).<br>-2GH Dual processor.<br>-2GO Memory. | ++ | ++ |
| Colasoft Capsa | S | Libpcap | ++ | Flow wired and wireless networks over 802.11a, 802.22b, 802.11g and 802.11n | -No Tolerant with the attacks: ARP, TCP port scanning,<br>-Signals DOS attacks<br>-653 MB on after windows 7 installation.<br>- Free version is available with limited features. | ++ | ++ |
| PRTG Network Monitor | S<br>H | Libpcap | ++ | High flow | -Integrates SNMP, Packet (Sniffing and Net flow).<br>-monitors 24/7 network.<br>- Includes over 200 types of sensors.<br>-Less than 30 protocols (Free).<br>- More than 30 protocols (Com) | ++ | ++ |
| Kismit | S | Libpcap | ++ | Flows of wireless networks 802.11n, 802.22b 802.11g and 802.11a | High (supports any wireless card rfmon) | ++ | ++ |
| Scapy | S | Libpcap and Libnet | ++ | Injectes the 802 frames | - Generates and receives quick and accurate traffic.<br>- Decodes packets of a number of protocols. | ++ | ++ |
| OmniPeek | S<br>H | Libpcap | ++ | Ethernet, Gigabit, 10 Gigabit, 208.11 a / b / g / n / ac wireless, VoIP, Video, MPLS and VLAN | -captures on multiple networks simultaneously.<br>- Several hundred protocols<br>- WPA, WPA2 and PSK Decoding. | ++ | ++ |
| ETHERAP | S | Libpcap | ++ | Flows of Ethernet, FDDI, Token Ring, ISDN. | - Is only available for GNU / Linux systems. | - - | ++ |
| Soft Perfect Network Protocol Analyzer | S | Libpcap | ++ | Flows of Ethernet networks | -Analyzes of fragmented floors.<br>-Defragments and reassembles the packets.<br>- Size of the installation file 4.87 Mb. | - - | ++ |
| Airodump | S | Libpcap | ++ | - Wireless networks 802.11.<br>- Supports 4.2 GHz channels | -Identification the coordinated access points.<br>-Writes the several files containing details of all seen access points and clients. | - - | ++ |

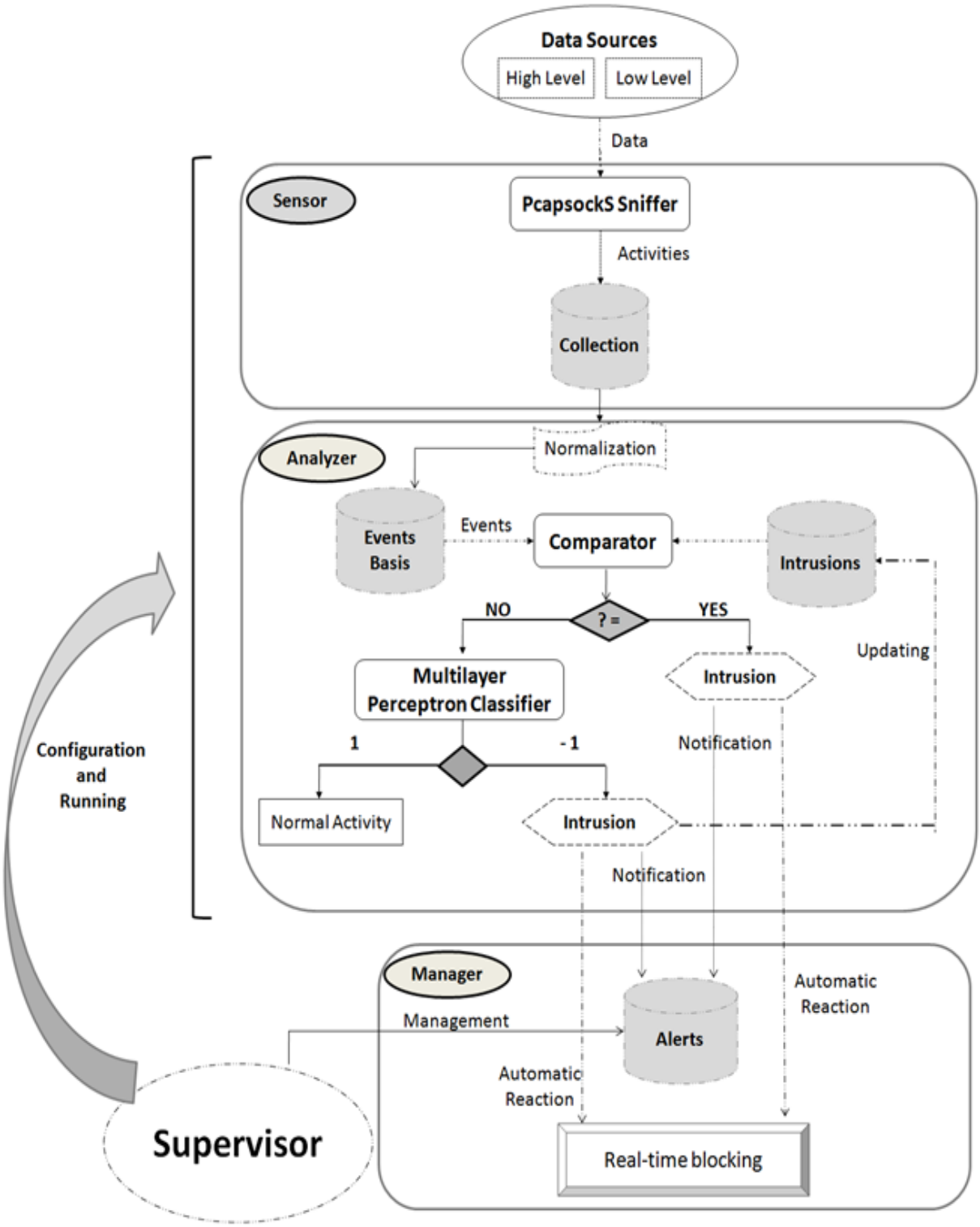


Figure 2: Proposed IDPS based on PcapSockS sniffer and MLP classifier

Table 3: Assessment based on security objectives

| | Snort | NetASQ | Suricata | Winpooch | MC Afee Intercept | Bro | Net Screen Intrusion | Cisco Net Ranger |
|---|---|---|---|---|---|---|---|---|
| Authenticity | High | Protocole IPSEC, Certificats X.509, PKI infrastructures, SSL | High | MAC algorithm | High | High | Medium | ACL Lists |
| Confidentiality | TowFish Algorithm | DES, 3DES, AES, BlowFish. | Cryptografic functions of TLS protocol | —— | —— | Include SSH functions | RC4 Algorithm | —— |
| Integity | 5MB/s, 10MB/s, 4GB/s | High speed MD5, SHA1, SHA2 | Hush functions of TLS protocol | Includes scan of ClamWin antuvirus | —— | high-level semantic analysis/ detect a large number of protocols | 5MB/s, 100MB/s, 1GB/s/ | High reliability |
| Availability | Continuous frequency | Continuous frequency | Continuous frequency | High | Continuous frequency | Continuous frequency | Continuous frequency | Continuous frequency |

The use and management of databases is very important in this approach; we opted for using of four databases:

- Collection basis is composed by activities that intercepted within networks by PcapsockS sniffer.
- Events basis is constituted by the normalized activities.
- Intrusions basis includes all known attacks by using a certain format. There is no standard for the coding of attacks. It is updated after detection.
- Alerts basis contains different alerts generated after detection by our IDPS.

Our IDPS performs the first monitoring based on signature detection. Thus, it needs signatures basis that will satisfy this type of detection. Therefore, we have to conceive intrusions basis which characterize the anomalies of the monitored network (Figure 3).

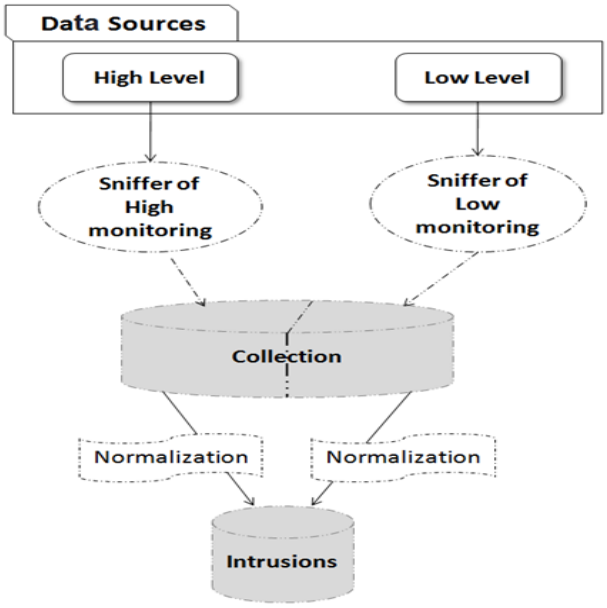


Figure 3: Conception of intrusions basis

The monitoring is done in a hybrid and complete way by controlling all levels of data sources. For this, we use the most famous sniffing tools more used currently to meet our needs. For example, we use Scapy [13,14,16] for high level sniffing and Wireshark [3,14,21] for the lower layers. The collected activities are recorded in a collection basis. In this case, the sniffing type takes place during so called abnormal operation of the network to collect abnormal activities characterizing the anomalies of the monitored network. The intrusions are used by detection systems. They are stored and integrated into their database at each infection. The detection is carried out by comparing the event collected the contents of the intrusion database. To implement the new approach, various phases are used:

#### 3.2.1 Collection and Filtering Phase

The proposed design in [1] focuses on the combination of current performances of high sniffers and minimization of various limitations. It is a distributed model consisted by two main components:

- The kernel is composed by two processors to capture and filter the traffic.
- The operator decodes and normalizes the elected traffic.

These components are described in the figure below (Figure 4):

This traffic is composed of a set of bits and frames, it's saved in a temporary basis to apply the BPF (Berkeley Packet Filter) and then meet adequate collection conditions. Libpcap provides the possibility to introduce the filters to filter traffic: PBF, SWIF. It applies the filters on traffic in the basis in order to choose the elected packets. This latter is redirected to the operator space. The decoding processor normalizes and stores the chosen traffic in the collection Database. In the high level, we use the sockets mechanism to ensure a reliable collection. The TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) sockets are implemented for this purpose. Raw sockets are used to reinforce the interception to the low level with libpcap. The collected traffic is saved in a temporary basis to apply the filter LSF and redirected directly to Collection Database. Our sniffer collects data in three modes:

- Connection oriented mode requires a prior connection establishment between communicating entities.
- Connectionless mode cannot guarantee a reliable connection, insertion errors, wrong delivery, duplication, or non sequencing delivery packets.
- Raw mode can provide both services in connection oriented and connectionless mode.

The filtering provides a considerable gain; it avoids the congestion and the saturation of memory. The filtering is a very useful to meet the various network services using mainly in intrusion detection. The treatments are in real time. Take into account the time constraints which are as important as the accuracy of the results for this system synchronizes multiple tasks that take place and the possibility of including several shorter threads in a single process. To show the performances provided by PcapSockS Sniffer, it is very useful to compare it with other sniffers which have demonstrated their reliability (Table 4):

The new model combines libpcap and sockets functions to capture the packets, filters traffic taking into account the capture needs. All treatments are in real-time and Encryption of transactions between the sniffer and Collection database.

#### 3.2.2 Preprocessing and Normalization Phase

The preprocessing phase is the most labor intensive, due in particular to the lack of structuring and the large amount of noise existing in the raw data used. It consists in structuring the activities in order to prepare them for a future analysis. The significant formatting is required before analyzing and classifying traffic. The normalization is carried out also to establish a pattern of activities facilitating the distinction between the activities and allowing an extraction of useful fields if necessary. The events are stored in a database table which contains columns to specify the fields and contains the occurrences with string type. A hash function is applied to compute the signatures of the content. We realize a particular coding for the enumeration of the occurrences and adapt them to the entries of the model which accept in principle only integer, real or Boolean entries. The hashing and coding techniques guarantee also a certain rapidity and integrity. The input layer receives successively the preprocessed occurrences. An occurrence is subdivided into a set of fields. Each field is received by a neuron representing a simple receptor that does not perform any treatment. A weighted sum is calculated on the input values. A transfer function is applied to the calculated sum. The sigmoïde function is implemented in the hidden layer.

#### 3.2.3 Classification Phase

We propose a rigorous algorithm for training and recognition. Thus, we use the multilayer perceptron. The proposed classifier is [15, 16, 19] (Figure 5).

Each layer has neurons directly linked to the neurons of the next layer. We find ourselves faced with an optimization model containing changeable variables that describe the problem, together with constraints representing limits on these variables. We define a cost function to minimize is:

$$a_i = \sum_{j=1}^{m} w_{i,j} x_{i,j} + w_{0,i} \text{ for } i = 1, \cdots, n.$$

The inputs preprocessing are used to remove redundant and irrelevant information in order to achieve a small and optimal network structure.

**Algorithm 2:** Training algorithm
Initialize weights $W^{(0)} = (w_{i,0})_{i=1\ldots n}$ such as $w_{i,0} \leqslant 10^{-3}$ for $i = 1 \ldots n$ and $w_{0,i} = 1$.
For $i$ from 1 to $n$ do

1) Present the iutputs $X_i = (x_{i,j})_{j=1\ldots m}$.
2) Calculate $W_i^{(op)}$ and $\epsilon_i$ :
$\epsilon_i = \min_{a_i}(1 - y(a_i))$

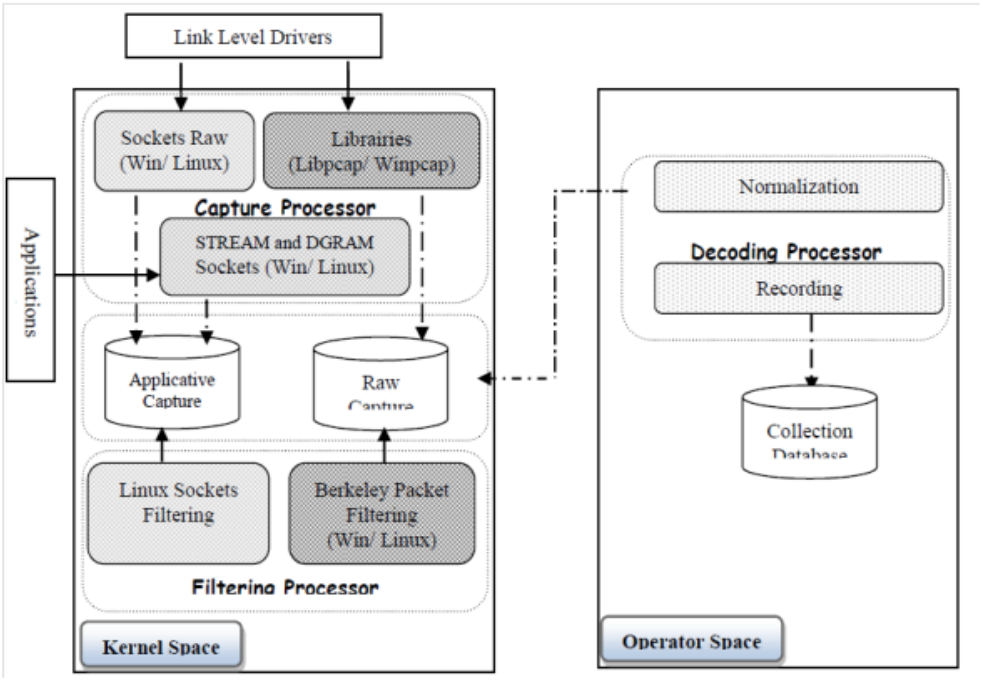


Figure 4: The pcapSockS sniffer

Table 4: Comparison between pcapSockS sniffer, scapy and wireshark

| Sniffer | Platforms | Low capture | High capture | Low filtering | High filtering | Network |
|---|---|---|---|---|---|---|
| **Scapy** | -Win<br>-Mac OS | -Libpcap<br>-Linux | -Libnet | -PBFfilter<br>-Python Functions | -No | -Wired<br>-Wireless |
| **Wireshark** | -Win<br>-Linux | -Libpcap | -No | -PBF Filter | -No | -Wired<br>-Wireless |
| **Pcap.Sock** Sniffer | -Win<br>-Linux | -Libpcap<br>-Raw Sockets | - Sock_Stream<br>-Sock_Dgram | -PBF Filter | -LSF Filter | -Wired |

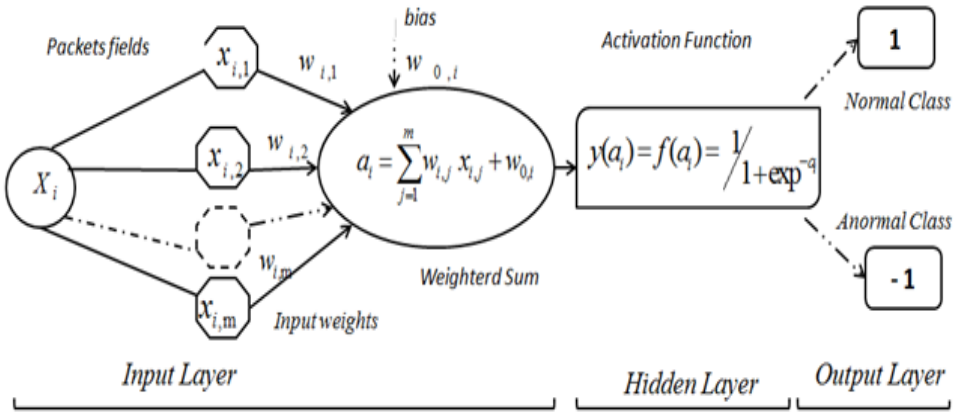


Figure 5: Multilayer perceptron classifier

$$\left\{\begin{array}{l} a_i = \sum_{j=1}^{m} w_{i,j}x_{i,j} + w_{0,i}; \\ y(a_i) = f(a_i); \\ w_{0,i} = w_{0,i} + [1 - y(a_i)]; \\ \text{For } j \text{ from 1 to } m \text{ do} \\ \quad w_{i,j} = w_{i,j-1} + [1 - y(a_i)]x_{i,j}; \\ \text{EndFor} \end{array}\right\}$$

3) EndFor

In the following, we denote with:

$W^{(max)} = (w_j^{(max)})_{j=1\ldots m}$ with $w_j^{(max)} = max\{w_{i,j}^{(op)}, i = 1\ldots n\}$ and $w_0^{(max)} = max\{w_{0,i}^{(op)}, i = 1\ldots n\}$, $a_i^{(max)} = \sum_{j=1}^{m} w_{i,j}^{(max)}x_{i,j} + w_{0,i}^{(max)}$ $a_i^{(op)} = \sum_{j=1}^{m} w_{i,j}^{(op)}x_{i,j} + w_{0,i}^{(op)}$ and $S = \{(X_{i=1}^{n} = (x_{i,j})_{j=1\ldots m}, W_i^{(op)} = (w_{i,j}^{(op)})_{j=1\ldots m}, w_{0,i}^{(op)}, \epsilon_i), i = 1\ldots n\}$ the obtained results during the training phase.

**Proposition 3.1.** *With the above assumptions, we then have for all* $i \in \{1, \ldots, n\}$

*1)* $a_i^{(max)} \geq a_i^{(op)}$.

*2)* $0 < 1 - y(a_i^{(max)}) \leq \epsilon_i$.

*Proof.*

1) As $w_j^{max} = max\{w_{i,j}^{(op)}, i = 1\ldots n\} \geq w_{i,j}^{(op)}$ and $x_{i,j} \geq 0$ for all $i = 1\ldots n$, then $a_i^{(max)} \geqslant a_i^{(op)}$ for all $i \in \{1, \ldots, n\}$.

2) We have $1 - y(a_i^{(op)}) = \epsilon_i$ for each $i \in \{1, \ldots, n\}$ , $y(a_i^{(op)}) = f(a_i^{(op)})$. Therefore $0 \leqslant f(a_i^{(op)}) \leqslant f(a_i^{(max)}) < 1$ for each $i \in \{1, \ldots, n\}$ because $f$ is an increasing function.

   Thereafter $\epsilon_i = 1 - y(a_i^{(op)}) = 1 - f(a_i^{(op)}) \geqslant 1 - f(a_i^{(max)}) = 1 - y(a_i^{(max)}) > 0$

   which shows (2).

□

This phase consists of validating the model: We use for this the optimized weights which are obtained during the training phase.

**Definition 3.1.** *Let* $K = (k_j)_{j=1\ldots m}$ *be an input occurrence and* $a = \sum_{j=1}^{m} w_j^{(max)}k_j + w_0^{(max)}$.

*1) K is a normal occurrence if there exits* $i \in \{1, \ldots, n\}$ *such that* $1 - y(a) \leqslant \epsilon_i$.

*2) K is an intrusion occurrence if for all* $i \in \{1, \ldots, n\}$ *we get* $1 - y(a) > \epsilon_i$.

**Proposition 3.2.** *Let* $K = (k_j)_{j=1\ldots m}$ *be an input occurrence,* $a = \sum_{j=1}^{m} w_j^{(max)}k_j + w_0^{(max)}$ *and* $\epsilon = \{\epsilon_i, i = 1, \ldots, n\}$.
*The following conditions are equivalent:*

*1) K is an intrusion.*

*2)* $1 - y(a) > \epsilon$.

The proof of this proposition relies on Definition 3.1.

**Corollary 3.1.** *: Let* $K = (k_j)_{j=1\ldots m}$ *be an input occurrence,* $a = \sum_{j=1}^{m} w_j^{(max)}k_j + w_0^{(max)}$ *and* $\epsilon = max\{\epsilon_i, i = 1, \ldots, n\}$. *The following conditions are equivalent:*

*1) K is a normal information.*

*2)* $1 - y(a) \leq \epsilon$.

The proof of this corollary relies on Definition 3.1 and Proposition 3.2.

**Algorithm 5:** Recognition algorithm

1) New input $X = (x^{(j)})_{j=1\ldots m}$, final output $d$, activation state $a$, calculated result $y(a)$.

2) Computing of output
   $a = \sum_{j=1}^{m} w_j^{(max)}x^{(j)} + w_0^{(max)}$;
   $y(a) = f(a)$;

3) Classification of activities

$$\left\{\begin{array}{l} if\ (1 - y(a) \leqslant \epsilon)\ then \\ \quad d = 1\ //Normal\ activity \\ \quad else \\ \quad d = -1\ //Intrusion \\ Endif \end{array}\right\}$$

The sigmoid is introduced into the two proposed algorithms for training and recognition. It presents certain constraints during its implementation which leads us to make an evaluation of the sigmoid on platforms more used in practice. In this case, we determine the random values of the weights in $w_i \leqslant 10^{-3}$ to ensure the possible results and avoid the falsified outputs. This modeling leads us to develop an optimal and restricted database containing the occurrences (Table 5):

Table 5: Database structure

| $W^{(max)} = (w_j^{(max)})_{j=0}^{m}$ | $\epsilon$ |
|---|---|

### 3.3 Evaluation Study of Proposed Approach

To accomplish such evaluation, a set of measures are available: precision, number of false positives and false negatives. We take into account the parameters cited in [5,11,14,16,20] such as data sources, intrusion response, frequency of use, real time analysis, intrusion blocking method, real time alert, logging, filtering methods, compatible operating systems (Table 6).

The proposed classifier defines a supervised method using a three layer perceptron and represents a perceptual analysis layer which can be integrated into our new detection system to monitor traffic and make control of the data flow more reliable. With a series of experimental studies on a set of intrusion detection and prevention systems much responded actually. We demonstrated that our new proposal model is more efficient, especially at the level, of data collection, pretreatment of the activities and their classification.

Table 6: Assessments based on proposed parameters

| | **Snort** | **NetASQ** | **Suricata** | **MC Afee Intercept** | **Bro** | **Cisco Net Ranger** | **Our Proposed IDPS** |
|---|---|---|---|---|---|---|---|
| Realtim Analysis | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Type | Nips/ Hips | Nips/ Hips | Nips | Nips | Nips | Nips | Nidps/ Hidps |
| Detection type | Signature | Signature/ Behavioral | Signature/ Behavioral | Signature/ Behavioral | Signature/ Behavioral | Signature | Signature/ Behavioral |
| Blocking | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Reaction | Active | Active | Active | Active | Passive | Active | Active |
| Real time Alerts | Yes | Yes | Yes | Yes/ Email, S NMP | Yes | Yes | Yes |
| Logging | Yes | Yes | Yes | Yes/ MS SQL Server | Yes | Yes | Yes |
| Filtering | Determined by Admin | Filtering and flow contol | Yes | Yes | Yes | Yes | Yes |
| Updating | Manual | Automatic | Manual | — | Automatic / Manual | Automatic | Automatic |
| Operating Systems | Linux/ Windows | Linux/ Windows | OS Commun | Windows/ Solaris | Linux/ Free BSD/ MAC OS | OS Commun | Linux/ Windows |
| Modularity of components | No | No | No | No | No | No | Yes |
| Data collection | Medium | Medium | Medium | Medium | Medium | Medium | High |
| Classification of atvities | Medium | Medium | Medium | Medium | Medium | Medium | High |

## 4 Conclusion

An IDPS tries to detect malicious activities and attacks. It attempts to control computers by monitoring traffic. Many methods have been employed. In this paper, we perform performances evaluation of a list of sniffing and intrusion detection tools and we deduct in the end that those tools suffer much vulnerabilities. So, we propose an optimal approach of intrusion detection based on multilayer perceptron technique aiming to improve the accuracy of detection. The modeled system aims to protect networks from attacks on service integrity, authentication and confidentiality. The preprocessing is performed to transform data evens into a new representation before being presented to a neural network inputs. With implementations carried on different parts, we demonstrate that our system gives the solutions more reliable and relevant to improve the o the network security. Our next work will focus and discuss in detail the various steps of implementation and validation of this global system describing the proposed solutions.

## References

[1] Amrita, K. Ravulakollu, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security (IJNS'18)*, vol. 20, no. 1, pp. 41-55, 2018.

[2] V. Alanou, Détection D'intrusion Dans Un Système Informatique : Méthodes Et Outils, 1996. (`http://www.secdev.org/idsbiblio/lme96methodes.pdf`)

[3] Asrodia, *et al.*, "Analysis of various packet sniffing tools for network monitoring and analysis," *International Journal of Electrical, Electronics and Computer Engineering*, vol. 1, no. 1, pp. 55-58, 2012.

[4] J. Balasubramaniyan, J. O. Garcia-Fernandez, D. Isacoff, E. Spafford, D. Zamboni, "An architecture for intrusion detection using autonomous agents," *COAST Laboratory Purdue University West Lafayette*, 1998. ISBN: 0-8186-8789-4.

[5] M. Boujnouni and M. Jedra, "New intrusion detection system based on support vector domain description with information gain metric," *International Journal of Network Security (IJNS'18)*, vol. 20, no. 1, pp. 25-34.

[6] L. Chappell, "Wirehark 101 essential skills for network analysis," *Protocol Analysis Institute, Inc*, 2013. (https://www.wiresharkbook.com/101v2_samplepages/Wireshark978-1893939752-toc.pdf)

[7] A. Chaudhary, V. N. Tiwari, and A. Kumar, "A New intrusion detection system based on soft computing techniques using neuro-fuzzy classifier for packet dropping attack in MANETs," *International Journal of Network Security*, vol. 18, no. 3, pp. 514-522, 2017.

[8] D. J. Day, B. Burns, "A performance analysis of snort and suricata network intrusion detection and prevention engines," in *The Fifth International Conference on Digital Society*, 2011. (https://www.researchgate.net/publication/241701294_A_Performance_Analysis_of_Snort_and_Suricata_Network_Intrusion_Detection_and_Prevention_Engines)

[9] O. Eldow, P. Chauhan, P. Lalwani, M. Potdar, "Computer network security ids tools and techniques (snort/suricata)," *International Journal of Scientific and Research Publications*, vol. 6, no. 1, pp. 593, 2016.

[10] Y. Farhaoui, A. Asimi, "Creating a complete model of an intrusion detection system effective on the LAN," *International Journal of Advanced Computer Science and Applications (IJACSA'12)*, vol. 3, no. 5, 2012.

[11] Y. Farhaoui and A. Asimi, "Performance method of assessment of the intrusion detection and prevention systems," *International Journal of Engineering Science and Technology (IJEST'11)*, vol. 3, 2011. ISSN: 0975-5462.

[12] Y. Farhaoui, "Design and implementation of an intrusion prevention system," *International Journal of Network Security*, vol. 19, no. 5, pp. 675-683, 2017.

[13] C. Gandhi, G. Suri, R. P. Golyan, P. Saxena, B. K. Saxena, "Packet sniffer – A comparative study," *International Journal of Computer Networks and Communications Security*, vol. 2, no. 5, pp. 179–187, 2014.

[14] A. Guezzaz, *et al.*, "A new hybrid network sniffer model based on Pcap language and sockets (PcapSockS)," *International Journal of Advanced Computer Science and Applications (IJACSA'16)*, vol. 7, no. 2, 2016.

[15] A. Guezzaz, *et al.*, "A novel scheme of an intrusion system using multilayer perceptron," *The Second International Day on Computer Science & Applied Mathematics in Saculty of Sciences and Techniques (ICSAM'17)*, vol. 9, no. 4, 2017.

[16] A. Guezzaz, *et al.*, "A hybrid NIPS based on pcapSockS sniffer and neural MLP," in *International Conference on Information Technology and Communication Systems in National School of Applied Sciences (ITCS'17)*, pp. 253-266, 2017.

[17] A. Guezzaz, *et al.*, "A lightweight neural classifier for behavioral detection," in *International Conference on Information Technology and Communication Systems in Faculty of Sciences and Techniques (IWAM'17)*, vol. 2, no. 2, pp. 57-66, 2017.

[18] A. Gupta, M. Kumar, A. Rangra, V. K. Tiwari, P. Saxena, "Network intrusion detection types and analysis of their tools," *Department of Computer Science and Information Technology, Jaypee University of Information Technology*, vol. 2, no. 1, pp. 63-69, 2012.

[19] A. Guezzaz, *et al.*, "A lightweight neural classifier for intrusion detection," *General Letters in Mathematics*, vol. 2, pp.57-66, 2017.

[20] A. Guezzaz, *et al.*, "A hybrid NIPS based on PcapSockS sniffer and neural MLP," *International Conference on Information Technology and Communication Systems (ITCS'17)*, pp. 253-266, 2017.

[21] A. Halaweh,"A taxonomy of free network snifers for teaching and research," *Journal of Computing Sciences in Colleges*, vol. 21, no. 1, pp. 64-75, 2005.

[22] B. Ma, "Using packet sniffing to teach networking concepts," *Journal of Computing Sciences in Colleges*, vol. 30, no. 6, pp. 67-74, 2015.

[23] N. Malik, "Artificial neural networks and their applications," *National Conference on Unearthing Technological Developments & their Transfer for Serving Masses GLA ITM*, 2005. (https://www.researchgate.net/publication/1958135_Artificial_Neural_Networks_and_their_Applications)

[24] L. Moulad, H. Belhadaoui, M. Rifi, "Implementation of a hierarchical hybrid intrusion detection mechanism in wireless sensors network", *International Journal of Advanced Computer Science and Applications (IJACSA'17)*, vol. 8, no. 10, 2017.

[25] M. Ridho, F. Yasin, M. Eng, "Analysis and evaluation snort, bro, and suricata as intrusion detection system based on linux server," *Department of Informatics, Faculty of Communications and Informatics Universitas Muhammadiyah Surakarta*, 2014. (http://etd.eprints.ums.ac.id/31281/)

[26] M. Rochaa, *et al.*, "Evolution of neural networks for classification and regression," *Neurocomputing*, vol. 70, no. 16-18, pp. 2809-2816, 2007.

[27] Rupam, A. Verma, A. Singh, "An approach to detect packets using packet sniffing," *International Journal of Computer Science & Engineering Survey (IJCSES'13)*, vol. 4, no. 3, 2013.

[28] B. Santos, T. Chandra, M. Ratnakar, S. Baba, N. Sudhakar, "Intrusion detection system types and prevention," *International Journal of Computer Science and Information Technologies*, vol. 4, no. 1, 2013.

[29] Schiffmann, *et al.*, *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*, 1994. (`http://www.cs.bham.ac.uk/~pxt/NC/schiffmann.bp.pdf`)

## Biography

**Guezzaz Azidine** received his Ph.D in design and validation of an intrusion detection and prevention system based on neural network for cloud computing. His research interest is Intrusion Detection and Prevention and Computer and Network Security and Cryptography. He is an assistant professor at the Technology High School Essaouira Cadi Ayyad University Marrakech, Morocco.

**ASIMI Ahmed** received his PhD degree in Number theory from the University Mohammed V – Agdal in 2001. He is reviewer at the International Journal of Network Security (IJNS). His research interest includes Number theory, Code theory, and Computer Cryptology and Security. He is a full professor at the Faculty of Science at Agadir Morocco since 2008.

**Younes Asimi** received his Ph.D. in Strong Zero-Knowledge Authentication Based on virtual passwords per session and the Session Keys in 2015. His research interests include Authentication Protocols, Computer and Network Security and Cryptography. He is an assistant professor at the Technology High School Guelmim, Ibn-Zohr University Agadir, Morocco.

**Tbatou Zakariae** received his Ph.D degree in. He is currently Ph.D student in Authentication Protocols Kerberos for distributed systems. His research interests include Authentication Protocols, distributed systems, cloud computing, Computer and Network Security and Cryptography.

**SADQI Yassine** received his Ph.D degree in the field of Computer Science and Distributed Systems at Ibn Zoher University in 2012. His research interest is Web Applications Security, Computer Security and Cryptography. He is an assistant professor at Polydisciplinairy Faculty,Sultan Moulay Slimane University, Beni Mellal, Morocco.