Open Access Library Journal
2021, Volume 8, e8000
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Implementation of Machine Learning Method for the Detection and Prevention of Attack in Supervised Network

Patrick Dany Bavoua Kenfack[1,2], Fabrice Kwefeu Mbakop[3], Edward Eyong-Ebai[4]

[1]Department of Electrical and Telecommunications Engineering, National Advanced Polytechnic School of Yaounde, University of Yaounde I, Yaounde, Cameroon
[2]Electrical Engineering, Mechatronics and Signal Processing Laboratory of University of Yaounde I, Yaounde, Cameroon
[3]Department of Renewable Energy, National Advanced Polytechnic School, University of Maroua, Maroua, Cameroon
[4]School of Posts, Telecommunications and ICT, Yaounde, Cameroon
Email: danybavoua@gmail.com

How to cite this paper: Kenfack, P.D.B., Mbakop, F.K. and Eyong-Ebai, E. (2021) Implementation of Machine Learning Method for the Detection and Prevention of Attack in Supervised Network. *Open Access Library Journal*, **8**: e8000.
https://doi.org/10.4236/oalib.1108000

Received: September 24, 2021
Accepted: December 27, 2021
Published: December 30, 2021

Abstract

The sustainability of a company depends on the permanent availability of its information system. This reality influences the behavior of companies, which are becoming increasingly mature in their investments in information system security, which is an absolutely vital element. The use of a service called "SYSLOG" to centralize the network event logs that are sent by printers, servers, routers, firewalls, IDS and IPS in an SYSLOG server is a perfect example for network optimization. In this work, which consists in setting up a Machine learning algorithm for detection and prevention of attacks, we are interested on one hand with the problems encountered on the SYSLOG service and on the other hand with the problems encountered during the detection and prevention of anomalies in the SYSLOG service. In order to ensure an optimal level of security within the network according to the criteria specified, we will first proceed to an analysis of the log files present in the server; followed by an attack detection based on an automatic machine learning algorithm using the signature and historical behavior of the different attacks. As result, we have the possibility to generate real-time alerts on malfunctions; real-time monitoring of the use of an application (number of users, functions used, etc.); the identification of the origin of incidents occurring in applications.

Subject Areas

Computer and Network Security, Information and Communication: Security, Privacy, Trust, Information and Communication Theory and Algorithms

Keywords

Big Data, Log Files, Machine Learning, Network Security, Supervised Network

1. Introduction

With the rapid development of Information and Communication Technologies (ICT) and the growth of the internet, web servers can easily be attacked because of their great value. Therefore, web security is of great importance. Detection of abnormalities plays an important role in web security and log file recordings of detailed system runtime information have become a very important object of data analysis [1]. Traditional log detection for abnormalities relies on programmers for manual inspection by keyword search and by correspondence of regular expressions. Although programmers can use an intrusion detection system to reduce their workload, the data in the logging system is large, the types of attacks vary widely and hacking skills improve, making traditional detection ineffective. To improve traditional detection technology, numerous detection mechanisms have been proposed in recent years, including the machine learning method.

Centralizing log files improves security in an information system however, in order to minimize risks and limit impacts, it is important to meet other major needs:

- **Storage capacity requirements:** IT Network infrastructure is increasingly large, log files are therefore becoming increasingly heavy and consequently require a large internal storage capacity.
- **Log file automation needs:** The task of analyzing log files remains an essentially manual activity, it is the administrator who must each time launch the analysis software and manage these files. So we can make things a lot easier by generating these log files on an ongoing basis.
- **Needs to improve variety:** The log files are not structured; a simple search reveals the impressive number of shapes in the log files. These formats, often incompatible and different, all need to be analyzed.
- **Needs to improve accuracy:** Ensure efficient file tracking generated by a given source.

In this context, we are entrusted with the following work: **Implementation of a machine learning algorithm for the detection and prevention of attacks in supervised network.**

The following section presents fundamental concepts for this work. In Section 2, we present the different detections method of intrusion. Section 3 is devoted to the proposed method using machine learning for detection of intrusion. In Section 4, the results and discussions are presented. Conclusion appears in Section 5.

2. Fundamental Concepts

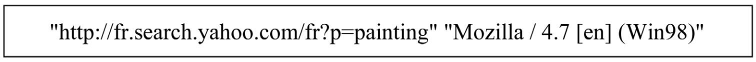
2.1. Log Files

Each action of a computer system (opening a session, installing a program, browsing the Internet etc.) produces a log file. These text files are listed chronologically as events are executed by a server or a computer application. They are useful for understanding the source of an error in the event of a bug. In the case of a web server, a log file will record the date and time of the access attempt, the client's IP address, the target file, and the operating system used, the browser, the response from the server to this request, possibly the type of error encountered, etc. [1].

Computer systems generate many types of log files in order to provide essential information about the system. Among these types, the following examples can be mentioned:

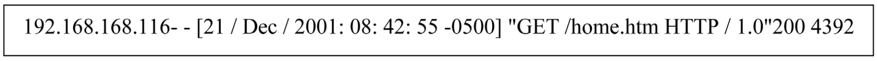
- Log files from servers,
- Log files from websites,
- Log files from intrusion detection systems,
- Log files from network monitoring systems,
- Log files from firewalls,
- Access log files,
- Error log files.

The structure and content of the log file allows further information to be obtained after certain processes. There are two types of format CLF (Common Log File) and ELF (Extended Log Format), the latter is the most common [2].



```
"http://fr.search.yahoo.com/fr?p=painting" "Mozilla / 4.7 [en] (Win98)"
```

In this part, we have explained these two formats through two examples. 161.31.132.116--[21/Dec/2001: 08:42:55-0500] "GET/home.htm HTTP/1.0" 200 4392



```
192.168.168.116- - [21 / Dec / 2001: 08: 42: 55 -0500] "GET /home.htm HTTP / 1.0"200 4392
```

The second format shows an example of a CLF log file format, this format is composed, for each user request by the following fields:

- 192.168.168.116: The IP address of the user who sent the request.
- [21/Dec/2001: 08:42:55-0500]: The date is the time of the request.
- GET/home.htm HTTP/1.0: The request method, the page requested and the protocol used.
- 200: The server response code number.
- 4392: The size of the requested page in bytes.

The ELF format has the same fields as the CLF format, adding other parameters for more details, such as:

- http://fr.search.yahoo.com/fr?p=painting: The reference page from which the request is launched.
- Mozilla/4.7 [en] (Win98): The browser and operating system used by the user.

In the example above, the machine with the IP address **192.168.168.116** issued the **GET/home.htm HTTP/1.0** request on **December 21, 2001 at 8:42:55 A.M**, His request was accepted by the server (code **200** means acceptance) and the machine received a file of size **4392 bytes**. The reference page from which the machine launches the request is **http://fr.search.yahoo.com/fr?p=painting** using the **Mozilla** browser under **English** version **4.7** of **Windows 98** environment.

2.2. Machine Learning

Machine learning (ML) is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

Architecture of Machine Learning

The diagram below focuses on client-server architecture of a "supervised learning system [3].

As we can see in Figure 1, Most Machine learning architectures have the following components:

- **Ground-truth collector:** In the real world, it is key to be able to continuously acquire new data for the machine to learn from. One type of data is particularly important: ground-truth data. This corresponds to what you want your ML models to predict, such as detection of abnormalities in log files.
- **Data labeler:** Sometimes, you'll have access to plenty of input data, but you'll need to create the associated ground-truth data manually. This is the case when building a spam detector, or an object detector from images. There are ready-made and open-source web apps to make data labeling easier (such as Label Studio), and dedicated services for outsourcing the manual task of labeling data
- **Evaluator:** When you have an initial dataset for the machine to learn from, it's important to define how to evaluate the planned ML system, before setting out to build any ML models. In addition to measuring prediction accuracy, evaluation of short-term and long-term impact via application-specific performance metrics, and system metrics such as lag and throughput, are desirable.
- **Performance monitor:** The next step towards deciding if a (baseline) model can be integrated into an application is to use it on the inputs encountered in production (called "production data"), in a production-like setting, and to monitor its performance through time.

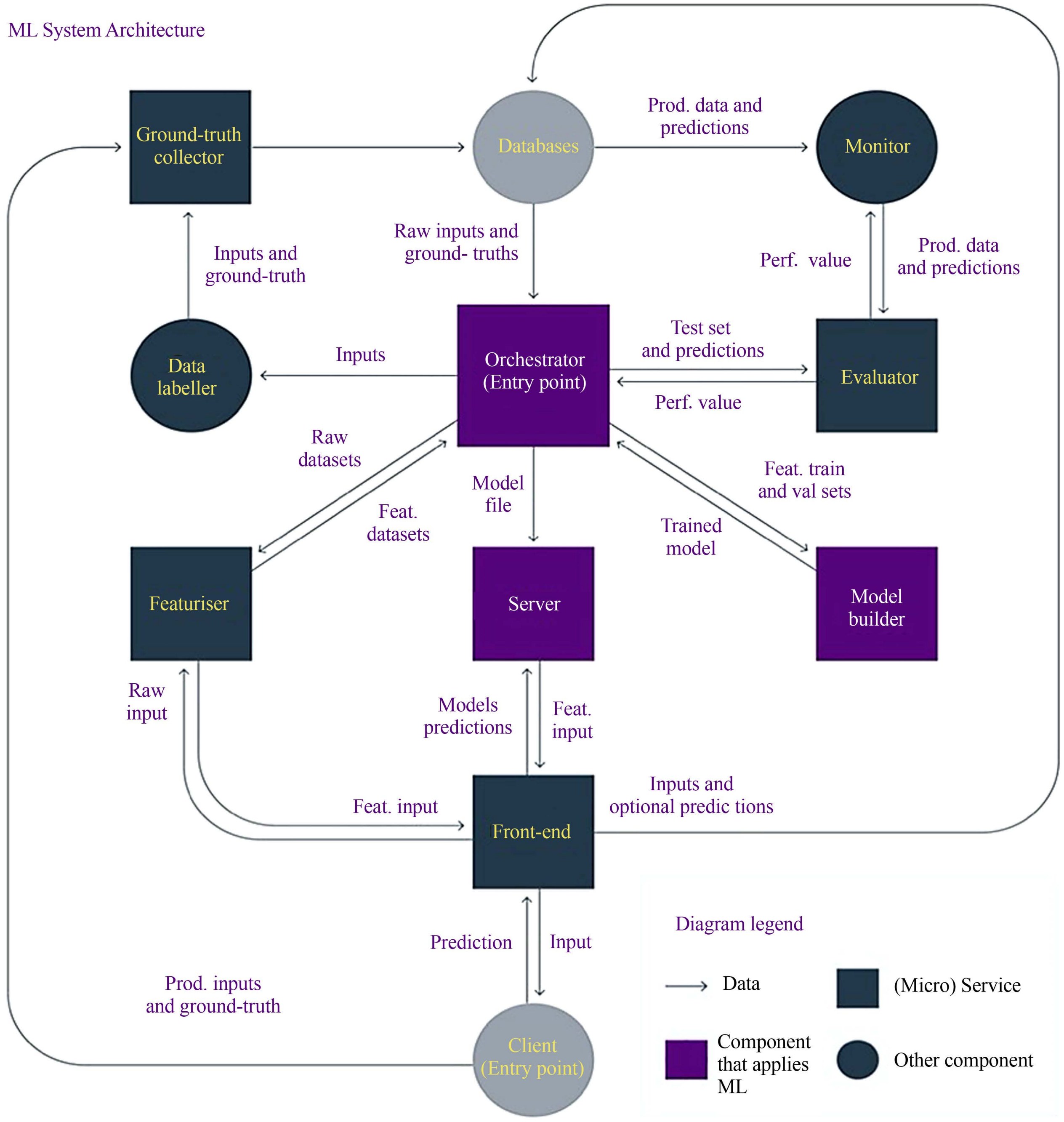


Figure 1. Machine learning architecture system [3].

- **Featurizer:** When designing a prediction API, a decision needs to be made as to what the API should take as input. For example, when making predictions about customers, should the input be the full feature representation of the customer, or just the customer id?
- **Orchestrator:** The orchestrator is at the core of the ML system and interacts with many other components. It follows a series of steps to run the workflow.
- **Model builder:** The model builder is in charge of providing an optimal model. For this, it trains various models on the training set and evaluates

them on the validation set, with the given metric, in order to assess optimality.

- **Model Server:** The role of a model server is to process API requests for predictions against a given model. For this, it loads a model representation saved in a file and applies it, thanks to a model interpreter, to the inputs found in the API request; predictions are then returned in the API response.
- **Front-End:** The front-end can serve multiple purposes such simplify the model's output, for instance by turning a list of class probabilities into the most likely class and also add to the model's output, for instance by using a black box model explainer and providing a prediction explanation (in the same way as Indico does).

Some machine learning methods

Machine learning algorithms are often categorized as supervised or unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- **In contrast, unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it/learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.
- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

2.3. Big Data

Big data refers to all the digital data produced by the use of new technologies for personal or professional purposes. This includes company data (emails, documents, databases, history of business processors, etc.) as well as data from sensors, content published on the web (images, videos, sounds, texts), e-commerce transactions, exchanges on social networks, data transmitted by connected objects (electronic tag, smart meters, smartphones, etc.), geolocated data, etc. It is therefore a set of technologies, architecture, tools and heterogeneous and changing quantities and content, and to extract the relevant information at an accessible cost.

To describe the principle of Big Data, it is customary to summarize its major characteristics [4]:

- The volume of data to collect, store and process;
- The speed at which data is produced and evolves in the time;
- Variety of data;
- Veracity or reliability of the data;
- Value representing the importance of the data.

Big data plays a very important role within the companies that use it. Big data comes to solve the problems of slow request processing; the storage is much more robust and allows mass storage of external or internal company data.

2.4. Most Common Types of Cyber Attacks

Man-in-the-middle (MitM) attack

A MitM attack occurs when a hacker inserts itself between the communications of a client and a server. Here are some common types of man-in-the-middle attacks: Session hijacking, IP Spoofing.

Phishing and spear phishing attacks

Phishing attack is the practice of sending emails that appear to be from trusted sources with the goal of gaining personal information or influencing users to do something. It combines social engineering and technical trickery. It could involve an attachment to an email that loads malware onto your computer. It could also be a link to an illegitimate website that can trick you into downloading malware or handing over your personal information.

Drive-by attack

Drive-by download attacks are a common method of spreading malware. Hackers look for insecure websites and plant a malicious script into HTTP or PHP code on one of the pages. This script might install malware directly onto the computer of someone who visits the site, or it might re-direct the victim to a site controlled by the hackers. Drive-by downloads can happen when visiting a website or viewing an email message or a pop-up window. Unlike many other types of cyber security attacks, a drive-by doesn't rely on a user to do anything to actively enable the attack; you don't have to click a download button or open a malicious email attachment to become infected. A drive-by download can take advantage of an app, operating system or web browser that contains security flaws due to unsuccessful updates or lack of updates.

Cross-site scripting (XSS) attack

XSS attacks use third-party web resources to run scripts in the victim's web browser or scriptable application. Specifically, the attacker injects a payload with malicious JavaScript into a website's database. When the victim requests a page from the website, the website transmits the page, with the attacker's payload as part of the HTML body, to the victim's browser, which executes the malicious script. For example, it might send the victim's cookie to the attacker's server, and the attacker can extract it and use it for session hijacking. The most dangerous consequences occur when XSS is used to exploit additional vulnerabilities. These vulnerabilities can enable an attacker to not only steal cookies, but also log key strokes, capture screenshots, discover and collect network information, and remotely access and control the victim's machine [5] [6].

Denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks

A denial-of-service attack overwhelms a system's resources so that it cannot respond to service requests [5] [6]. A DDoS attack is also an attack on system's resources, but it is launched from a large number of other host machines that are infected by malicious software controlled by the attacker. Unlike attacks that are designed to enable the attacker to gain or increase access, denial-of-service doesn't provide direct benefits for attackers. For some of them, it's enough to have the satisfaction of service denial. However, if the attacked resource belongs to a business competitor, then the benefit to the attacker may be real enough. Another purpose of a DoS attack can be to take a system offline so that a different kind of attack can be launched. One common example is session hijacking, which I'll describe later (**Figure 2**).

Distributed Denial of Service (DDoS) [5] when the attack involves a network of machines (often compromised) to interrupt the desired services (**Figure 3**).

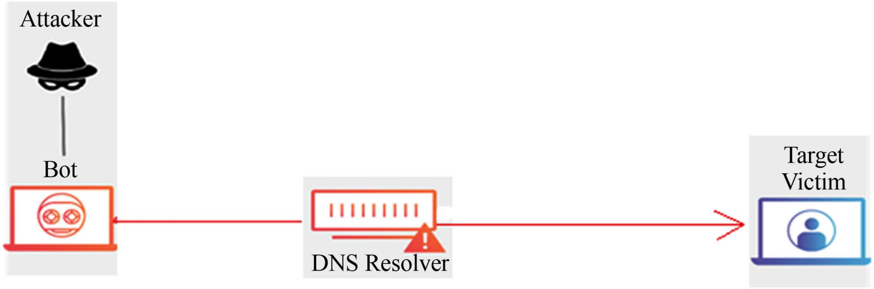


Figure 2. Denial of service attack [5].

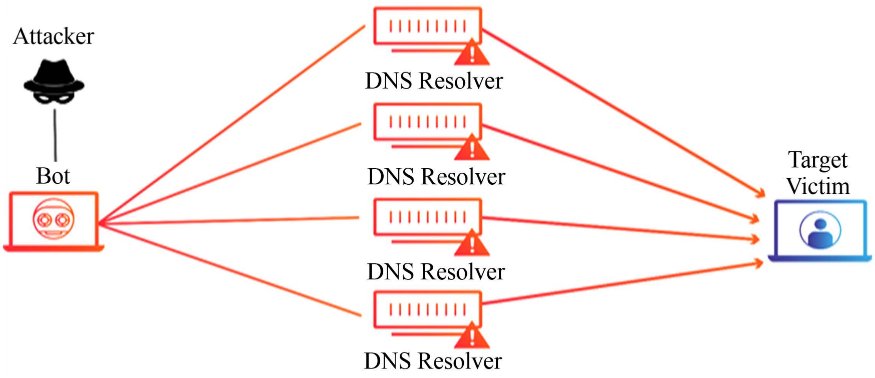


Figure 3. Distributed denial of service [5].

3. State of the Art of Detections Methods

Recently, as the damage caused by denial-of-service attacks has increased, it has become one of the most important security issues to be addressed. Many methodologies and tools have appeared in society to detect DoS attacks and reduce their damage. However, each method has limited success, as most of them cannot simultaneously achieve the expected goals which are detection of errors with a few alarms and ensuring a real-time transfer of data of all packages.

3.1. Detections of DDos

SYN flood case

Figure 4 presents *Wireshark tool*.

As noted above, the term SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target system. The detection of this attack can be done manually via sniffing tools like Wireshark. Administrators should be able to note the start of the attack by a **huge stream of TCP traffic.** We can **filter SYN packages** without delivery confirmation using the following filter: **tcp.flags.syn == 1 et tcp.flags.ack == 0**.

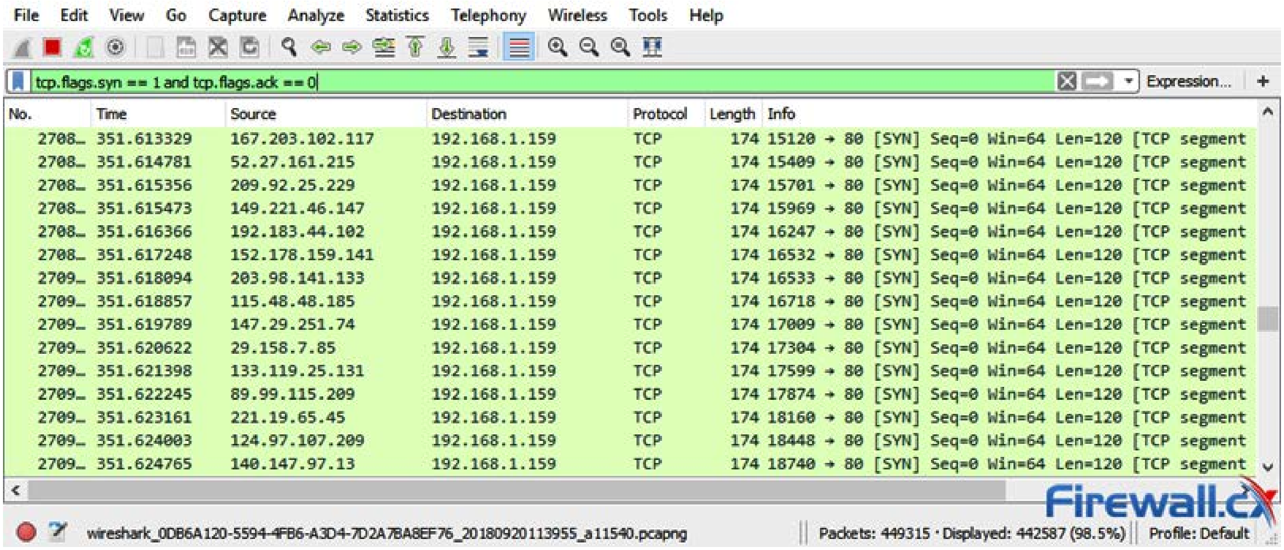
As we can see, the SYN package **v**olume is very **high and varies** very little over time (Figure 5).

Each SYN package indicates that it comes from a different source IP **address** with a **destination port 80** (HTTP), an identical length of **120** and a **window size** (64). In addition, if we **filter** with **tcp.flags.syn == 1 et tcp.flags.ack == 1,** we can see that the number of **SYN/ACK** is comparatively very low. A sure sign of a TCP SYN attack.

It is also possible to view **Wireshark's charts** for a **visual representation** of the increase in traffic.

The graph above shows a massive **increase in** the total number of packets ranging from 0 to **2400 second packs.** All of these metrics indicate an **SYN flood attack** with little room for interpretation (Figure 6).

In this model, the attacker aims to cause the system to shut down. Generally, in a network, several different IP addresses are connected to specific ports when communicating between entities as shown in Figure 7.



tcp.flags.syn == 1 and tcp.flags.ack == 0

No.	Time	Source	Destination	Protocol	Length	Info
2708…	351.613329	167.203.102.117	192.168.1.159	TCP	174	15120 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708…	351.614781	52.27.161.215	192.168.1.159	TCP	174	15409 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708…	351.615356	209.92.25.229	192.168.1.159	TCP	174	15701 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708…	351.615473	149.221.46.147	192.168.1.159	TCP	174	15969 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708…	351.616366	192.183.44.102	192.168.1.159	TCP	174	16247 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2708…	351.617248	152.178.159.141	192.168.1.159	TCP	174	16532 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.618094	203.98.141.133	192.168.1.159	TCP	174	16533 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.618857	115.48.48.185	192.168.1.159	TCP	174	16718 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.619789	147.29.251.74	192.168.1.159	TCP	174	17009 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.620622	29.158.7.85	192.168.1.159	TCP	174	17304 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.621398	133.119.25.131	192.168.1.159	TCP	174	17599 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.622245	89.99.115.209	192.168.1.159	TCP	174	17874 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.623161	221.19.65.45	192.168.1.159	TCP	174	18160 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.624003	124.97.107.209	192.168.1.159	TCP	174	18448 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment
2709…	351.624765	140.147.97.13	192.168.1.159	TCP	174	18740 → 80 [SYN] Seq=0 Win=64 Len=120 [TCP segment

wireshark_0DB6A120-5594-4FB6-A3D4-7D2A7BA8EF76_20180920113955_a11540.pcapng | Packets: 449315 · Displayed: 442587 (98.5%) | Profile: Default

Figure 4. Packet sniffing with Wireshark.

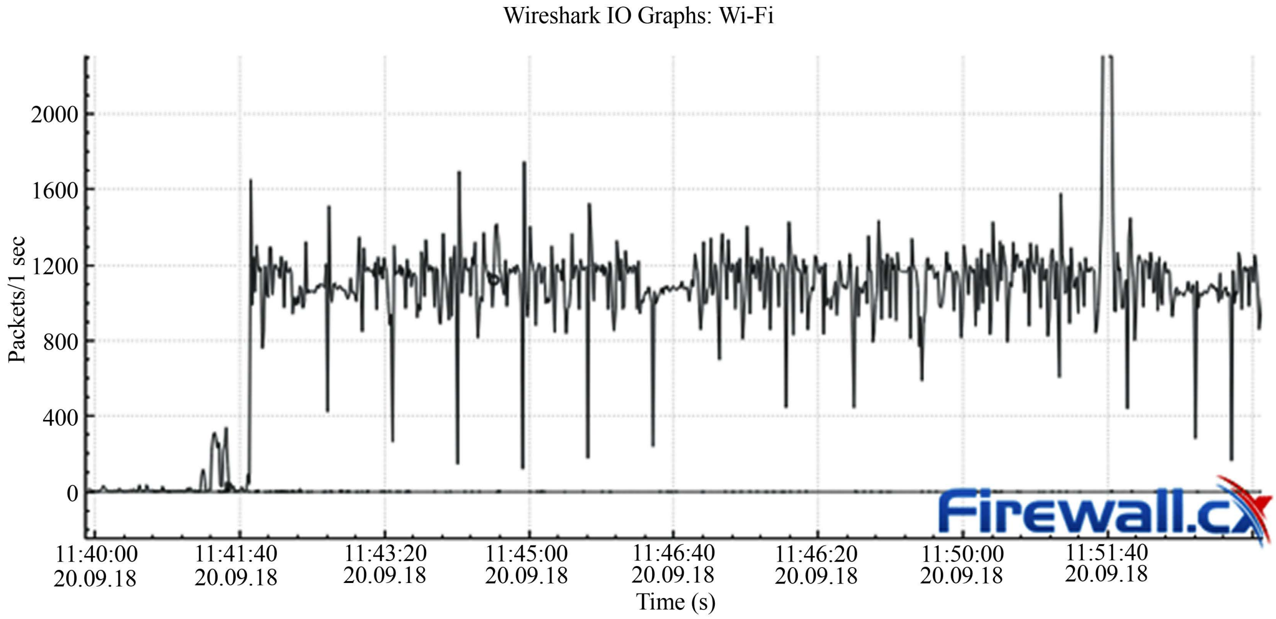
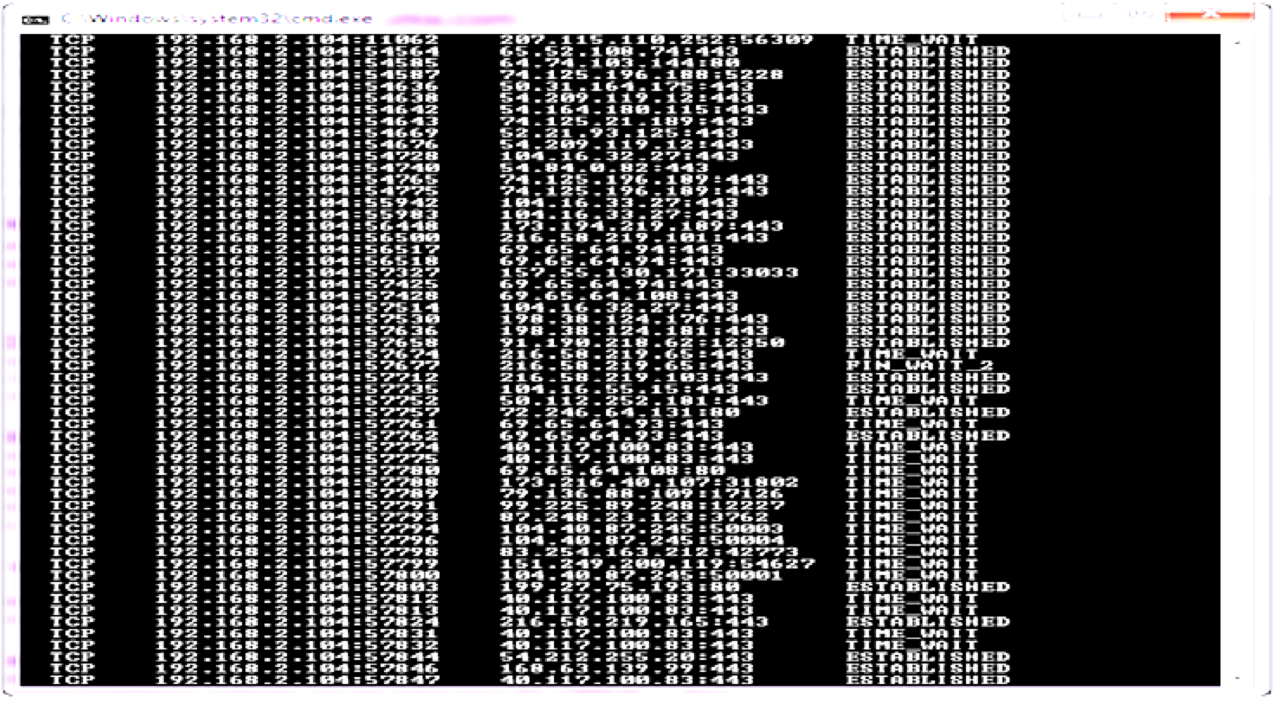


Figure 5. Network traffic graph with Wireshark.

In the case of denial-of-service attacks with a ping of death, it is very often noticed that the same IP address connects to adjoining ports and that the connection is coming to an end.

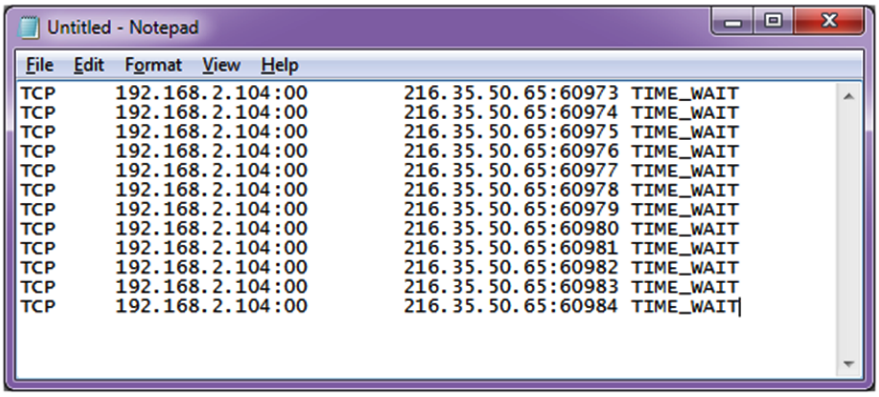
We only show a handful, but a real DDoS attack should show hundreds of connections (sometimes thousands) as seen above.

If any of the above cases arise, an alarm should be triggered. But detecting anomalies isn't as simple as attackers often make packages to mimic real-world user transactions. Therefore, detection tools must be based on algorithms, mathematical statistics and machine learning. This works well for application-based attacks and protocol attacks.



```
  TCP    192.168.2.104:11062    207.115.110.252:56309  TIME_WAIT
  TCP    192.168.2.104:54564    65.52.108.74:443       ESTABLISHED
  TCP    192.168.2.104:54585    64.74.103.144:80       ESTABLISHED
  TCP    192.168.2.104:54587    74.125.196.188:5228    ESTABLISHED
  TCP    192.168.2.104:54636    50.31.164.175:443      ESTABLISHED
  TCP    192.168.2.104:54638    54.209.119.12:443      ESTABLISHED
  TCP    192.168.2.104:54642    54.164.180.115:443     ESTABLISHED
  TCP    192.168.2.104:54643    74.125.21.189:443      ESTABLISHED
  TCP    192.168.2.104:54667    52.21.93.125:443       ESTABLISHED
  TCP    192.168.2.104:54676    54.209.119.12:443      ESTABLISHED
  TCP    192.168.2.104:54728    104.16.32.27:443       ESTABLISHED
  TCP    192.168.2.104:54740    54.84.0.82:443         ESTABLISHED
  TCP    192.168.2.104:54765    74.125.196.189:443     ESTABLISHED
  TCP    192.168.2.104:54775    74.125.196.189:443     ESTABLISHED
  TCP    192.168.2.104:55942    104.16.33.27:443       ESTABLISHED
  TCP    192.168.2.104:55903    104.16.33.27:443       ESTABLISHED
  TCP    192.168.2.104:56448    173.194.219.189:443    ESTABLISHED
  TCP    192.168.2.104:56500    216.58.219.101:443     ESTABLISHED
  TCP    192.168.2.104:56517    69.65.64.94:443        ESTABLISHED
  TCP    192.168.2.104:56518    69.65.64.94:443        ESTABLISHED
  TCP    192.168.2.104:57327    157.55.130.171:33033   ESTABLISHED
  TCP    192.168.2.104:57425    69.65.64.94:443        ESTABLISHED
  TCP    192.168.2.104:57428    69.65.64.108:443       ESTABLISHED
  TCP    192.168.2.104:57514    104.16.32.27:443       ESTABLISHED
  TCP    192.168.2.104:57530    198.38.124.176:443     ESTABLISHED
  TCP    192.168.2.104:57636    198.38.124.181:443     ESTABLISHED
  TCP    192.168.2.104:57658    91.190.218.62:12350    ESTABLISHED
  TCP    192.168.2.104:57674    216.58.219.65:443      TIME_WAIT
  TCP    192.168.2.104:57677    216.58.219.65:443      FIN_WAIT_2
  TCP    192.168.2.104:57712    216.58.219.103:443     ESTABLISHED
  TCP    192.168.2.104:57735    104.16.55.15:443       ESTABLISHED
  TCP    192.168.2.104:57752    50.112.252.101:443     TIME_WAIT
  TCP    192.168.2.104:57757    72.246.64.131:80       ESTABLISHED
  TCP    192.168.2.104:57761    69.65.64.93:443        TIME_WAIT
  TCP    192.168.2.104:57762    69.65.64.93:443        ESTABLISHED
  TCP    192.168.2.104:57774    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57775    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57780    69.65.64.108:80        TIME_WAIT
  TCP    192.168.2.104:57788    173.216.40.107:31802   TIME_WAIT
  TCP    192.168.2.104:57789    79.136.88.109:17126    TIME_WAIT
  TCP    192.168.2.104:57791    99.225.89.248:12227    TIME_WAIT
  TCP    192.168.2.104:57793    87.248.23.123:3762     TIME_WAIT
  TCP    192.168.2.104:57794    104.40.87.245:50003    TIME_WAIT
  TCP    192.168.2.104:57796    104.40.87.245:50004    TIME_WAIT
  TCP    192.168.2.104:57798    83.254.163.212:42773   TIME_WAIT
  TCP    192.168.2.104:57799    151.249.200.119:54627  TIME_WAIT
  TCP    192.168.2.104:57800    104.40.87.245:50001    TIME_WAIT
  TCP    192.168.2.104:57803    199.27.75.193:80       ESTABLISHED
  TCP    192.168.2.104:57812    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57813    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57824    216.58.219.165:443     ESTABLISHED
  TCP    192.168.2.104:57831    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57832    40.117.100.83:443      TIME_WAIT
  TCP    192.168.2.104:57844    54.212.255.20:443      ESTABLISHED
  TCP    192.168.2.104:57846    160.63.139.99:443      ESTABLISHED
  TCP    192.168.2.104:57847    40.117.100.83:443      ESTABLISHED
```

Figure 6. Normal traffic within a network.



```
TCP       192.168.2.104:00          216.35.50.65:60973 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60974 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60975 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60976 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60977 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60978 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60979 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60980 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60981 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60982 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60983 TIME_WAIT
TCP       192.168.2.104:00          216.35.50.65:60984 TIME_WAIT
```

Figure 7. Ping of death running.

3.2. Machine Learning Detection

The "classic" Machine Learning aims to give a machine the ability to learn how to solve a problem without having to explicitly program each rule. The idea of Machine Learning is therefore to solve problems by modeling behaviors through data-based learning.

However, before you can model a problem through a Machine Learning algorithm, it is often necessary to make a good number of transformations on the data. These transformations, which are done manually, are dictated by the type of problem that one seeks to solve, and by the choice of the algorithm used. This data processing (commonly known as feature engineering) is often very time-consuming and may require expertise in order to be relevant [7] [8] (**Figure 8**).

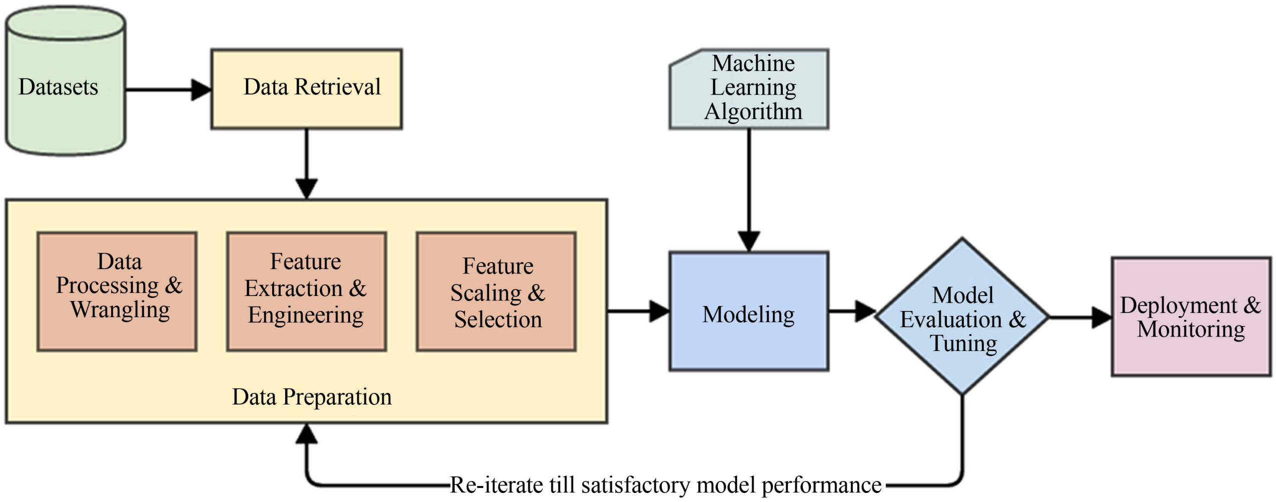

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Figure 8. Solution's operating architecture [6].

Most of the time, in a Machine learning, *Data Sets* come in different orders of magnitude. This difference in scale can lead to lower performance. To compensate for this, preparatory treatments on the data exist. This data processing goes through several steps:

- **Data Wrangling**, also known as Self-Service Data Preparation, is the process that allows raw data to discover, structure, clean, enrich, validate and publish results in a format suitable for data analysis.
- **Feature extraction and engineering** refers to all interventions performed on raw data before they are taken into account by a learner algorithm.

After these steps, we estimated a model based on data, called observations, that are available and in finite numbers, during the design phase of the system.

Once the machine learning model is in place, to predict events we need to apply a proper machine learning algorithm to train these models and detect classes of unknown instances.

Decision Tree Algorithm

A decision tree is a tree in which each node represents a characteristic (attribute), each link (branch) represents a decision (rule) and each sheet represents a result (categorical or continuous value). The idea is to create such a tree for all the data and to process a unique result with each sheet (or minimize the error in each sheet) (**Figure 9**).

4. Machine Learning Solution Using Centralization of Log Files

4.1. Method and Architecture

To counter the existing threats, having an overview of all the equipment and carrying out processing such as: research or statistics, intrusion detection, crash diagnosis; we need to use a centralized log system process. The figure below shows the centralization of log files of the system.

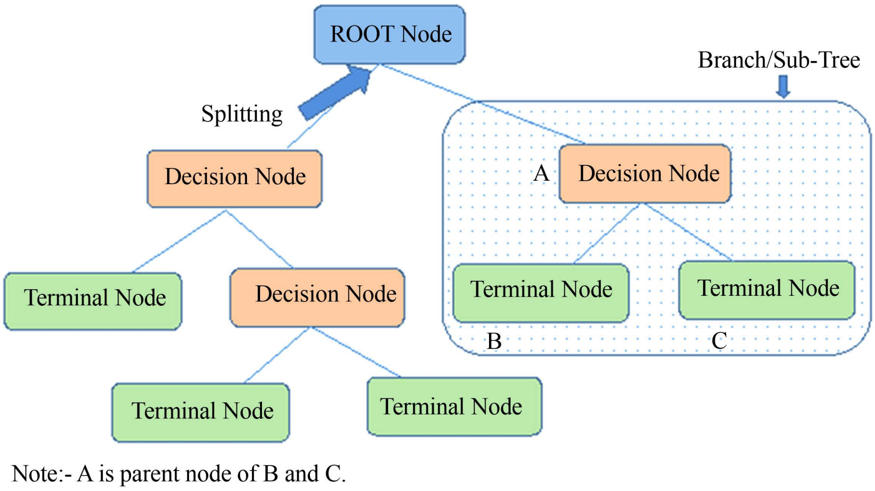


Figure 9. Decision tree algorithm [7].

The centralization of logs consists of placing all the logs of the systems, applications and services of the surrounding machines on a single system, a single platform.

This process thus allows us:

- To have an overview of the crucial elements for good management of the information system to carry out processing such as:

○ **IDS:** Some types use event logs to detect abnormal behavior on the network or on monitored systems. They therefore do not use the network surveillance to carry out their monitoring but rather the logs, these IDS are then called log-based IDS. Known examples of such systems are Fail2ban and Deny Hosts.

○ **Forensic, more broadly attack or intrusion analysis;** we can describe forensics as follows: "it is a material or virtual investigation method based on the extraction of raw or altered data to construct an event chronology, to establish a sequential logic or to reconstitute a coherent set of data in order to derive a contextual explanation."

Logs prove to be of great help when we want to trace the course, actions and damage of an attacker on a variety of systems. These are facilitated firstly because the logs are centralized, but also because they are exported from the attacker's area of effect, which then has more difficulty removing them to erase his tracks.

○ **Research/statistics:** The centralization of logs will make it possible to carry out very precise research on the activity of several systems while being on one machine. This makes research and statistics easier, because everything is on the same platform.

- In case of crash or deletion of logs

○ **Diagnosing a crash:** It can be very useful to know precisely what happened when a system becomes completely unreachable (after a severe crash for example). If the logs which make it possible to diagnose the failure are found on the said machine, they are difficult to reach (you can always extract the disks, read them on another device, etc.). If the logs are exported to a machine with guaranteed availability, it is then possible to quickly recover the latest system events from the damaged machine for easier diagnosis.

- **Guarantee the survival of logs after deletion:** One of the post-intrusion steps during a computer attack is the erasure of traces that the hacker may have left. We then go, in part, by deleting logs and histories which can give indications on the actions carried out during the intrusion and possibly on a network identity (IP, MAC address, OS, etc.). When the logs are stored locally, the logs that are deleted are then difficult to recover, when they are also sent to another machine, it is possible to recover information. Note that the deletion, or loss, of logs can also happen in other contexts closer to the daily life of an IS such as the loss of a disk, of a partition dedicated to logs, the improper handling of a system's information.

Thus, to our network solution, we will attach a log analysis and correlation solution as in Figure 10.

4.2. UML Design

We will now present our design. We will use UML as a modeling language.

UML meaning Unified Modeling Language: It is a graphic language of data and processing modeling, it is the accomplishment of the fusion of previous object modeling languages: Booch, OOSE, OMT, it is a standard defined by the OMG (Object Management Group).

The IT sector has adopted UML as a pivotal language, very comprehensive, object-oriented and independent of programming languages, it is rich and open. We have adopted the UML language as a modeling language for our application, it is not an arbitrary choice, but this is actually due to several reasons:

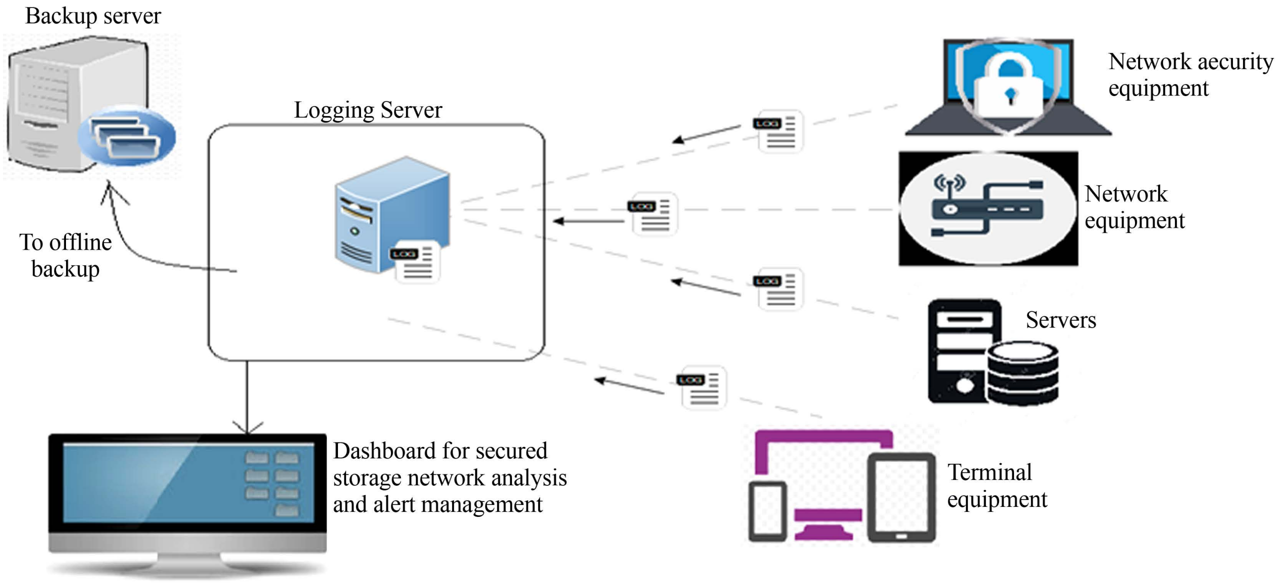

Backup server
Logging Server
To offline backup
Network aecurity equipment
Network equipment
Servers
Dashboard for secured storage network analysis and alert management
Terminal equipment
LOG

Figure 10. Architecture to be implemented.

- UML is based on the mechanisms of abstraction, hierarchy and decomposition.
- It considers a system as an organized whole, the elements of which can only be defined in relation to each other.
- Object-oriented modeling with UML makes it easy to reuse applications.
- We aim to achieve the application with the Python language which is an object-oriented language.

UML allows systems to be modelled in the form of a collection of objects. UML has several types of models called diagrams (13 in all), each diagram representing a separate view of the system. It is of course not necessary to use all the diagrams to model a system, it is necessary to choose from all the diagrams available those that are most relevant and best suited to the modeling of our system. These 13 diagrams are divided into two groups (**Figure 11**):

- Structural or static diagrams (structure diagrams).
- Behavioral or dynamic diagrams (behavior diagrams).

4.3. Use Case Diagrams

The use case diagram is a representation of the behavior of the user's point of view of the system, it is a definition of the needs that a user of the system expects, it contains all cases of use in direct or indirect connection with the actors.

Actors: Actors are represented in the form of small named characters; an actor represents the role played by a person or class that interacts with a system.

Use case: A use case is a classifier that models a feature of a system or class. The instantiation of a use case results in the exchange of messages between the system and its actors. It defines the expected activities of different users in relation to the application. The use case describes what a system does without specifying how it does it: it gives an external perspective on the system.

For our solution, its general use case diagram is as in **Figure 12**.

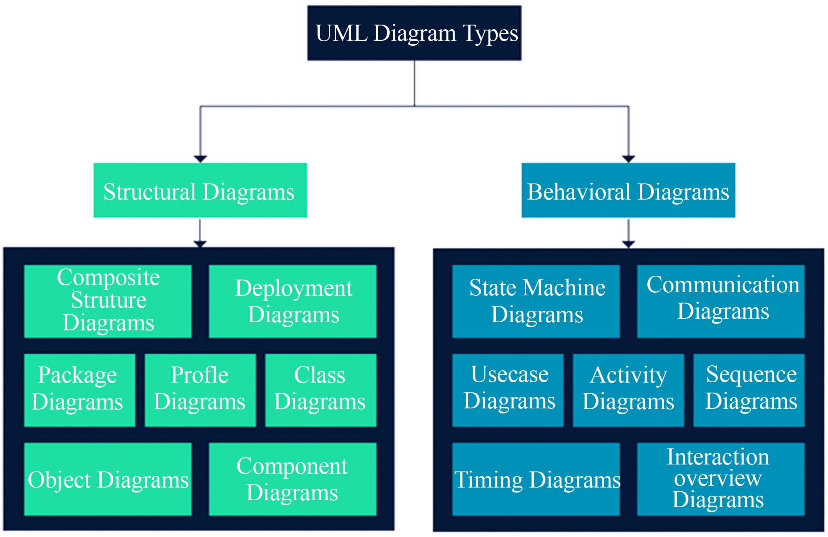


Figure 11. UML diagrams.

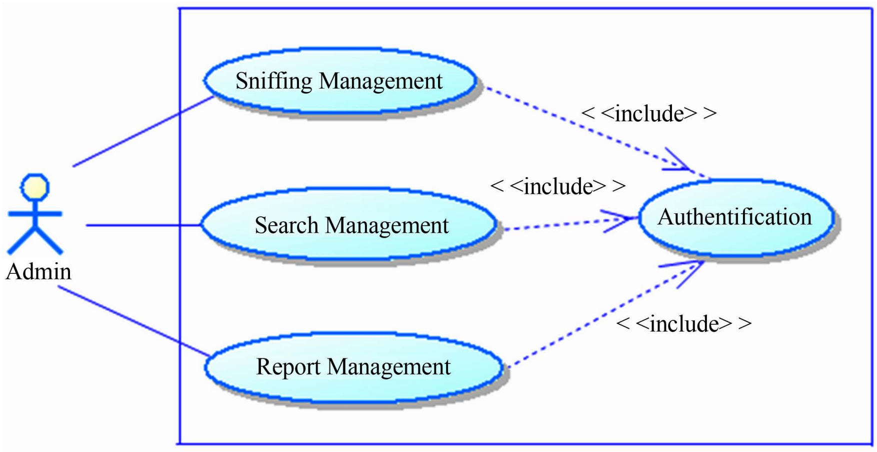


Figure 12. General use case diagram.

It is represented by a single user who comes in by the information system security manager as well as the following use cases:

- **Authentication**

The information system security officer must authenticate himself (type in a login and a password) to carry out all the operations on the system.

- **Sniffing management**

The "listening management" will ensure listening to log events generated on every equipment in the local network.

- **Search Management**

"Search management" will allow you to browse all the log files created after haven launched the sniffing on the network server in order to store in a database relevant information about reported attacks and events.

- **Report management**

"Report management" will enable a graphical interface to provide clear and reliable information and up-to-date statistics on data extracted from the log file database.

Refinement of use cases

- **Sniffing management**

This feature consists of listening to a specific port or network equipment, which may be different depending on their type. A firewall for example sends all daily events to this port. The operation of this module is to listen to the previously specified port, using the principles of the sockets, then to have a copy of the information sent by the firewall and put it in a log file that is created at the beginning of the application.

Figure 13 describes all the treatments performed through the Listening Management package.

The information system security manager identifies the firewall to be listened to through a well-defined address and then creates a file to back up Logs events to process it.

- **"Search Management"**

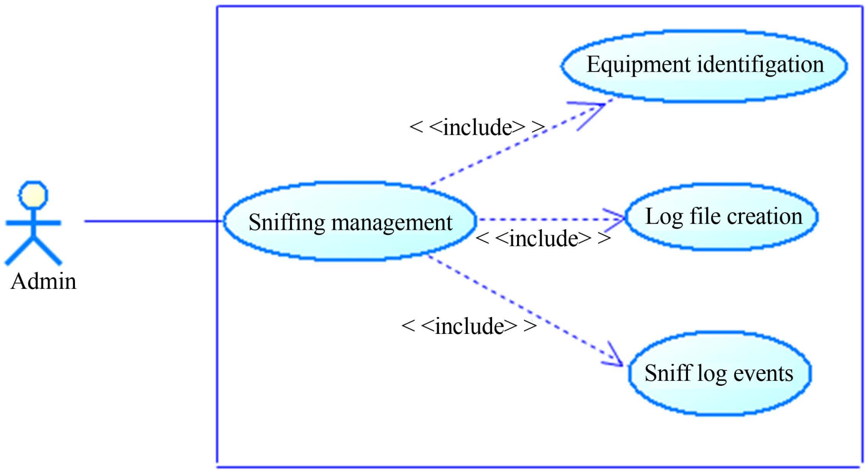


Figure 13. "Sniffing management" use case charts.

It involves extracting the data that exists in the log file and organizing it into a database (Figure 14).

The information system security manager must authenticate for each integration and backup of data from log files.

- **"Report Management"**

"Report management" provides a graphical interface displaying information and statistics on data extracted from the database. This information can be of several types (Figure 15):

- IP addresses (origins or recipients)
- Date and time of attacks
- Protocols used
- Type of daily event

The Information System Security Manager consults the dashboard so he can generate custom HTML reports. It also manages the rotation of log files. This rotation can be configured according to the size of the log files or a rotation frequency (daily, weekly) so the manager can view the characteristics of the log files (Name, Size, last change date) and choose which files to delete.

4.4. Platform's Class Diagram

Figure 16 represents the class diagram of our application, revealing how the database is built.

It is on this basis that the tables in our database as well as the classes involved in the system will be presented.

5. Results and Discussions

Materials and Tools

To set up this simulation, we used: Two laptops; one acting as a customer who will be responsible for the attack using the kali linux operating system and the other playing the role of a web server in which our applications will be hosted and which will be considered the victim running windows 10 operating system.

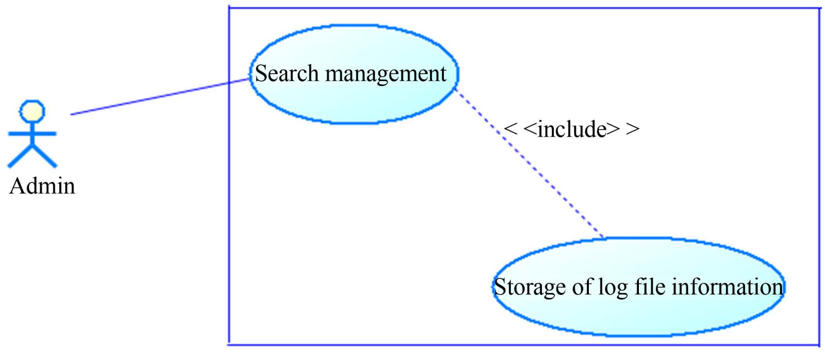


Figure 14. "Search management" use case charts.

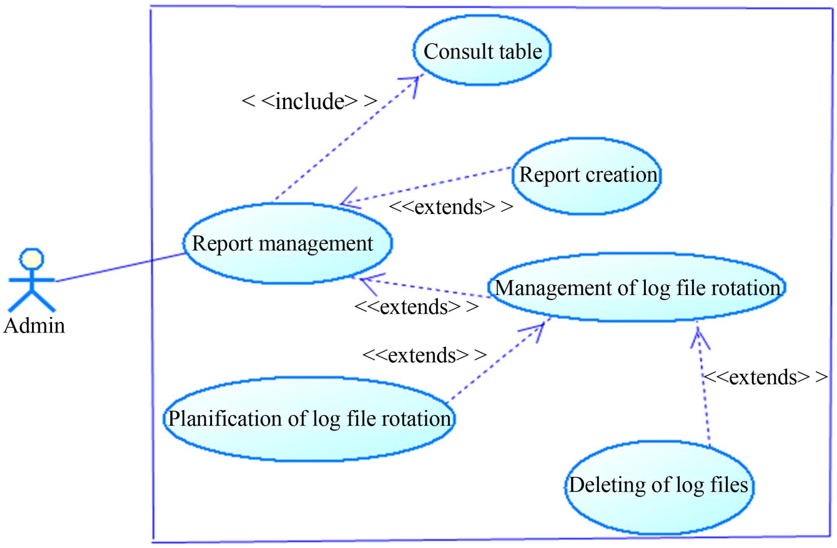


Figure 15. "Report management" use case charts.

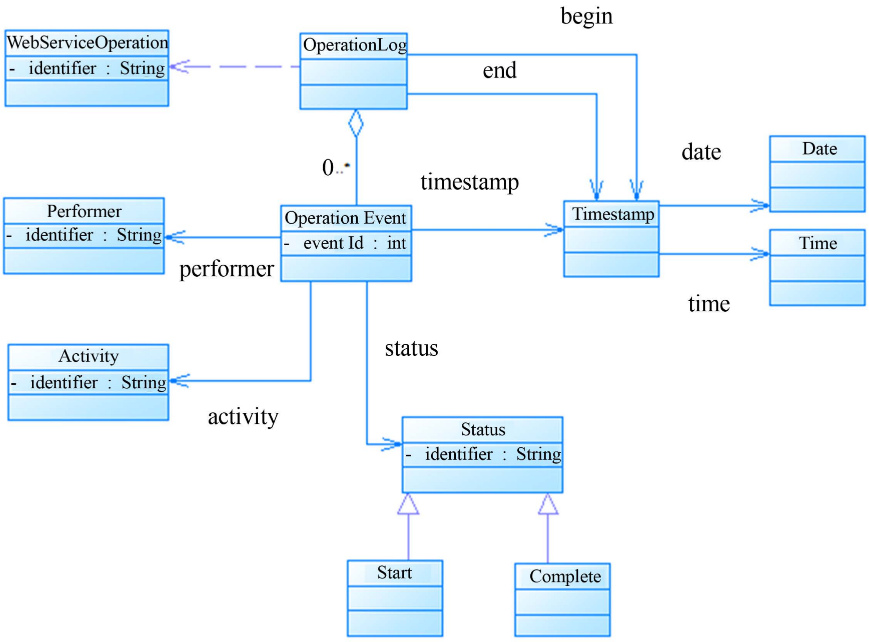


Figure 16. Class diagram of application.

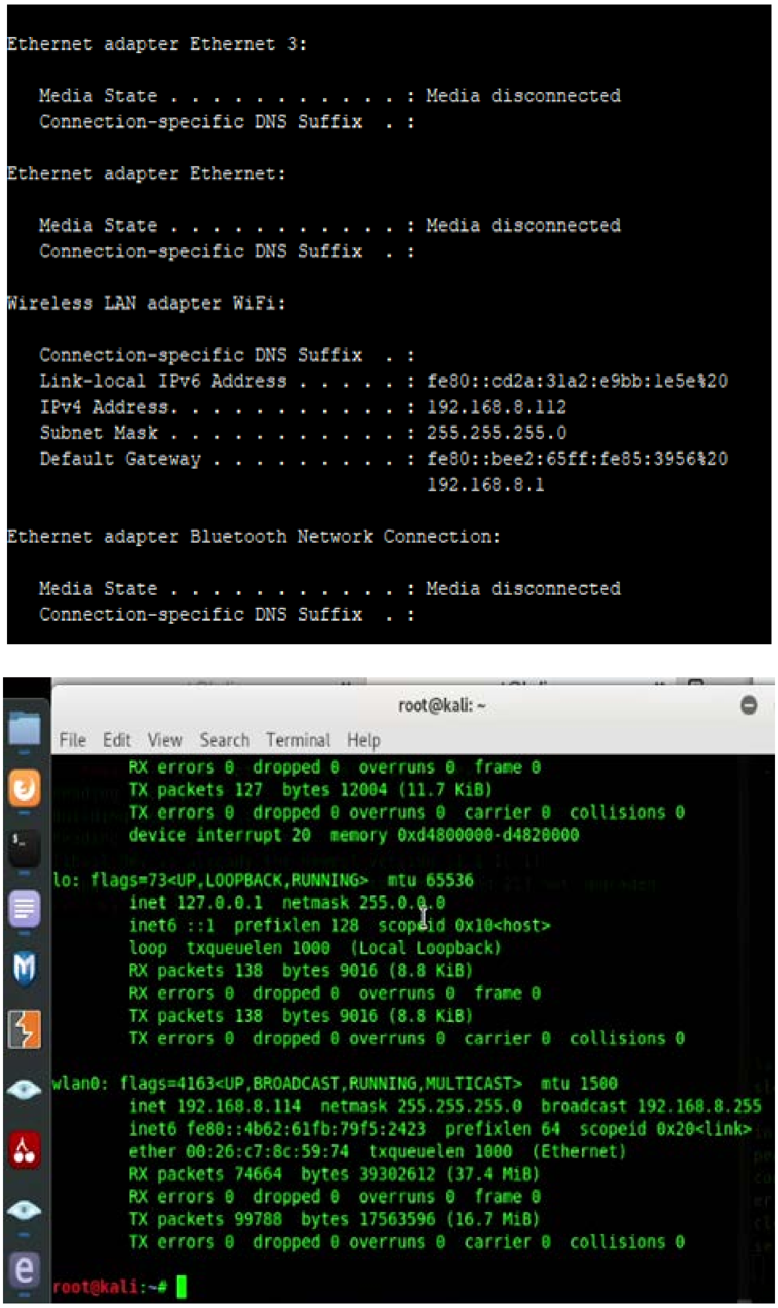
As Software, we use EasyPHP software which will make the second laptop act as a local web server.

Setting up the DOS

To set up the attacks, the hacker will use the following steps:

- **Connect to the server network**

To allow the exchange of packages between two entities, they must belong to the same network. Thus, the following screenshots present the result of *ipconfig* and *ipconfig* commands launched on the client and the server respectively (Figure 17):



```
Ethernet adapter Ethernet 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter WiFi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::cd2a:31a2:e9bb:1e5e%20
   IPv4 Address. . . . . . . . . . . : 192.168.8.112
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::bee2:65ff:fe85:3956%20
                                       192.168.8.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 127  bytes 12004 (11.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xd4800000-d4820000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 138  bytes 9016 (8.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 138  bytes 9016 (8.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.8.114  netmask 255.255.255.0  broadcast 192.168.8.255
        inet6 fe80::4b62:61fb:79f5:2423  prefixlen 64  scopeid 0x20<link>
        ether 00:26:c7:8c:59:74  txqueuelen 1000  (Ethernet)
        RX packets 74664  bytes 39302612 (37.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 99788  bytes 17563596 (16.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@kali:~#
```

Figure 17. IP addresses of both laptops.

Addresses 192.168.8.112 and 192.168.8.114 are well within the same network so the client and server can already communicate.

- **Starting the attack process**

Once the two laptops are connected, two tools will be needed:

- **Libssl-dev:** This is a package that is part of the implementation of the SSL project of cryptographic protocols SSL and TLS to communicate securely on the internet. This package provides development libraries, header files and guidelines for libssl and libcrypto.
- **Slow http-test:** is a highly configurable tool that implements most low-bandwidth DoS type of attacks on the applications layer, such as slow reading attacks (based on the persistent TCP exploit) by attracting a pool of simultaneous connections causing very large memory and processor use on the server.

- **Launching the attack**

With the tools already installed, all we have to do is set up the attack. For this, our target will be the website: **192.168.8.112/test1.0/index.php** hosted on our web server, which is as in Figure 18.

The attack is launched on the terminal via the command (Figure 19).

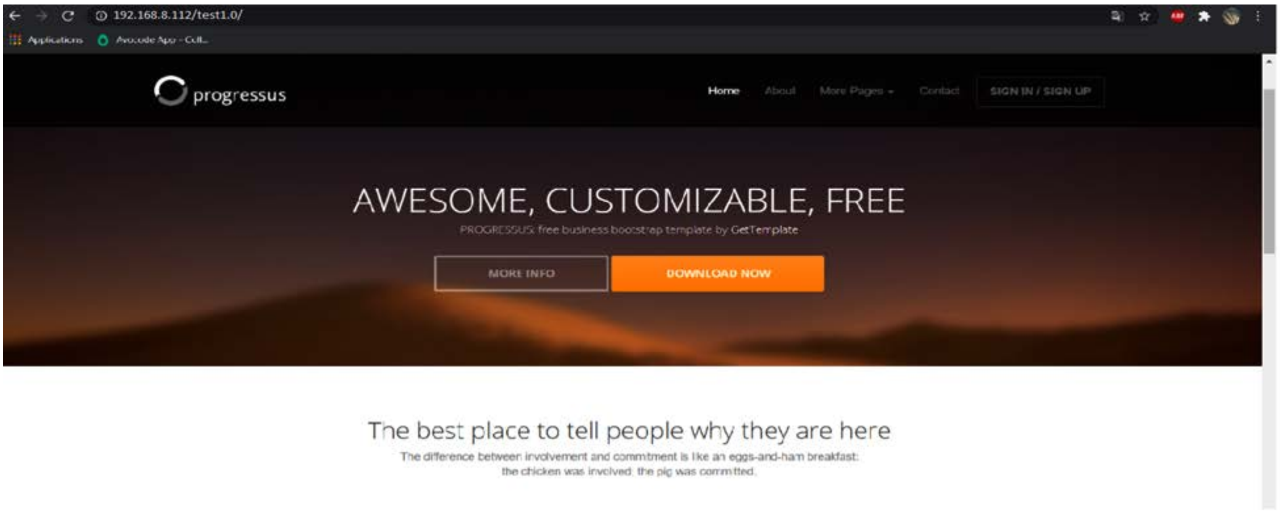
Start a slow loris test of 192.168.8.114 with 3000 connections, statistics are recorded in my header stats, the interval between the follow up data is 10 seconds and the connection rate is 300 connections per second:

- **Result of the attack**

Once the attack is launched it becomes impossible to connect to our website as shown in the screenshot (Figure 20).

The consultation of the logs file shows us that the server (192.168.8.112) has been saturated and will therefore remain unavailable until the intervention of network security administrator (Figure 21).

Solution to denial-of-service attacks:



192.168.8.112/test1.0/

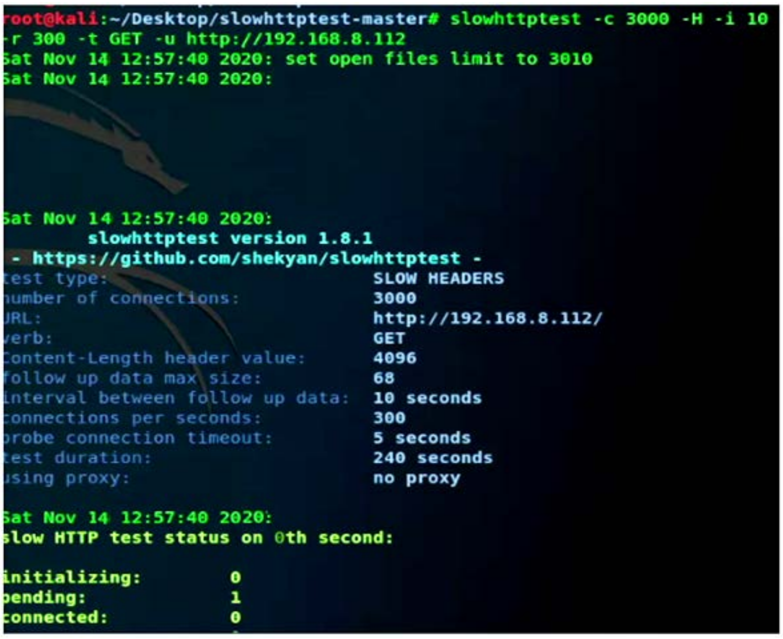
AWESOME, CUSTOMIZABLE, FREE

PROGRESSUS: free business bootstrap template by GetTemplate

The best place to tell people why they are here

The difference between involvement and commitment is like an eggs-and-ham breakfast: the chicken was involved; the pig was committed.

Figure 18. Interface of the site to be attacked.



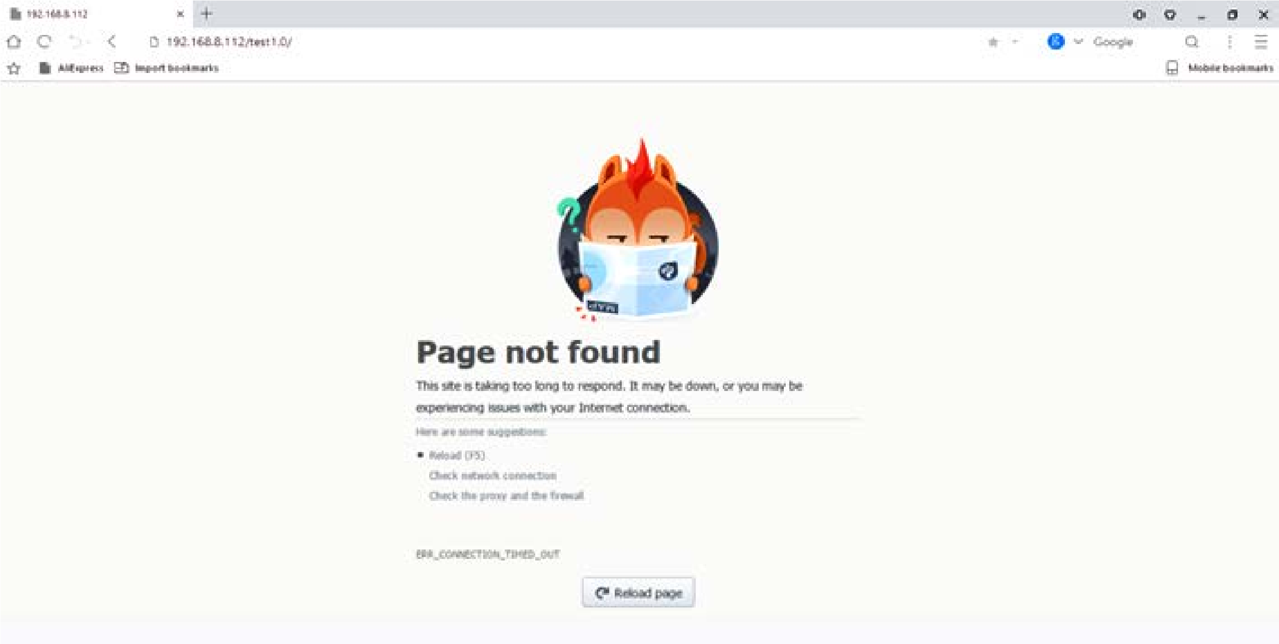
```
oot@kali:~/Desktop/slowhttptest-master# slowhttptest -c 3000 -H -i 10
r 300 -t GET -u http://192.168.8.112
at Nov 14 12:57:40 2020: set open files limit to 3010
at Nov 14 12:57:40 2020:


at Nov 14 12:57:40 2020:
        slowhttptest version 1.8.1
 - https://github.com/shekyan/slowhttptest -
est type:                           SLOW HEADERS
umber of connections:               3000
RL:                                 http://192.168.8.112/
erb:                                GET
ontent-Length header value:         4096
ollow up data max size:             68
nterval between follow up data:     10 seconds
onnections per seconds:             300
robe connection timeout:            5 seconds
est duration:                       240 seconds
sing proxy:                         no proxy

at Nov 14 12:57:40 2020:
low HTTP test status on 0th second:

nitializing:          0
ending:               1
onnected:             0
```

Figure 19. Launching of denial of service.



Page not found

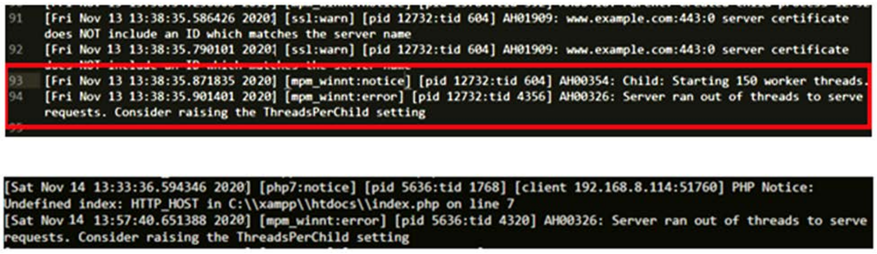
This site is taking too long to respond. It may be down, or you may be experiencing issues with your Internet connection.

Here are some suggestions:

- Reload (F5)
  Check network connection
  Check the proxy and the firewall

ERR_CONNECTION_TIMED_OUT

Figure 20. Site unavailable.



```
[Fri Nov 13 13:38:35.586426 2020] [ssl:warn] [pid 12732:tid 604] AH01909: www.example.com:443:0 server certificate
does NOT include an ID which matches the server name
[Fri Nov 13 13:38:35.790101 2020] [ssl:warn] [pid 12732:tid 604] AH01909: www.example.com:443:0 server certificate
[Fri Nov 13 13:38:35.871835 2020] [mpm_winnt:notice] [pid 12732:tid 604] AH00354: Child: Starting 150 worker threads.
[Fri Nov 13 13:38:35.901401 2020] [mpm_winnt:error] [pid 12732:tid 4356] AH00326: Server ran out of threads to serve
requests. Consider raising the ThreadsPerChild setting
```

```
[Sat Nov 14 13:33:36.594346 2020] [php7:notice] [pid 5636:tid 1768] [client 192.168.8.114:51760] PHP Notice:
Undefined index: HTTP_HOST in C:\\xampp\\htdocs\\index.php on line 7
[Sat Nov 14 13:57:40.651388 2020] [mpm_winnt:error] [pid 5636:tid 4320] AH00326: Server ran out of threads to serve
requests. Consider raising the ThreadsPerChild setting
```

Figure 21. Apache logs after attack.

- **The Homepage**

This is an authentication page consisting of a login and password to guarantee the principle of confidentiality. This page incorporates tokens against the CSRF to avoid execution of actions on the administrator's computer without his knowledge (Figure 22).

- **The supervision page**

This is a visual representation of the most important information, grouped on a screen so that it can be easily understood: the amount of traffic coming in and out, the percentage of queries having been successful (Figure 23).

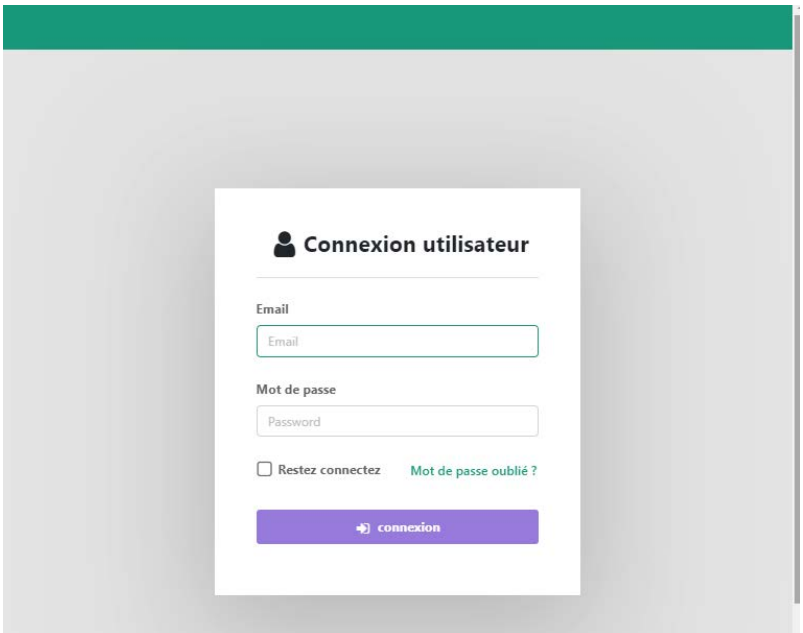


Figure 22. Authentication page.

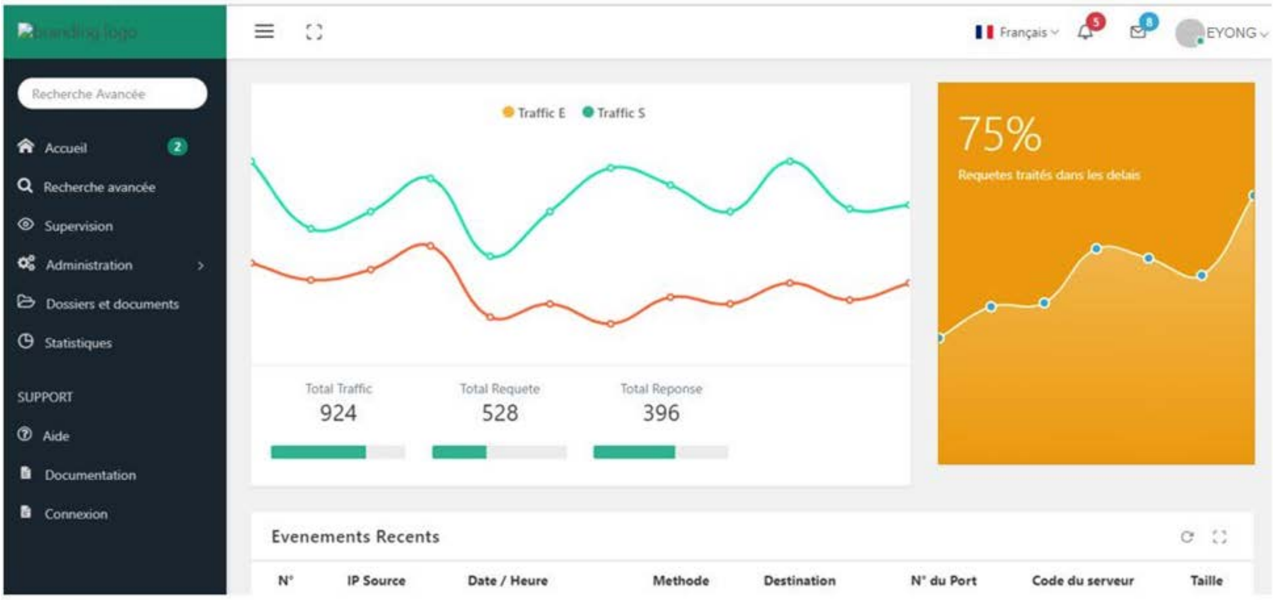


Figure 23. Supervision interface.

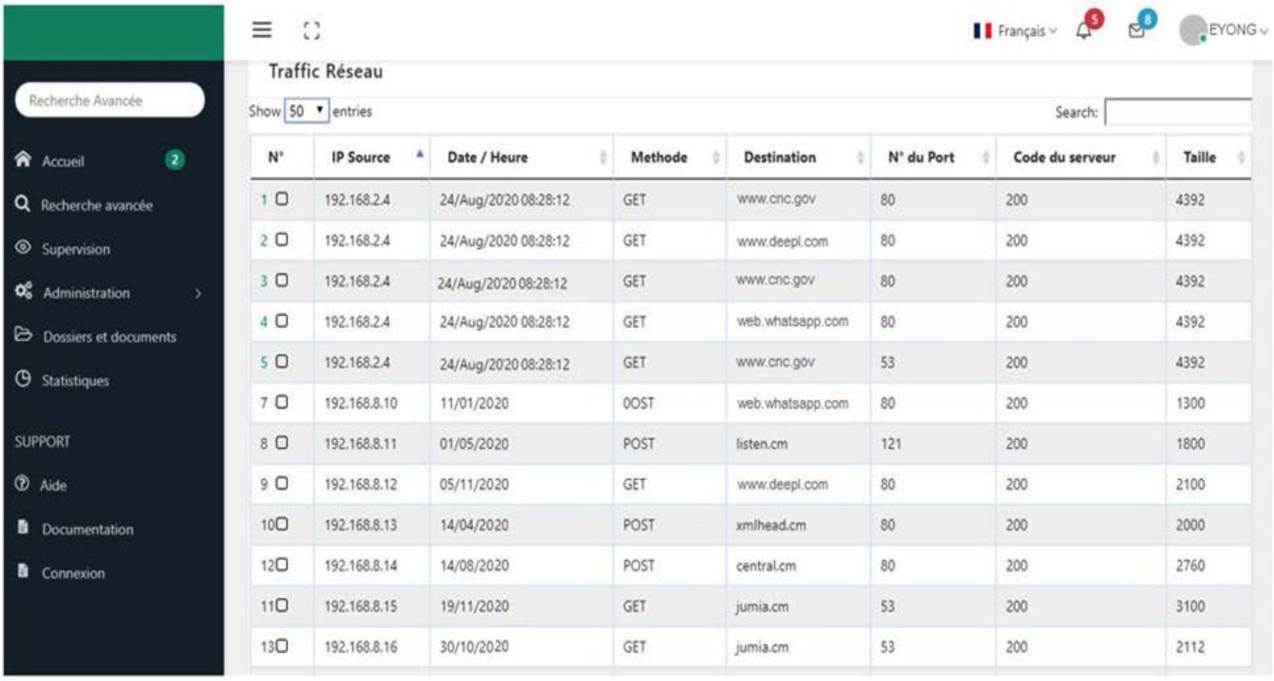
- **Log management page**

It will consist of "smart" control panels that display logs in real time in a clear and simple way. This page also includes features to get details on each line of the file (Figure 24).

This page allows, among other things, to search logs based on IP addresses or port numbers (Figure 25).

- **Alert page**

This is a modal page that will be displayed automatically to notify the administrator about an anomaly or attack going on within the system. It must also provide details about the anomaly: the origin of the impact (Figure 26).

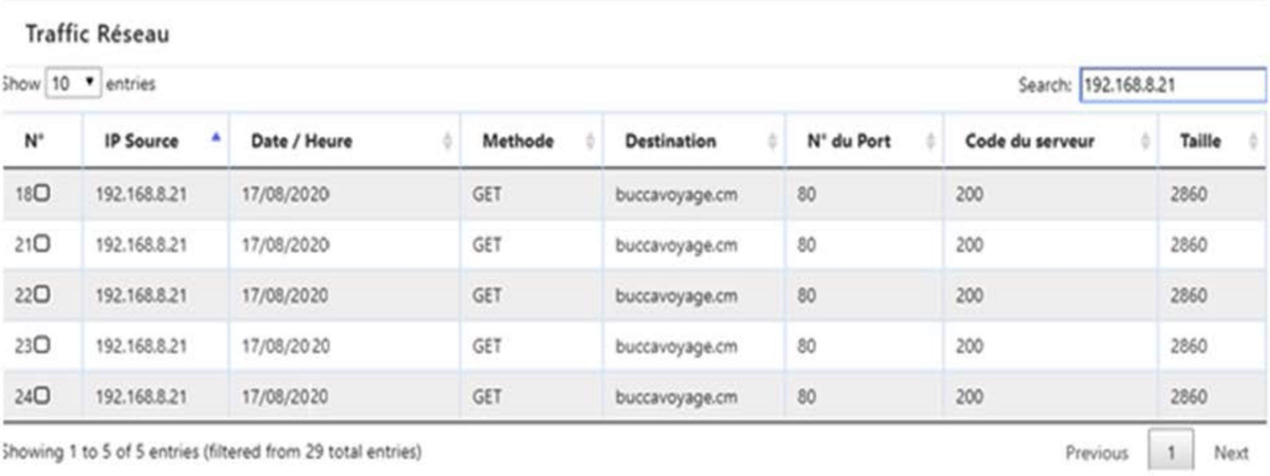


Traffic Réseau

Show 50 entries Search:

N°	IP Source	Date / Heure	Methode	Destination	N° du Port	Code du serveur	Taille
1 □	192.168.2.4	24/Aug/2020 08:28:12	GET	www.cnc.gov	80	200	4392
2 □	192.168.2.4	24/Aug/2020 08:28:12	GET	www.deepl.com	80	200	4392
3 □	192.168.2.4	24/Aug/2020 08:28:12	GET	www.cnc.gov	80	200	4392
4 □	192.168.2.4	24/Aug/2020 08:28:12	GET	web.whatsapp.com	80	200	4392
5 □	192.168.2.4	24/Aug/2020 08:28:12	GET	www.cnc.gov	53	200	4392
7 □	192.168.8.10	11/01/2020	0OST	web.whatsapp.com	80	200	1300
8 □	192.168.8.11	01/05/2020	POST	listen.cm	121	200	1800
9 □	192.168.8.12	05/11/2020	GET	www.deepl.com	80	200	2100
10□	192.168.8.13	14/04/2020	POST	xmlhead.cm	80	200	2000
12□	192.168.8.14	14/08/2020	POST	central.cm	80	200	2760
11□	192.168.8.15	19/11/2020	GET	jumia.cm	53	200	3100
13□	192.168.8.16	30/10/2020	GET	jumia.cm	53	200	2112

Figure 24. Log management page.



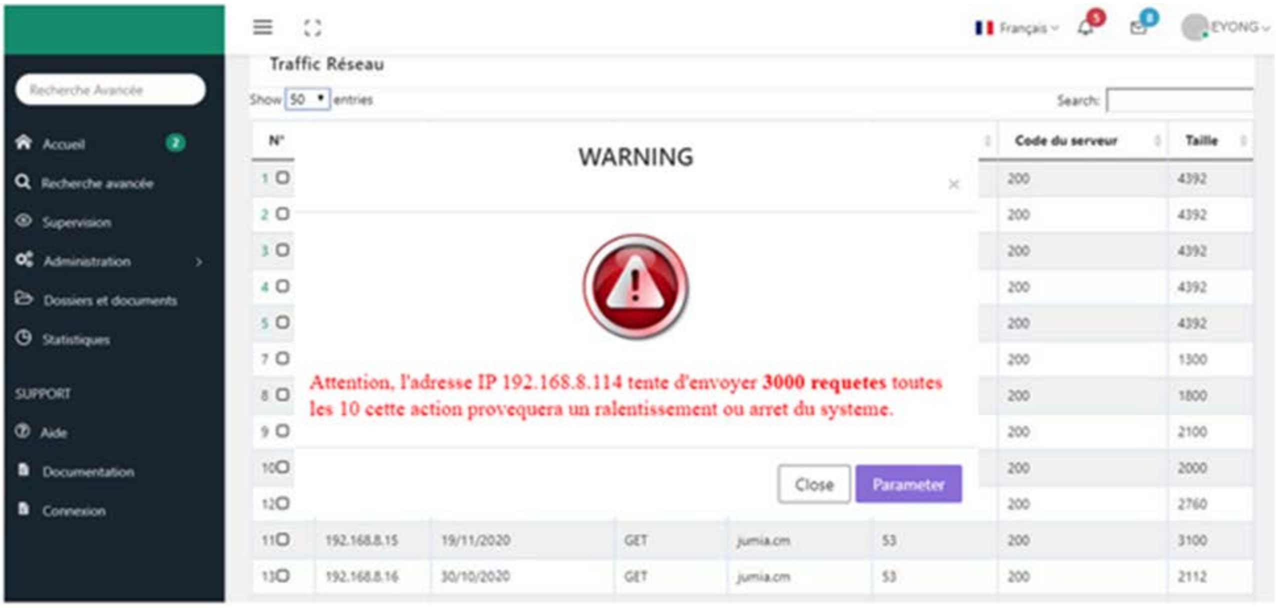
Traffic Réseau

Show 10 entries Search: 192.168.8.21

N°	IP Source	Date / Heure	Methode	Destination	N° du Port	Code du serveur	Taille
18□	192.168.8.21	17/08/2020	GET	buccavoyage.cm	80	200	2860
21□	192.168.8.21	17/08/2020	GET	buccavoyage.cm	80	200	2860
22□	192.168.8.21	17/08/2020	GET	buccavoyage.cm	80	200	2860
23□	192.168.8.21	17/08/2020	GET	buccavoyage.cm	80	200	2860
24□	192.168.8.21	17/08/2020	GET	buccavoyage.cm	80	200	2860

Showing 1 to 5 of 5 entries (filtered from 29 total entries)

Figure 25. Search for events issued by 192.168.8.21.



WARNING

Attention, l'adresse IP 192.168.8.114 tente d'envoyer **3000 requetes** toutes les 10 cette action provequera un ralentissement ou arret du systeme.

N°						Code du serveur	Taille
1 ○						200	4392
2 ○						200	4392
3 ○						200	4392
4 ○						200	4392
5 ○						200	4392
7 ○						200	1300
8 ○						200	1800
9 ○						200	2100
10 ○						200	2000
12 ○						200	2760
11 ○	192.168.8.15	19/11/2020	GET	jumia.cm	53	200	3100
13 ○	192.168.8.16	30/10/2020	GET	jumia.cm	53	200	2112

Figure 26. Alert page.

6. Conclusions

With the completion of our work in the "Implementation of a machine learning algorithm for the detection and prevention of attacks in supervised network", we can say that our objectives have been achieved. Using machine learning algorithms and big data technology, we were able to implement a solution that allows us to visualize network events, to quickly search for logs in the Syslog server and to automatically detect attacks within the network.

The implementation of this platform in the supervised network will allow us to have: a highly available network architecture; a reinforcement of security and detection of illegitimate flows; the possibility to generate real-time alerts on malfunctions; real-time monitoring of the use of an application (number of users, functions used, etc.); the identification of the origin of incidents occurring in applications.

Nevertheless, "event storms" do not only occur during network attacks such as denial of service attacks but also during campaign applications which is why this work could be extended by integrating the possibility of automatic standby for low priority events.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Bourget, E. (2016) New Perspectives for the Use of Logs in a Context of Computer Security. University of Montreal, Montreal.

[2] Sendrier, N. (2007) Introduction à la théorie de l'information. Centre Inria de Paris, Ecole Polytechnique.

[3] Dorard, L. (2020) Architecture of a Real-Word Machine Learning System. https://medium.com/louis-dorard/architecture-of-a-real-world-machine-learning-system-795254bec646

[4] Kone, A. (2013) Research and Development and Technology Watch on Big Data Trends and Technological Concepts. INSA of Lyon, Lyon.

[5] Melnick, J. (2018) Top 10 Most Common Types of Cyber Attacks. https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/

[6] Kang, M.S., Lee, S.B. and Gligor, V.D. (2013) The Crossfire Attacks. *Proceedings of the* 2013 *IEEE Symposium on Security and Privacy*, San Francisco, May 2013, 127-141. https://doi.org/10.1109/SP.2013.19

[7] Benzaki, Y. (2017) Nine Machine Learning Algorithms Every Data Scientist Should Know. https://mrmint.fr/9-algorithmes-de-machine-learning-que-chaque-data-scientist-doit-connaitre

[8] Moutarde, F. (2011) Machine Learning. CAOR, MINES ParisTech.