International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

A PRACTICAL APPROACH TO BUILD A SYSTEM FOR THE COLLECTION AND ANALYSIS OF INTERNET TRAFFIC RECORD FOR LAW ENFORCEMENT

Ebot Ebot Enaw
University of Yaounde I, Cameroon
National Advanced School of Engineering
Department of Computer Sciences

Abstract

In recent years, Internet has become more and more present in our daily lives and has emerged as a major driver of economic growth. Unfortunately, some people use this precious and helpful tool to commit malicious acts and crimes usually called cybercrimes. In an effort to identify and track these cybercriminals, law enforcement officers have to carry out investigations on several evidences notably the Internet traffic flow record of ISP. Some articles related to this topic have been published namely [7] which while identifying the strengths and limitations of netflow and sflow proposed an architecture where netflow/sflow are combined with a distributed network analyzer to provide better performance for traffic monitoring, [2] which firstly identifies weaknesses of netflow and makes some proposals to enhance it namely the dynamic adapting sampling rate of Netflow to achieve robustness without sacrificing accuracy. This paper presents the prototype of the system that NAICT, the Cameroon Government Agency in charge of ICT development and cyber security developed to collect and analyze Internet traffic flow record of ISPs. Our article is structured as follows: section 1 introduces the topic, section 2 presents some research papers related to our topic, section 3 presents the global architecture of an ISP, section 4 states the problem, section 5 presents the ETSI framework related to Telco Data Retention, section 6 presents the Non-Relational Database concept, section 7 presents data compression techniques, section 8 presents our methodology and section 9 illustrates our methodology and presents some results of our prototype.

***Keywords**: Data Retention, Lawful Interception, Internet Monitoring, NoSQL, parallel computing.*

1 Introduction

Our society is increasingly dependent on ICT and Internet to assist us in almost every aspect of daily life. The wide spread of cybercriminality combined with the ever growing importance of Internet in our daily lives make the issue of Internet data collection and analysis critical for governments and particularly for law enforcement. In this light, standards have been developed namely ETSI and CALEA which are applicable in the European Union. Nevertheless, these standards don’t dwell deeply on the technical specifications regarding the implementation of this kind of system. In this paper we present the technical specifications of the solution that NAICT developed for collecting and analyzing Internet traffic record of ISPs and present some results obtained. The aim of this paper is to provide governments especially those of developing countries with a methodology and technical clues to develop a scalable system for collecting and analyzing Internet traffic record data at an affordable price. In order to better understand the problem this article is trying to solve, a good mastery of the technical architecture that an ISP uses to deliver Internet to its customers is required; section 3 covers this issue. But before that, let’s have an overview of some research papers published on topics related to the one we are dealing with in this article.

2 Related work

Some research have been done on topics related to this issue namely [7] which identifies the strengths and limitations of netflow and sflow, proposes an architecture where netflow/sflow are combined with a distributed network analyzer to provide better performance for traffic monitoring, [2] which identifies weaknesses of netflow and makes some proposals to enhance it namely; the dynamic adapting sampling rate of Netflow to achieve robustness without sacrificing accuracy. These two articles deal only with a single component of the issue at hand, that is IP traffic collection, the other components which are data (IP, AAA, and customer’s personal information) storage and analysis are not taken into account. However, other articles deal with the storage issue including [12] that presents the concept of NoSQL database which is designed to handle a huge volume of data, [9] presents Relational database and Non-Relational database and then compares several NoSQL database management systems and presents their advantages and inconveniences in different situations. [10] makes a comparative study between several compression algorithms that can be used to save space when dealing with huge volumes of data. The aforementioned articles deal with a specific part of the whole issue, none of them treats the entire problem. Our article presents a complete solution that encompasses collecting, storing, compressing and analyzing the data of ISP with a view to identifying the author of cybercrimes so as to diligently respond to law enforcement requests.

3 Typical network architecture

The typical network architecture of an ISP is depicted in the figure below: It has three main components:

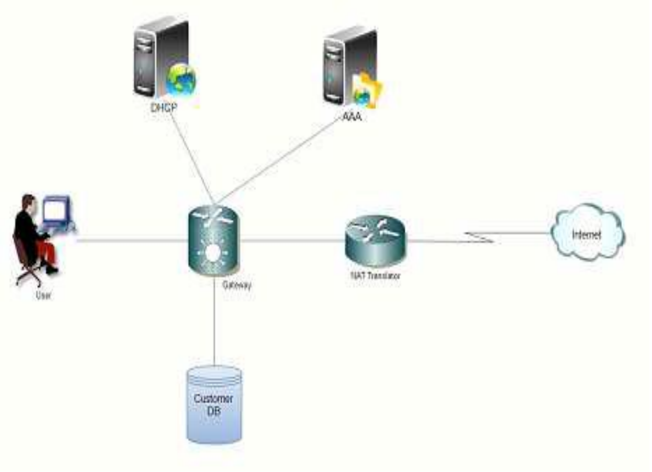


Figure 1 : Typical ISP architecture

The AAA server: It is a server that carries out three main functions: user authentication, authorization and accounting. User authentication refers to the process of authenticating an entity's identity, typically by providing evidence that it holds a specific digital identity such as an identifier and the corresponding credentials. Examples of types of credentials are passwords, one-time tokens, digital certificates, digital signatures and phone numbers (calling/called). Authorization is the process of checking to see if a user is entitled to access a particular service, based on a configured policy which can be based on several parameters namely the time, the location, etc. Accounting consists of tracking and logging predefined users activities. Accounting can serve several purposes such as billing and trend analysis.

The NAT server: Because private addresses are not reachable on the Internet and public IP addresses are scarce especially IPv4, coupled with the fact that ISPs often have many clients, they have no choice than to implement NAT translation to translate private addresses into public ones. Usually this function is performed by a network equipment such as a router or a firewall. There are two types of NAT: static and dynamic. Static NAT translates a private address to the same public address while dynamic NAT translates a private address to a randomly chosen public IP in a pool.

The Customer Database: It is the database that holds personal information of clients such as names and addresses. Usually it is embedded in a CRM (Customer resource management).

The workflow of a user going to the Internet can be described as follows: The user is authenticated to the AAA and then authorized to access specific services, after that, he is allocated a private address by the DHCP server, that he will use until he disconnects. The private address is then translated by the NAT device along the way to Internet into a public one. It is this public IP address that the server with which the user is communicating, will see in its log file. When the client disconnects, the private address is released and the activities of the user are logged in the accounting server according to the policy configured.

4 Research problem

In order to determine the identity of a person that commits a particular crime through the Internet, Law enforcement officers regularly issue requests to NAICT. Usually, these requests state the public IP address used by the client and the timestamp of the connection gathered during evidence collection. Thus, to be able to determine the identity of the culprit, we have to correlate information from the three aforementioned data sources namely the AAA server, the NAT log and the client database given that we supposed the timestamp and the public IP address implicated in the crime are known.

This process is made complex firstly, by the usage of the NAT and the DHCP protocols in ISP networks which permit; millions of users to have the same public address at the same time and a private IP address to be allocated dynamically to several users at different times and secondly, by the fact that the data we have to correlate are bulky, unstructured and scattered among different equipments. In this article, we will propose an architecture and a methodology based on open source tools and concepts like parallel computing, Non-Relational Database, compression algorithm to collect, store and correlate the huge volume of data of the aforementioned data sources in order to identify the author of cybercrimes.

5 ETSI standard

In an effort to harmonize the way lawful interception activities are carried out while lowering costs, the European Telecommunications Standards Institute (ETSI) designed a set of documents including ETSI 102-656, 102-657, 102-528, 102-661 ([6], [4], [5], [3]) that define a systematic and extensible means by which network operators and law enforcement agents (LEAs) can interact, especially as networks grow in sophistication and scope of services. It is important to note that this standard applies not only to traditional wireline and wireless voice calls, but to IP-based services such as Voice over IP, email, instant messaging, etc. The architecture is now applied worldwide including in the United States in the context of CALEA compliance. The architecture proposed by ETSI includes three stages namely:

1. Collection; where target-related call data and content are extracted from the network
2. Mediation; where the data is formatted to comply with specific standards
3. Delivery of the data and content to the law enforcement agency (LEA).

The architecture illustrating the interaction of the different components in the Data Retention System is depicted in the figure below:

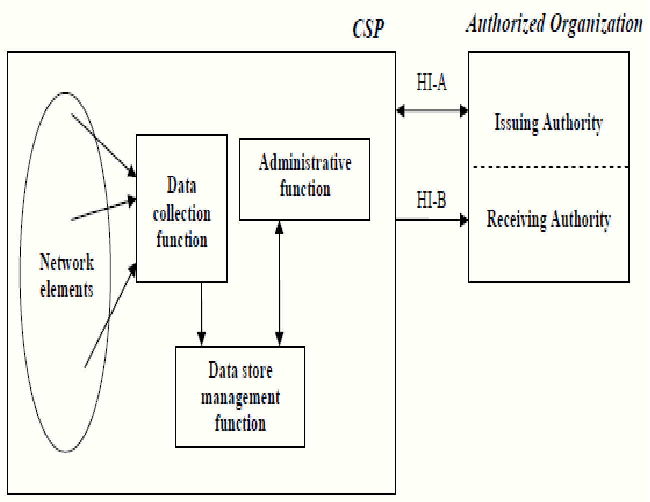


Figure 2 : Functional model of Data Retention System

According to this figure, there are two main blocks:

CSP (Content Service Provider) which can be an ISP. In the CSP, the *data collection function* collects the data from the network devices and stores them in the *Data Store Management* module which indexes them and handles queries. The *Administrative function* handles the administrative requests submitted by the Authorized organization through the HIA interface.

Authorized organization: The Authorized organization can be law enforcement or any organization authorized by the legislation of the country concerned to issue requests and to collect traffic record data from Telco and ISP. Within the *Authorized organization* block, there are: *issuing authority* which is responsible for emitting the requests while the *receiving authority* is responsible for receiving and handling the requested retained data corresponding to a request previously submitted by the issuing authority.

Although the ETSI 102-657 annexes described some specifications of the data to be collected and the way they can be transported and secured, the ETSI fails to give technical guidelines on how to practically collect and analyze the data of a CSP network. In subsequent sections we propose a practical approach to collect and analyze these data in order to identify clients based on the public source ip address and the timestamp of the connection.

6 Non-Relational databases

Since Relational databases, were neither able to cope with the scale and agility challenges that face modern applications, nor take advantage of the cheap storage and processing power available today, a new type of database has emerged since 2000: NoSQL. NoSQL encompasses a wide variety of different database technologies and were developed in response to an increase in the volume of data stored about users, objects and products, the frequency with which this data is accessed, and performance and processing needs. Non-relational databases differ from relational database in several aspects: first relational database management system store data in tables based on mathematics properties while Non-relational databases don’t, secondly relational database use the query language SQL while its counterpart doesn’t. The main advantages of using Non-relational databases instead of relational ones are:

- high throughput: NoSQL database can handle huge volumes of data and requests on these data;
- high scalability with the support of auto-sharding, NoSQL scale horizontally which means that to add capacity, the database administrator simply needs to add servers with little configurations;
- high flexibility: NoSQL databases are schema-less which means that data with different structure can be stored together and the ”schema” of the database can be modified as needed. This feature is very useful with the agile development approach where the database is modified as new features are developed.

Nevertheless, NoSQL databases also have some flaws including the fact that it doesn’t guarantee natively the ACID properties (Atomicity, Consistency, Isolation and Durability) which are very important for the reliability of database operations, so to overcome this issue, the developer has no choice than to deal with that in his code. Many research papers have been published on this topic namely [9] which presents the various features of relational and non-relational databases and secondly states their advantages and disadvantages. [8] adopts a more elaborate approach to compare Non-relational databases to Relational databases as it first proposes a benchmark methodology for database and a tool to carry out the benchmarking, and secondly applies its methodology and uses its tools to compare MongoDB to MySQL on specific criteria including read/write performance. Its comparison reveals that MongoDB is better than MySQL in terms of read operations, performance as well as write operations.

7 Data compression techniques

Data compression is a method used to reduce storage cost by eliminating redundancies that occur in most files [13]. The goal of data compression is to represent a source in digital form with as few bits as possible while meeting the minimum requirement of reconstruction of the original. There are two types of data compression algorithms: lossless and lossy. Lossless algorithms permit to exactly reconstruct the original file from the compressed version whereas lossy compression algorithms can only reconstruct an approximation of the original file from the compressed version. Usually lossless algorithms are used when the original data is very important whereas lossy algorithms are used for data like music and video where the whole data is not indispensable. Data compression has provoked a lot of interests among researchers leading to the publication of some articles namely [10] which presents two lossless data compression methodologies namely Huffman encoding and arithmetic encoding and then conducts a comparative study between them which ends up concluding that arithmetic encoding is more powerful than Huffman encoding in terms of compression ratio; [11] presents statistical compression techniques (Shannon-Fano encoding, Huffman encoding, Adaptive Huffman encoding, Run Length Encoding and Arithmetic encoding) and dictionary based(LZ77 and LZ78 families) and then compares their performances on text files that results in the conclusion that among statistical compression algorithms compared, arithmetic encoding is the best, among dictionary based algorithms of LZ77 family, LZB is the best while in the LZ78 family, LZFG is the best; [1] presents and conducts a comparative study of four lossless compression algorithms namely LZ77, LZW,PPM and BWT on mobile platforms that lead to the conclusion that LZ77 is the most efficient algorithm to be implement on mobile platform. In our article, since each data is important to us, as a loss in one bit can compromise our search, we will use lossless algorithm.

8 Our solution

8.1 Methodology

The methodology used for our system consists of five main steps:

1. Identify in the ISP architecture, the equipments holding the information that matter to us namely NAT table, AAA accounting repository, Customer database ;
2. Develop a module that will connect to these equipments and collect the aforementioned information in the raw format ;
3. Develop a module that will transform the format of the data collected in the previous step into a more comprehensive, efficient and structured one ;
4. Develop a module that will store the data after their transformation in order to ensure: optimal data storage, improved read and write operations in order to efficiently deal with the huge volume of data and the speed with which they are generated ;
5. Develop a module that will query and correlate those data when needed in order to respond diligently to law enforcement requests.

Following this methodology and as depicted in the figure below, we came up with an architecture that comprises four main modules namely Collector, Parser, Archiver and Analyzer, which will be described in the subsequent sections. The architecture of our system is depicted in the figure below:

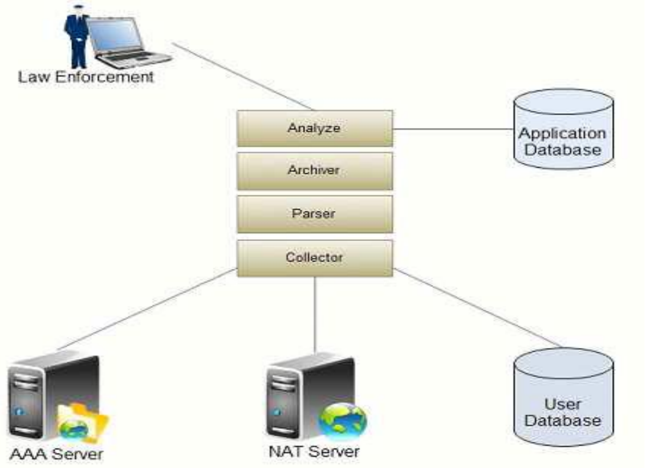


Figure 3 : Functional model of Data Retention System

8.2 Presentation of the environment

The ISP where we experimented our solution has around 3 million customers across the country. Their core network is distributed among two facilities located in two different places which handle each, fifty percent of their customers. We deployed our solution in only one of their core network facilities, which handles almost 1.7 million customers. The volume of data in their raw format generated by the NAT table and the AAA every day is around 20GB. The NAT table generates around 864 million entries per day, and 100,000 every minutes. The AAA accounting repository generates about 50,000 files everyday each containing around 100 entries at least. The ISP delivers Internet to its customers through mobile devices that have a SIM number, using the EVDO technology.

According to the first step of our methodology, we identified the equipments that handle the data we need (NAT translations, AAA accounting logs, customers information) but unfortunately, for security purposes we cannot reveal their characteristics in this article.

8.2.1 Collector

This module maps to the second step of our methodology. It collects rough data from the three data sources namely:

The AAA server: in fact, the AAA server has an FTP server embedded where it copies all the log files in real time. These log files are limited to the size of 3Ko, so when a file reach this size, a new file is created. To collect these files we developed an FTP client in Java that checks and downloads new files from the FTP server every 30 seconds. The structure of this log file is presented in annex 2.

The NAT device: Every time a user connects to a website or to any Internet server, the NAT device translates its private IP address to a public one and keeps a trace of this translation in the memory. In our case, the NAT device is a firewall that uses dynamic NAT. We configured the firewall so that it logs all the NAT entries to a log server (syslog-ng) we deployed. The syslog-ng server was configured in such a way that it stores the NAT session record of each minute in a file whose name respects the following syntax: **yyyy-mm-dd-hh-ii.log** where **yyyy** represents the year, **mm** the month, **dd** the date, **hh** the hour and **ii** the minutes. This syntax facilitates the indexing of these files and in turn eases the search process. A NAT log file comprises a set of lines which represents NAT translations which occurred during one minute. Each line is a collection of 08 fields which are described below:

Table 1:Initial NAT log structure

N°	Fields
1	Timestamp of the SYSLOG message
2	Gateway IP address
3	Protocol used
4	Private Source IP address + port
5	Public Source IP address + port
6	Destination IP address + port
7	Timestamp of the beginning of the session
8	Timestamp of the end of the session

The user database: This database keeps all information about the clients (name, address, profession, SIM number, etc.). In order to identify cybercriminals, we developed a module that interacts with this database through web-services so as to match the phone number to customer's personal information (name, address, etc.).

8.2.2 Parser

This module first of all parses the rough data collected from the AAA and the NAT device and keeps only the relevant data in a specific format. Through the Parser module we transformed the initial AAA log file that has 92 fields into a new one that has only 09 fields which are described, in the table below:

Table 2:AAA log structure

N°	Field	Description
1	Phone number	Phone number of the client
2	Private address	Private address allocated to the client upon connection
3	BSID	Identification of the base station to which the client was linked when he connected
4	USERZONE	USERZONE
5	MEID	Identifier of the terminal
6	TIMESTAMP	TIMESTAMP of the connection
7	Session Continue	Session stop flag, indicating whether this CDR record is the last CDR record in the current session
8	Beginning Session	Session start flag, indicating whether this CDR record is the first CDR record generated in the current session

The other fields were deleted because they were not necessary for the identification of the client through its public IP address and the time period. One should note that, since the AAA server logs all the information related to connection/de-connection of users as they come, these information are not arranged in order making it difficult to identify the beginning and the end of a session which are mandatory for the Analyzer module. To that end, the Parser module after collecting and selecting the relevant fields, reconstructs (identifies the beginning and the end of each session) and places the different sessions in the new file. After the parser execution, the NAT log file lines comprises 05 fields instead of the 8 that were initially present:

Table 3: Final NAT log structure

N°	Fields
1	Protocol(TCP/UDP)
2	Private IP address + port
3	Destination IP address + port
4	Timestamp of the beginning of the session
5	Timestamp of the end of the session

8.2.3 Archiver

This module maps to the third steps of our methodology. It has two main features:

- **Data compression**: Given the large amount of data generated by the network, and the limited capacity of our servers we had to compress the data in order to save space. This module compresses the AAA and NAT log files collected and previously parsed, and indexes them by date to facilitate the search. When we want to search information on this data within a specific period, it decompresses it. In order to choose the best compression tool, we carried out a comparative study among 7 tools based on five main criteria namely: the compression rate, the compression speed, the decompression speed, the CPU and RAM load. The results of this study are presented in the table below:

Algorithm	Operating System	Compression level	Initial size of the file (Gb)	Final size of the file (Mb)	Compression rate (%)	Compression time	Decompression time	CPU load (%)	RAM usage (Mb)
UHA	CentOS 6.4	PPM	1.91	290.4	14.8	32min00s	01min43s	5	41
7z	CentOS 6.4	LZP	1.91	290.4	14.8	32min09s	01min45s	5	41
		Ultra	1.91	234.2	11.9	42min32s	00min49s	11	679.5
		PPMd	1.91	203.5	10.37	03min27s	03min51s	6	18.6
RAR	CentOS 6.4	Ultra	1.91	200.47	10.25	10min00s	01min00s	7.94	20
GZIP	CentOS 6.4	Ultra	1.91	340.12	17.39	10min00s	04min00s	6	1.05
BZ2	CentOS 6.4	Ultra	1.91	570.12	29.15	09min00s	08min00s	5	3
BZA	CentOS 6.4	Ultra	1.91	564	28.86	15min00s	12min00s	5	5
KZIP	CentOS 6.4	Ultra	1.91	340.12	17.39	10min00s	04min00s	6	1.05

Table 4: Comparison of compression tools

With regards to the results of this study, we choose 7z algorithm with the PPMd level because it is the most efficient in terms of compression/decompression time and compression rate.

- **Data storage**: When data has to be fetched, it is decompressed and loaded into a database. This specific feature is in charge of loading and handling data in the database. Given the huge volume of data we have to fetch, and for other reasons mentioned in section 6 we decided to use a NoSQL Database management system namely MongoDB.

8.2.4 Analyzer

This module maps to the last steps of our methodology. It handles queries and correlates information out of the three data sources (AAA, NAT log file and user database) to deliver the result.

A synopsis of the Analyzer module is described as follows:

```
Analyzer(public_ip, beg_time, end_time){
begin
    var myprivate_ip, myphone_number, client_info ;
    NAT_log_file= decompress (NAT_directory) ;
    AAA_log_file= decompress (AAA_directory) ;
    myprivate_ip = getPriv(NAT_log_file, public_ip,
    beg_time, end_time) ;
    myphone_number= getPhonenumber(AAA_log_file,
    myprivate_ip, beg_time, end_time) ;
    client_info = getClientinfo(User_DB, myphone_number) ;

    return client_info ;

end
```

}

A pictorial representation of the whole analyzer process is provided below:

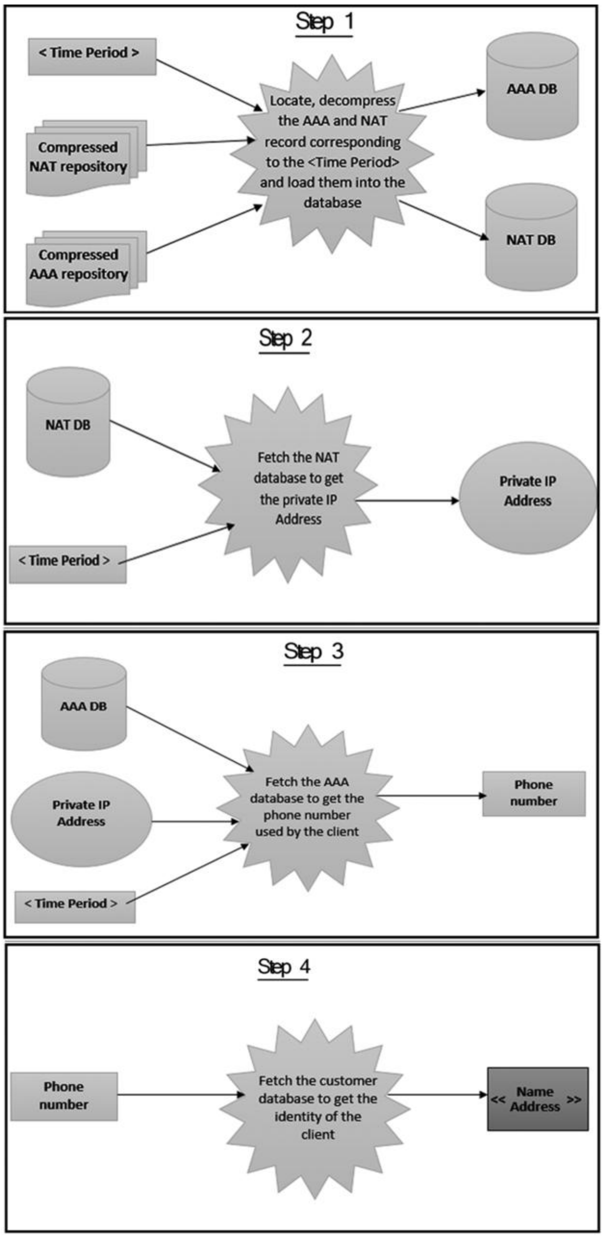

Step 1
< Time Period >
Compressed NAT repository
Compressed AAA repository
Locate, decompress the AAA and NAT record corresponding to the <Time Period> and load them into the database
AAA DB
NAT DB
Step 2
NAT DB
< Time Period >
Fetch the NAT database to get the private IP Address
Private IP Address
Step 3
AAA DB
Private IP Address
< Time Period >
Fetch the AAA database to get the phone number used by the client
Phone number
Step 4
Phone number
Fetch the customer database to get the identity of the client
<< Name Address >>

Figure 4 : Analyzer process

The public IP address, the time interval defined by the **beg_time** and **end_time** are passed into the Analyzer module which processes them as follows:

Firstly since the AAA and NAT files are compressed after being collected, they cannot be fetched. With regards to that, the **decompress** function of the **archiver** module has to locate the appropriate NAT log file and AAA log file and decompress them so that they can be loaded into the database to be fetched.

The **getPriv** function fetches the NAT log files to get the private IP address that was translated to the public IP address public ip during the period **beg_time – end_time**. That private address **myprivate_ip** and the period (**beg_time – end_time**) are then passed as parameters to the **getPhonenumber** function to fetch the AAA log files and output the phone number **phone_number** associated with the suspected client. That phone number is then passed in turn to the **getClientinfo** function that fetches the client database **User_DB** to output the information of the suspected client.

8.3 Technical environment

The said ISP has more than 3 millions clients and uses the CDMA/EVDO technology to deliver telephony and Internet. For security purpose, we will not give details of its network architecture. To develop our solution we used:

- A server with the following characteristics: 16 GB RAM, 1To Hard disk, Intel Xeon 1.87Ghz *16
- **Netbeans 7.3**: Netbeans is a popular free IDE (Integrated Development Environment) that supports many languages like Java and PHP ;
- **Java** ;
- **Tomcat 7.0.34.0** ;
- **Syslog-ng**: syslog-ng is an open source implementation of the Syslog protocol for Linux and Unix-like systems. It extends the original syslogd model with content-based filtering, rich filtering capabilities, flexible configuration options and adds important features to syslog, like using TCP for transport.

9 Results

Requests submitted by law enforcement officers usually state the IP address and the time interval during which the crime is committed henceforth denoted time interval. The performance of our system was later evaluated by measuring the time it takes to provide a response to a given request (processing time) over several time intervals of varying widths. This exercise was repeated daily over a period of one month for the following time intervals: 1 min, 20 min, 1 hour, 6hours, 12 hours, 18 hours and 24 hours. The average of each time interval over a period of one month is provided in table 5 and figure 5 below:

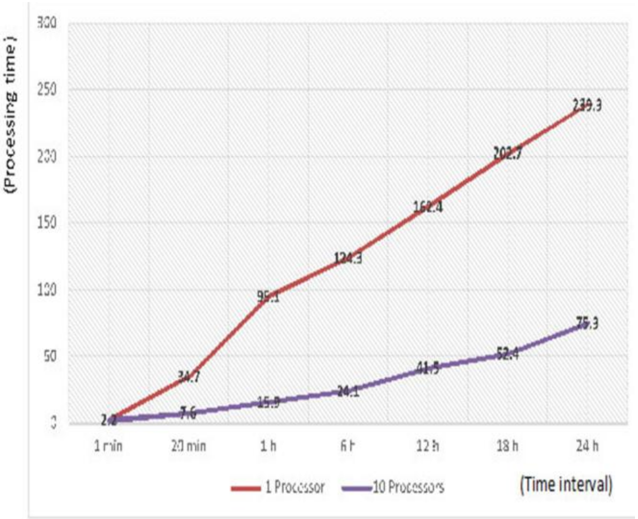
Table 5: Time it takes to deliver the results per time interval

Number of processors used	**Time interval width**						
	1 min	20 min	1 h	6 h	12 h	18 h	24 h

1	2.2 s	34.7 s	95.1 s	124.3 s	162.4 s	202.7 s	239.3 s
10	2.2 s	7.6 s	15.9 s	24.1 s	41.5 s	52.4 s	75.3 s

The figure below (figure 5) depicts a pictorial representation of the results presented in the table above. It shows the evolution of the time it takes to process a request (processing time) with the width of time interval, during which the crime is committed (time interval).

Figure 5: Time spent to deliver the result per time interval



In a bid to ascertain the impact of parallel computing on the results obtained in terms of the processing time, the aforementioned tests which were initially carried out using one processor were later conducted using 10 processors. The results obtained revealed that the processing time witnessed a significant improvement (4 times) with 10 processors. Some screen captures of the application are also presented in the annexes.

10 Conclusion and future work

Due to the omnipresence of Internet in our daily lives (e-commerce, e-government, etc.) and the surge in cybercrimes, Internet traffic record collection and analysis has become an important issue for law enforcement officers. Although some standards have been designed to describe the architecture of a system for collecting the aforementioned data, they fail to give practical and technical specifications to design this kind of system.

Internet traffic collection and analysis is made complex by some issues including: the usage of NAT protocol by ISP especially those of developing countries, the usage of DHCP server for dynamic allocation of private addresses, the scattering of network data among different equipments, and the huge number of Internet user connections that raise the issue of BigData storage and analysis optimization.

Although some private companies have designed systems for that purpose, they are very expensive especially for developing countries. Thus the aim of this paper was to propose an approach that can help developing countries design their own systems with open source technologies at a very affordable price.

In this light, we present a methodology comprising five steps for designing a system for the collection and analysis of Internet traffic record. The methodology was later illustrated by the system we developed and experimented in one of Cameroon’s ISPs. It’s worth mentioning that the methodology can adapt to any ISP architecture. To develop our prototype, we used several concepts and technologies including Non-relational databases, compression algorithms and parallel computing.

Though the methodology proposed was specifically designed for Internet traffic record, it can be adjusted to cope with Phone call data (SMS, Call Data Record) of 2G/3G/4G networks.

Future work can include the design of a predictive model to compute customer’s phone call and Internet activities combine with their geolocation in order to assess several relevant information including people’s social habit and relationships, the dispersion of people around the country/town at different times which can be useful when predicting the way an epidemic can spread, or the way a city can expand considering the high proportion of people who own a phone or smart device (smartphone, tablet, etc.).

Annex

Annex 1: Screen capture of our system

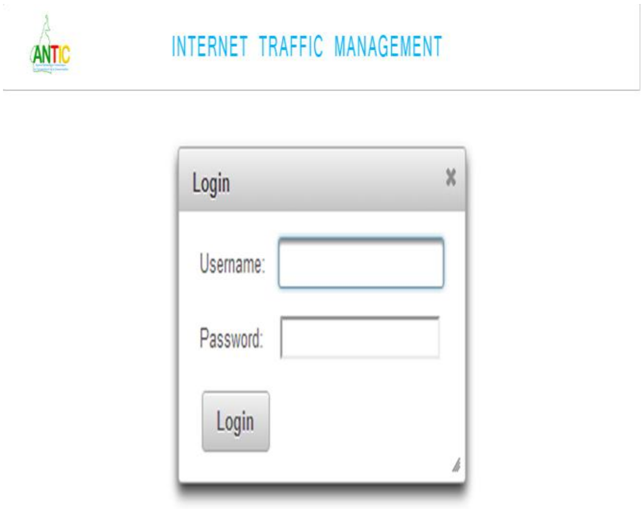
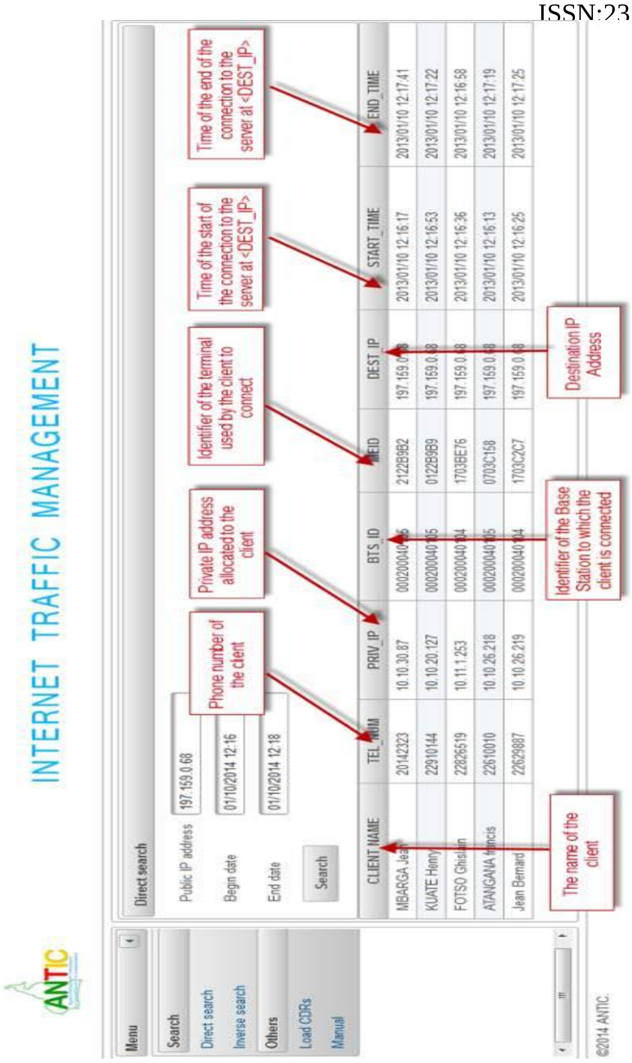


Figure 6: Login screen


INTERNET TRAFFIC MANAGEMENT
ANTIC
Menu
Search
Direct search
Inverse search
Others
Load CDRs
Manual
Direct search
Search
Phone number of the client
Private IP address allocated to the client
Identifier of the terminal used by the client to connect
Time of the start of the connection to the server at <DEST_IP>
Time of the end of the connection to the server at <DEST_IP>
Destination IP Address
Identifier of the Base Station to which the client is connected
The name of the client
©2014 ANTIC.

Public IP address: 197.159.0.68

Begin date: 01/10/2014 12:16

End date: 01/10/2014 12:18

CLIENT NAME	TEL_NUM	PRIV_IP	BTS_ID	MEID	DEST_IP	START_TIME	END_TIME
MBARGA Jean	20142323	10.10.30.87	000200040105	2122B9B2	197.159.0.68	2013/01/10 12:16:17	2013/01/10 12:17:41
KUATE Henry	22910144	10.10.20.127	000200040105	0122B9B9	197.159.0.68	2013/01/10 12:16:53	2013/01/10 12:17:22
FOTSO Ghislain	22826519	10.11.1.253	000200040104	1703BE76	197.159.0.68	2013/01/10 12:16:36	2013/01/10 12:16:58
ATANGANA Francis	22610010	10.10.26.218	000200040105	0703C158	197.159.0.68	2013/01/10 12:16:13	2013/01/10 12:17:19
Jean Bernard	22629887	10.10.26.219	000200040104	1703C2C7	197.159.0.68	2013/01/10 12:16:25	2013/01/10 12:17:25

Figure 7: Search screen

Annex 2 : AAA file structure

N°	Field	Description
1	Streamnumber	Serial number used for identifying a CDR record.
2	AcctType	Accounting status, used for identifying the type of a CDR record.
3	RoamFlag	Roaming flag, used for identifying the roaming type of a user.
4	PaidType	Payment type, used for identifying the charging type of a user.
5	Remark	Reserved.
6	MSID	Mobile station identification.
7	ESN	Electronic serial number. An ESN uniquely identifies an MS.
8	IP Address	IP address of an MS.

9	Network Access Identifier (NAI)	Network access identifier, that is, the account that a user uses to access the Internet. Format : user@domain
10	Account Session ID	ID of an accounting session, identifying an accounting session in a session.
11	Correlation ID	ID of a session, identifying a PPP session of a user.
12	Session Continue	Session stop flag, indicating whether this CDR record is the last CDR record generated in the current session.
13	BeginningSession	Session start flag, indicating whether this CDR record is the first CDR record generated in the current session.
14	MIP Home Agent(HA)	IP address of the home agent (HA).
15	PDSN/FA Address	IP address of the PDSN.
16	Serving PCF	IP address of the PCF that is providing services.
17	BSID	Base station identification.
18	User Zone	User zone.
19	Forward Mux Option	Forward multiplexing option.
20	Reverse Mux Option	Reverse multiplexing option.
21	Service Option	Service option of the air interface of the RASYS.
22	Forward Traffic Type	Forward communications type.
23	Reverse Traffic Type(Primary, Secondary)	Reverse communications type.
24	Fundamental Frame Size	Size of a fundamental channel frame.
25	Forward Fundamental RC	Forward fundamental resource capability.
26	Reverse Fundamental RC	Reverse fundamental resource capability.
27	IP Technology	IP technology type, identifying the IP technology used for calls.
28	Compulsory Tunnel Indicator	Type of a compulsory tunnel.
29	Release Indicator	Cause for sending an accounting stop message.
30	DCCH Frame Format (0/5/20ms)	Format of a frame in the dedicated control channel.
31	Always On	Whether the Always On service is enabled for a user.
32	Data Octet Count (Terminating)	Number of bytes in the IP packet received by a user.
33	Data Octet Count (Originating)	Total number of bytes in the IP packet sent by a user
34	Bad PPP Frame Count	Total number of bad frames discarded by the PDSN due to error rectification failures.
35	Event Time	Time when an event happens, which is recorded by the PDSN.
36	Active Time	Total active connection duration on the traffic channel, in seconds.
37	Number of Active Transitions	Number of times that the status changes from inactive to active in one PPP link.
38	SDB Octet Count(Terminating)	Total number of bytes received by a user through Short Data Burst (SDB) mode.
39	SDB Octet Count (Originating)	Total number of bytes sent by a user through the SDB mode.
40	Number of SDBs(Terminating)	Number of times that a user receives the SDB.
41	Number of SDBs(Originating)	Number of times that a user sends the SDB.
42	Number of HDLC layer bytes received	Number of bytes that the PDSN receives in the high-level data link control (HDLC).
43	In-Bound Mobile IP Signaling Octet Count	Number of bytes in the registration or proxy request message sent by an MS.
44	Outbound Mobile IP Signaling Octet Count	Number of bytes in the registration response or proxy announcement sent to an MS.
45	IP Quality of Service(QoS)	User quality level code of the IP network, identifying the IP service quality of user data.
46	Airlink Quality of Service(QoS)	Quality level code of wireless links, identifying the priority of airlinks of users.
47	Airlink Record Type	Type of a wireless link record.
48	R-P Session ID	ID of an R-P session.
49	Airlink Sequence Number	Serial number of a wireless link.

50	Mobile Originated/Mobile Terminated Indicator	Whether the SDB is initiated or terminated by a terminal.
51	Container	Charging container, consisting of the 3GPP2 VSAs and RADIUS accounting attributes.
52	NAS-Port	Number of the physical port of the PDSN
53	ServiceType	Type of the service requested by a user or provided for a user.
54	AcctDelayTime	Charging delay duration, that is, the time that the PDSN spends in sending an accounting request message.
55	AcctAuthentic	Mode for authenticating users.
56	AcctSessionTime	Duration of an accounting session that is the duration in which a user uses the service. Unit: second.
57	AcctInputPackets	Number of uplink packages, that is, the number of packages that the PDSN receives from the port through which a user uses the service.
58	AcctOutputPackets	Number of downlink packages, that is, the number of packages that the PDSN sends to the port through which a user uses the service.
59	AcctTerminateCause	Cause of the session disconnection.
60	AcctMultiSessionID	ID of a multi-session.
61	AcctLinkCount	Number of links in a multilink session.
62	AcctInputGigawords	Number of times that the uplink volume is calculated from zero.
63	AcctOutputGigawords	Number of times that the downlink volume is calculated from zero.
64	NASPortType	Type of the port used for authenticating users through NAS.
65	MDN	The number that a calling party dials to call a local mobile user.
66	Service Reference	For CDMA 1X, this field indicates a referenced ID of a service instance.
67	FLOW_ID parameter	ID of the IP stream.
68	Subnet	Subnet of the HRPD RAN and the sector ID.
69	RSVP_Signaling_Inbound_Count	Number of bytes in the SR VP signaling that an MS sends.
70	RSVP_Signalling_Outbound_Count	Number of bytes in the SR VP signaling that an MS receives.
71	RSVP_Signaling_In_Pkts	Number of packets in the SRVP signaling that an MS sends.
72	RSVP_Signaling_Out_Pkts	Number of packets in the SRVP signaling that an MS receives.
73	Granted QoS Parameters	Quality of Service (QoS) of the IP stream for authentication.
74	Last User Activity Time	Time when a user performs the last activation.
75	MEID	Mobile equipment identifier (MEID) in the format of an ASCII character in string.
76	Foreign Agent Address	IPv4 address of the PDSN CoA contained in the registration request (RRQ) message.
77	CarrierID	ID of visitor carrier that generates CDRs.
78	GMTTimeZoneOffset	Offset between the GMT time and the time on the PDSN.
79	Forward PDCH	Radio frequency configuration of the forward packet data channel.
80	Forward DCCH Mux Option	Multiplexing option of the forward dedicated control channel(DCCH)
81	Reverse DCCH Mux Option	Multiplexing option of the reverse DCCH.
82	Forward DCCH RC	Format and structure of the wireless channel of the forward DCCH.
83	Reverse DCCH RC	Format and structure of the wireless channel of the reverse DCCH.
84	Reverse PDCH RC	Radio frequency configuration of the reverse packet data channel.
85	Hot-Line Accounting Indication	Background system of the Hot-Line session.
86	Flow Status	Status of the IP stream.
87	Remote IPv4 Address Octet Count	IPv4 address and the number of bytes received and sent through the IPv4 address when a user uses the service.
88	Filtered Octet Count(Terminating)	Number of bytes in the IP package that the PDSN receives from the IP network but does not send to a user.
89	Filtered Octet Count(Originating)	Number of bytes in the IP package that the PDSN receives from an MS but does not send to the Internet.
90	PMIP Indicator	Proxy mobile IP address, indicating whether the current data session uses the proxy mobile IP service.
91	SUBSTYPE	Subscription Attribute.
92	Granted QoS Parameters	The granted QoS parameters for the IP flow.

References

[1] Brittan Paul, "Evaluating lossless data compression algorithms for use on mobile.".

[2] Estan Cristian, Keys Ken, Moore David and Varghese George, "Building a better netflow," in *SIGCOMM '04 Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, 34:245-256.

[3] ETSI, "Lawful interception (li);interception domain architecture for ip networks," Technical report, ETSI, 2006.

[4] ETSI, "Lawful interception (li); retained data," Technical report, ETSI, 2007.

[5] ETSI, "Lawful interception (li);retained data handling;handover interface for the request and delivery of retained data," Technical report, ETSI, 2009.

[6] ETSI, "Lawful interception (li);security framework in lawful interception and retained data environment," Technical report, ETSI, 2009.

[7] Network Instruments, "Extending network visibility by leveraging netflow and sflow technologies," Technical report, Network Instruments, 2011.

[8] Bogdan George TUDORICA Ion LUNGU, "The development of a benchmark tool for nosql databases," *Database Systems Journal, 4,* 2013.

[9] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria and Dishant Gosain, "A survey and comparison of relational and non-relational database," *International Journal of Engineering Research & Technology (IJERT), 1,* 2012.

[10] Shrusti Porwal, Yashi Chaudhary, Jitendra Joshi and Manish Jain, "Data compression methodologies for lossless data and comparison between algorithms," *International Journal of Engineering Science and Innovative Technology (IJESIT), 2,* 2013.

[11] Robert Lourdusamy Senthil Shanmugasundaram, "A comparative study of text compression algorithms," *International Journal of Wisdom Based Computing, 1,* 2011.

[12] Christof Strauch, *Nosql databases.*

[13] I Made Agus Dwi Suarjaya, "A new algorithm for data compression optimization," *(IJACSA) International Journal of Advanced Computer Science and Applications, 3,* 2012.

Biography

Dr. EBOT EBOT ENAW obtained his B.Eng hons degree from Liverpool University in Electronic Engineering in 1986. He later obtained an M.Eng degree in Telecommunication Engineering from The University of Manchester England in 1991. He returned home where he was recruited in the University of Yaounde I, as an assistant lecturer. He pursued his university studies and obtained a PhD in Computer Sciences from the National Advanced School of Engineering of the University of Yaounde I, where he is currently a senior lecturer. His area of specialization include: computer network security, cryptography and formal specification and verification; theorem proving and model checking. He has published some research articles in international journals namely *Formal model of a group key agreement protocol. Journal of Computational Technologies, 7(4):4–20, 2002.* In 2006 he was appointed Director General of the National Agency for Information and Communication Technologies Cameroon, a position he occupies till date. Major activities of the agency include amongst others: securing the Cameroon cyberspace through three key services: Computer Incidence Response Team (CIRT), Public Key Infrastructure (PKI) and Computer Security Audits.

Dr. EBOT EBOT ENAW may be reached at ebotenaw@yahoo.com