# Obtaining Digital Evidence from Intrusion Detection Systems

Mboupda Moyo Achille
University of Yaounde I
Faculty of Sciences
Cameroon

Atsa Etoundi Roger
University of Yaounde I
Faculty of Sciences
Cameroon

## ABSTRACT
Intrusion detection techniques have appeared to inspect all of the inbound and outbound network activities, and to identify suspicious patterns that indicate an attack that might compromise an information system. However, related information can be collected so as to supply evidence in criminal and civil legal proceedings. Several works have been carried out in the domain of Intrusion Detection and Prevention System (IDPS) but, none of the resulting models taking into account the possibility to collect intrusion related information in such a way that some of it can be turned in evidence in a proactive digital forensic purpose. In the literature, some authors have mentioned the possibility of re-designing IDPS as sources of evidence but, a formal model has never been proposed. This paper proposes an intrusion detection architecture for digital forensic purposes implemented using SNORT program.

## Keywords
Intrusion detection and prevention system, Digital forensic, Cybercrime investigation

## 1. INTRODUCTION
This Many organizations and public administrations put in place today IDPS in order to prevent attacks on their information system. In fact, once an attacker intends to manipulate data into a computer system, he gathers information about the target computer, probe it for vulnerabilities and attempt to exploit them. After gain unauthorized access into the computer, the attacker escalates from an unprivileged account to privileged account. He hides tracks and instantiate a persistent reentry. Next, he can extend unauthorized access to other areas of the network and pursue goal of it intrusion which can include stealing information or destroying data.

Largely used, the IDPS appears to be a countermeasure which produces satisfactory results. The IDPS simplifies the task of detecting attacks quite before the actual attack by tracing the trails that the attacker leaves while gathering intelligence about a network. They can be passive (in this case they can only give on alert) or active (when

in addition to alerting, they can react against attacks). The IDPS is the method of identifying unauthorized use, misuse and abuse of computer systems by both system insiders and external attackers. Basically, there are three steps in the process of intrusion detection and prevention which can be decline to monitoring and analyzing traffic; identifying abnormal activities; assessing severity and raising alarm [1,2].

When an IDPS detects an intrusion, it will actually log the event, store relevant data or traffic, notify an administrator and in some cases it will intervene. If it is exploited, the consequently stored data and the logs can be valuable forensic information that may be used as evidence in a legal case against the attacker. In fact, forensic computing appears to identify, preserve, analyze and present digital evidence in a manner that is legally acceptable or accepted in a court. Digital forensic is not a single activity, but draws upon many disciplines [3,4]. It involves the application of information technology to the search for digital evidence either by media and electronic device analysis, network intrusion or misuse detection, or data interception.

The IDS/IDPS becomes today a very useful source of information about an attack. However, they are not originally designed to collect and preserve the integrity of the type of information required to conduct law enforcement investigations. In the course of numerous digital forensic operations, it has been establish that IDS are useless to the investigators whereas, they constitute an important source of information. This is due to the fact that the integrity and the authenticity of information that come thereof are not guaranteed [5]. To face this difficulty, it is necessary to put in place a means of data collection (following a chain of custody) which can produce the first aspects of inquiry in case of investigation. Yuill et al [6] state that IDS can collect enough information during an on-going attack to profile or to identify the attacker. Our aim in this paper is to provide a model of IDS, combined to digital forensic primitives which can proactively or actively brings out relevant information about an attack without materially affecting it primary mission.

The rest of the paper is organized as follows: section 2 outlines the previous works; section 3 describes the proposed IDS architecture and gives some experimental results; section 4 deals with the conclusion and highlighting some perspectives as future works.

## 2. PREVIOUS WORKS
Over the past years, there have been some controversies about the applicability of IDS to the forensic evidence collection process. Two aspects have essentially emerged. The first one views forensic evidence collection and preservation in the case of a computer or network security incident to be inappropriate for an intrusion detection system. Another perspective submits that the IDS are the most likely candidate for collecting forensically pristine evidentiary data in real or near real time. The main idea was to know whether it was possible to use intrusion detection systems to gather forensic evidence in the case of a detected penetration or abuse attempt [7,8]. Several authors have tried to find relevant contribution to this idea. Many authors have mentioned the possibility to

rebuilt IDS in such a way that its output can serve as evidence in a court of law.

[9] address IDS and a view to its forensic applications. They view a forensic application within the framework of intrusion detection and detail the advantages and disadvantages of some IDS. They point that IDS are the places to look for evidence during an investigation process besides Hard drives, Memory, System logs, Email servers, Network traffic.

[6] puts in place a formal descriptive method named Investigative Intrusion-Detection. They show that IDS can collect enough information during an on-going attack to profile, if not to identify, the attacker. The ability of IDS to gather significant information about an attack in progress without materially affecting the primary mission of the intrusion detection system suggests that IDS could be deployed that would provide both detection/response and forensically pristine evidence in the case of a security incident. They focus on attacker activities concerning what he has done, what he can do, what he does, what he knows, what he wants, and what identifies him.

[10] states that although the main aim of IDSs is to detect intrusions to prompt evasive measures, a further aim can be to supply evidence in criminal and civil legal proceedings. However the features that make an IDS product good at providing early warning may render it less useful as an evidence acquisition tool. But, he gives direction and condition to Re-designing IDSs as sources of evidence before concluding that if logs are to be produced from IDS tools, a prosecutor must be prepared to disclose complete details of the tool, and how it was configured and operated.

[7] describes a project which reviews the performance and forensic acceptability of several types of intrusion detection systems in a laboratory environment. He develops a theoretical model and architecture for an intrusion detection system that can also perform forensic tasks. This theoretical model also concerns the case of host based intrusion detection systems.

[11] states that IDS belong to the set of log records along the path. They show that log records contain a substantial amount of content that may be relevant in a criminal case. The log records may reveal identity information that connects the activity to user attributes, including the IP address used and the type of operating system, browser, and applications of the computer user. Logs are timestamp-centric, making them ideal for filling in time line gaps in an investigation. But they also address the fact that log records, like other forms of electronic evidence, can be modified by a third party, but they precise that it would be highly improbable that all the log records along the path of transmission could be altered because each of the devices creating log records would have to be compromised to some degree. This brings out the fact that proofs from IDS are not sufficient enough to accountability; they must be backed by proofs from other sources.

[12] proposed a digital forensic investigation process model including proactive, active and reactive processes. They claimed that this model can be used in a proactive way to identify opportunities for the development and deployment of technology to support the work of investigators, and to provide a framework for the capture and analysis of requirements for investigative tools, particularly for advanced automated analytical tools. The authors implemented a digital forensic model which is divided into three components: The Proactive digital forensics (ProDF) component, the Reactive digital forensics (ReaDF) component and the Active digital forensics (ActDF) component. So doing, they stated that the ProDF component is the ability to proactively identify, collect, gather an event, preserve and analyze evidence to detect an incident as it eventually occurs.

In addition, an automated documentation is generated for a later investigation by the active and reactive components. The evidence that will be gathered in this component is the proactive evidence that relates to a specific event or incident as it occurs. The ProDF as described in [12] can be efficiently associated to IDS to ensure the integrity of evidence and preserve it in a forensically sound manner. Furthermore, the analysis of the evidence will be done in such a way that it can enable prosecution of the suspect and admission to the court of law. Phases under the proactive component fall into Alert, Identification, Collection, Preservation, Analysis and Documentation.

# 3. INTRUSION DETECTION SYSTEM AND DIGITAL FORENSIC: MODEL BUILDING

## 3.1 Output of IDS

Depending on the precise IDS, its outcomes can include [10]:
- The skill to react in a promptly manner to prevent or to reduce substantive damage by automatic or manual intervention;

- The skill to identify an attacker or an activity which can cause more serious attack;

- The skill to discover new attack patterns or as a preventive measure, to provide an additional measure of system protection beyond that available from other forms of security measure.

During our experimentation, it appears that SNORT saves many messages under */var/log/snort* direction. These messages contain relevant information about an incident whenever it occurs, depending on some specified rules indicated in Snort source code. Information concerns Time/date, Source IP address, Destination IP address, Time to Live (TTL) value in the IP packet header, the Type Of Service (TOS) value in the IP packet header, length of IP packet header, total length of IP packet, ICMP Type field, ICMP code value, IP packet ID, Sequence number, ICMP packet type [13].

Unfortunately, the repository where SNORT keeps relevant data is not secure. The data integrity can easily be compromised by an attacker. Furthermore, IDS evasion techniques can also be used to compromise data or to make IDS inefficient.

## 3.2 Requirement of evidence in court

Evidence is used to establish the truth of a particular fact or state of affairs. Generally, evidence has to satisfy tests of admissibility and weight. For admissibility, evidence must conform to certain legal rules which are applied by a judge [14]. For weight, evidence must be understood by, and be sufficiently convincing to the court, whether there is a jury or a judge acting as a trier of fact. Before a court, evidence can be real, testimonial, documentary, expert or derived [9]. Therefore, to be accepted in a court, there should be a clear chain of custody or continuity of evidence and the forensic method used needs to be transparent, that is, freely testable by a third party expert. Anyway, before a court, the prosecutors

need to demonstrate that an information system was involved, it was accessed, such accessing was unauthorized access knew at the time that the access was unauthorized. Nevertheless, a clear chain of custody will be respected if the following basic principles for evaluating the acceptability of evidence as describe by [10] is applied:

- Authentic: the evidence should be specifically linked to the circumstances and persons alleged, and produced by someone who can answer questions about them. Unless a party shows that the evidence is what that party claims it to be, the court will view the evidence as irrelevant.

- Accurate: the evidence should be free from any reasonable doubt about the quality of procedures used to collect the material, analyze the material if that is appropriate and necessary, and finally, to introduce it into court – and produced by someone who can explain what has been done.

- Complete the evidence should be able to tell, within its terms, a complete story of a particular set of circumstances or events.

Among all, other sources can be used to support some given evidence. These can be extract in Firewall Logs, Web Server Access Logs, Simple Mail Transfer Protocol / Internet Message Access Protocol Servers (email), FTP Servers (file transfer protocol), Proxy Server Logs, Secure Shell Servers (remote access), Routers and Switches, Chat Servers, DNS Servers (Domain Name System), Victim and Attacker Systems.

## 3.3 Bridge between IDS outputs and DF evidence

From an evidential point of view, what one looks for is something one can demonstrate to others long after the event itself is over. IDS provide it through logs of various kinds. These include system, audit, application and network management logs. Other sources of potential evidence are network traffic capture and contemporaneous manual entries [15,16]. However, the derived data can be split into a form in which it is easier to analyze and understand. Otherwise, to be admissible in a court of law, the collection of potential evidence will respects a chain of custody.

Thus, outcomes of IDS will be efficient enough to persuade a third party. So doing, logs issued by IDS which intend to provide relevant information for digital forensic purpose must respect the following specification [10]:

- the logs may not have been compromised during or prior to collection as potential evidence and during post collection analysis;

- in the case of real-time network, monitoring the network location of the device hosting the monitoring tool may be such that it is able to capture all relevant traffic, some of the packets using other routes;

- in the case of real-time monitoring, the monitoring tool may be able to keep up with the stream of traffic with which it is expected to deal;

- the logs may sufficiently distinguish between a legitimate and an unwanted access; the logs may exist over a sufficient period of time for comparisons of normal and abnormal activities to be made;

- the logs may be helpful to identify the perpetrator in any useful way, complete for the relevant period of time, rich in detail;

- the logs will gather relevant information.

Hence, the contribution the IDS can make in case of prosecution is to prove that an information system was involved and was accessed. Then, digital forensic will subsequently bring out sufficient legal evidence and will allow to investigate in order to identify the perpetrator. One should note that current IDS are not fully designed to collect and protect the integrity of all information require to conduct investigations in respect of law enforcement. Basically, there are two broad categories of analysis performed to look for signs of intrusion. The first is misuse detection. It works by looking for known indications of misuse, whereas the basis for allowable activity is specified in the security policy of an organization. The second type of analysis performed is anomaly detection. It works by defining parameters for normal activity for a given set of resources [17,18]. This defined normal activity becomes a baseline against which all activity is measured. Actions falling outside the scope of normal activity are flagged as anomalous for investigation as potential security violations.

The implementation of the conditions listed above allows us to define a model of detection (Fig.1). This model is primarily supported on the basic model of intrusion detection but has the particularity of being able to produce information that can serve as digital evidence.

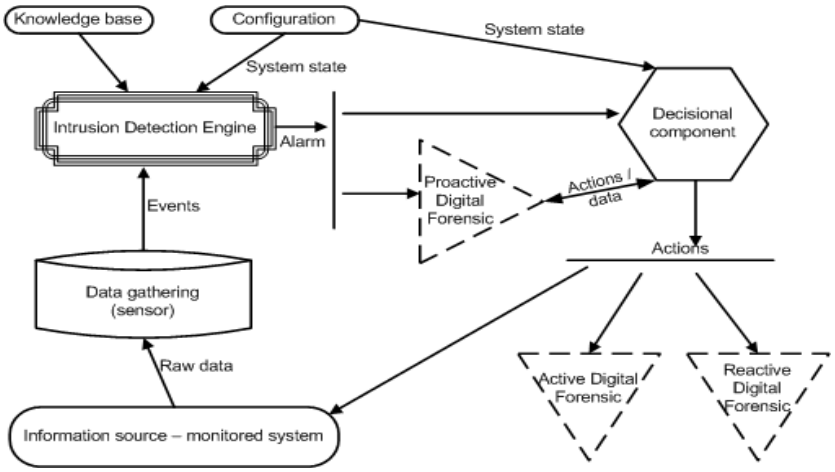


**Fig. 1. The basic IDS architecture for digital forensic purpose**

The proposed model in figure 1 above fall in 9 components which are described as follow:

- Sensor: it is a data gathering device which is responsible for collecting raw data from a monitored system;

- Intrusion Detection Engine: this engine processes the data collected by sensors to identify intrusive activities;

- Knowledge base: it contains information collected by the sensors, but in preprocessed format such as knowledge base of attacks and their signatures, filtered data, data profiles. This information is usually provided by network and security experts;

- Configuration device: it provides information about the current state of the intrusion detection system;

- Proactive digital forensics (ProDF) component: it allows to ensure successful cost of effective digital investigations with minimal business activity disruption and ensuring that admissible evidence and sound processes are in place and available when needed for an investigation or as required during the normal flow of business [12,19]. Each time an alarm is triggered, this component start the collection of all information related to the intrusive activity. It actively safeguards the integrity of collected information and preserves it in a forensically sound manner;

- Decisional component: it initiates actions when an intrusion is detected. These responses can either be automated or involve human interaction;

- Reactive digital forensics (ReaDF) component: it targets the traditional digital forensic investigation that will take place after an incident had been detected and confirmed. This involves identifying, preserving, collecting, analyzing, and generating the final report. This module is active when an attack could not be detected via the intrusion detection engine;

- Active digital forensics (ActDF) component: it allows to gather (identify, collect, analyze and preserve) receivable digital evidence in a live environment to facilitate a successful investigation. When the alert is enabled, the response component triggers the ActDF component throughout the duration of the attack.

## 3.4 Experimental setting: the place of the IDS in Network Topology

Our experimental device consists of one router and six workstations. The router is connected to the internet and the workstations are set to the local network. In order to detect only external intrusion activities, the intrusion detection system was placed directly inside the router as shown in figure 2.

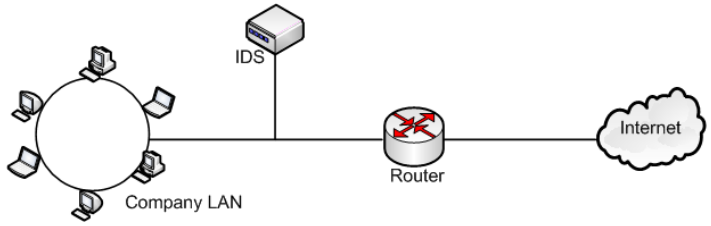


**Fig. 2. Experimental network architecture**

## 3.5 Experimental setting: the updated architecture of SNORT

Snort is known to be a powerful application. This software is free and it can run either on Linux or Windows environments. Understanding the functioning of the internal components of Snort helped us to customize it to our network and helped us to avoid some of the common Snort pitfalls. Snort can be divided into five major components that are each critical to intrusion detection (Fig.3). The first is the packet capturing mechanism. Snort relies on an external packet capturing library to sniff packets from the backbone. After packets have been captured in a raw form, they are passed into the packet decoder. The decoder is the first step into Snort's own architecture. The packet decoder translates specific protocol elements into an internal data structure. Once the initial preparatory packet's capture and decode is completed, traffic is handled by the preprocessors. Any number of pluggable preprocessors either examines or manipulates packets before handing them to the next component: the detection engine. The detection engine performs simple tests on a single aspect of each packet to detect intrusions. The last component is the output plug-in, which generates alerts to lay out suspicious activities. In order to collect digital evidence, some codes have been added in the *snort.conf* file. Therefore, the *snort.conf* file have been implemented in such a way that once the alert is triggered, the incriminated packets are simultaneously preserved in the binary database of snort logs files, and converted to serve as input to the Proactive Digital Forensic component. Figure 3 shows a simplified graphical representation of the dataflow.

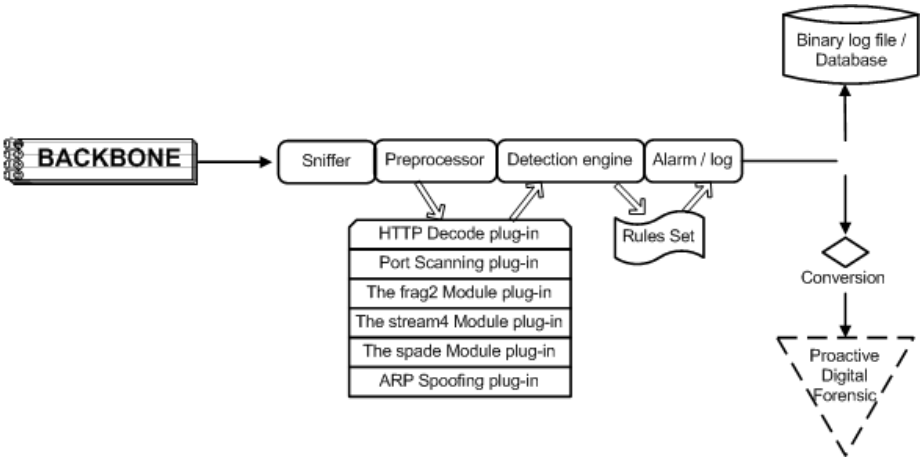


**Fig. 3. SNORT architecture for digital forensic purpose**

## 3.6 Experimental setting: the algorithms for a new implementation of *snort.conf* file

Snort is a lightweight but powerful tool for detecting malicious traffic on a given network. With a flexible and robust rules definition language, Snort is capable of detecting nearly any threat that crosses the network. However, reporting is not its strength. It records tens or hundreds of thousands of suspicious events every day on a busy network. Snort has been made valuable by reviewing and acting on the data it produces. So doing, using the following algorithms derived from the Proactive Digital Forensic component [12], the *snort.conf* file have been modified in such a way that it can produce evidence.ids as another output.

*Algorithm1 Identification algorithm*
*REQUIRE A set of traffic transmitted by the detection engine of Snort /*an Alert file*/;*
*ENSURE All information related to an incident;*
*begin*
*initialisation;*
*repeat*
*select an alert;*
*extract all relevant information that characterizes this alert;*
*create or update the temporary identification file;*
*until (there are no more alerts)*
*end*

*Algorithm2 Collection algorithm*
*REQUIRE temporary identification file*
*ENSURE temporary collection file*
*begin*
*initialisation;*
*repeat*
*select an alert in the temporary identification file;*
*extract source IP address;*
*retrieve information associated with the source IP address;*
*make a record referenced by the source IP address;*
*create or update the temporary collection file;*
*until (the end of the temporary identification file)*
*end*

*Algorithm3 Preservation algorithm*
*REQUIRE temporary collection file*
*ENSURE temporary collection file*
*begin*
*initialisation;*
*protect temporary collection file;*
*save temporary collection file;*
*end*

*Algorithm4 Analysis algorithm*
*REQUIRE temporary collection file, temporary identification file*
*ENSURE temporary analysis file,*
*begin*
*initialisation;*
*identify the rule that has triggered the alert;*
*categorize the attack;*
*indicate the nature of the attack;*
*specify the IP source address;*
*specify the IP destination address;*
*indicate the connection ports used;*
*indicate the timestamp;*
*indicate the protocols used;*
*update the temporary analysis file;*
*end*

*Algorithm5 Documentation algorithm*
*REQUIRE temporary analysis file*
*ENSURE evidence.ids*
*begin*
*initialisation;*
*sort the temporary analysis file by type of attack, source IP, destination IP, protocol, port, and time stamp;*
*create or update evidence.ids file;*
*end*

*Algorithm6 Proactive Digital Forensic /*(ProDF) Documentation algorithm */*
*REQUIRE evidence.ids*
*ENSURE evidence.ids*
*begin*
*initialisation;*
*run identification;*
*run collection;*
*run preservation;*
*run analysis;*
*run documentation;*
*update evidence.ids;*
*end*

The *evidence.ids* file that is outputted by ProDF component contains digital evidence, while *Alert file* is the set of derived useful information that constitute a chain of evidence (Time/date, Source IP address, Destination IP address, Time to Live (TTL) value in the IP packet header, the Type Of Service (TOS) value in the IP packet header, length of IP packet header, total length of IP packet, ICMP Type field, ICMP code value, IP packet ID, Sequence number, ICMP packet type).

## 3.7 Results

To complete our experience, a Honeypot have been deployed in our network to prosecute hackers by gathering evidence of their activities. It is a system used to lure hackers by exposing known vulnerabilities deliberately. The honeypot had among others some services running on it such as Telnet server (port 23), Hyper Text Transfer Protocol (HTTP) server (port 80), File Transfer Protocol (FTP) server (port 21) and others. It was placed somewhere so that the hackers could easily take it for a real server, using an IP address very close to the real server. Attacks recorded at the end of this experience have enabled us to achieve many log files. Figure 4 is a snapshot of one of these log files. The detection of Christmas or XMAS tree attack was the focused case study. A Christmas tree attack sends a large number of Christmas tree packets to an end device. A Christmas tree packet has all the options set so that any protocol can be used. It require much more processing by routers and end devices than other packets.

Large numbers of these packets can use up so much processing power that it ties up these devices effectively making any other task nearly impossible thus denying service to legitimate traffic. Receiving these types of packets is not usual and therefore should be regarded as suspicious. Intrusion detection systems can detect these packets as do some firewalls. An XMAS scan, is a port scan typology with flags set to Fin, Push and Urg at the same packet. The SNORT output file translated with *tcpdump* is shown in Figure 5 below.

```
[cc lang="VHDL"]
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
08:15:22.748471 IP (tos 0×0, ttl 37, id 45231, offset 0, flags [none], proto TCP (6), length 30)
192.168.1.108.33434 > 192.168.1.101.369: Flags [FPU], cksum 0x65ec (correct), seq 30145115847, win 2541, urg 0, length 0
08:15:22.755673 IP (tos 0×0, ttl 121, id 57432, offset 0, flags [none], proto TCP (6), length 30)
192.168.1.101.369 > 192.168.1.108.33434: Flags [R.], cksum 0x87ff (correct), seq 0, ack 30145115848, win 0, length 0
[/cc]
```

```
[cc lang="VHDL"]
09:21:35.873683 IP (tos 0×0, ttl 27, id 50647, offset 0, flags [none], proto TCP (6), length 30)
192.168.1.108.33434 > 192.168.1.101.369: Flags [FPU], cksum 0xdf4a (correct), seq 2649036211, win 2541, urg 0, length 0
09:21:35.886578 IP (tos 0×0, ttl 121, id 51626, offset 0, flags [none], proto TCP (6), length 30)
192.168.1.101.369 > 192.168.1.108.33434: Flags [R.], cksum 0xec4d (correct), seq 0, ack 2649036212, win 0, length 0
[/cc]
```

**Fig. 4. SNORT output files**

In this figure, the current version of SNORT detects an attack which source IP address is 192.168.1.108. The attacker launches an XMAS attack on the destination IP address 192.168.1.101, where the flags FPU are activated.

```
[cc lang="VHDL"]
snort2.9.0.6: listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
[**] [021:02:1] spp_stream5: STEALTH ACTIVITY (nmap XMAS scan) detection [**]
08:15:22.748471 192.168.1.108:33434 -> 192.168.1.101:369
TCP TTL:42 TOS:0x0 ID:55954 IpLen:20 DgmLen:40
**U*P**F Seq: 0x0 Ack: 0x0 Win: 0xC00 TcpLen: 20 UrgPtr: 0x0
```

```
[cc lang="VHDL"]
snort2.9.0.6: listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
[**] [033:03:2] spp_stream5: STEALTH ACTIVITY (nmap XMAS scan) detection [**]
09:21:35.873683 192.168.1.108:33434 -> 192.168.1.101:369
TCP TTL:42 TOS:0x0 ID:55954 IpLen:20 DgmLen:40
**U*P**F Seq: 0x0 Ack: 0x0 Win: 0xC00 TcpLen: 20 UrgPtr: 0x0
```

**Fig. 5. SNORT output files after applying ProDF algorithms**

In this figure, the updated version of SNORT, using Proactive Digital Forensic derived algorithms, reveal a STEALTH ACTIVITY (which is XMAS Scan) on a target computer at 192.168.1.101. Immediately, a file of evidence is built and an instance of this file is given in Figure 6 below.

```
Attack    No.    Time          Source              Destination    Prot Port Info
Xmas_scan 224512 08:15:22.748471 192.168.1.108.33434>192.168.1.101  TCP  80   369 > https [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 224512 09:21:35.873683 192.168.1.108.33434>192.168.1.101  TCP  80   369 > https [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 224512 18:10:22.234231 41.205.86.200>192.168.1.101        TCP  22   45125 > ssh [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 225121 19:15:01.324561 77.175.26.168>192.168.1.101        TCP  80   45631 > http [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 225530 19:25:15.325467 79.233.134.80>192.168.1.101        TCP  53   48123 > domain [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 226910 20:05:18.348854 82.83.205.73>192.168.1.101         TCP  443  48032 > https [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
Xmas_scan 227011 20:29:09.426495 83.163.68.56>192.168.1.101         TCP  111  49569 > sunrpc [FIN,PSH,URG] Seq=0 Ack=0
Win=1024 Urg=0 Len=0
```

**Fig. 6. SNORT output files containing evidences**

The file of evidence is sorted by type of attack, sequence number, time, IP source, IP destination, protocol, port and other relevant information. The process of collecting these information in respect of the chain of custody, unsure the integrity of the data which can be use in case of legal inquiries. The table I below gives a short description of information found in the SNORT output files:

**Table 1. Logs file description.**

| No. | Field or Activity | Context/Notes |
|---|---|---|
| 1 | 08:15:22.748471 | This is the timestamp of the request, it was made on h 08H 15min 22 sec pm |
| 2 | IP | This are all IP (protocol) related settings |
| 3 | tos 0x0 | Type of service field |
| 4 | ttl 37 which is time to live | Number of hops that the packets have to reach its destination. This indicate throw how many routers the packets should pass, this is for not living the packets travel the net for ever |
| 5 | id 45231 | In a case of hijacking (such as man in the middle attack), the attacker should be able to hack the packet ID and present as a response a packet with the same ID but with malicious data |
| 6 | proto TCP | It is the protocol type. It can be some times UDP or ICMP |
| 7 | length 30 | The length of the TCP packet |
| 8 | 192.168.1.108.33.434 | It is the source IP address and 33434 is the port used by the hacker |
| 9 | 192.168.1.101.369 | It is the destination IP address (The honeypot IP address) and 369 is the port used |
| 10 | Flags [FPU] | It is the TCP flag FPU (Fin, Push or Urg) when running an XMAS scan. It could be [S] to mean an ACK reply from the honeypot, or [R] which means RESET and in this case the connection is reseted, or [F] for finishing a transfer, etc. |
| 11 | cksum 0x65ec | This is the TCP-header check sum of the packet (for checking packet integrity) |
| 12 | seq 3014515847 | The TCP sequence number |
| 13 | win 2541 | The amount that will send before requiring a response from the server |
| 14 | urg | The urgency |

As the illustration shows, log records contain a substantial amount of content that may be relevant in a criminal case. The log records may reveal identity information that connects the activity to user attributes, including the IP address used and the type of operating system, browser, and applications of the computer user. Logs are timestamp-centric, making them ideal for filling in time line gaps in an investigation.

## 3.8 Discussion

The aim of this section is to evaluate the behaviors of the modified IDS. As part of this experiment, a free version of Snort has been used. Thereafter, the architecture of Snort has been changed by implementing ProDF component through the algorithms presented above. The duration of the execution of the IDS in both cases is presented in the tables below.

**Table 2. Running Snort without ProDF component.**

| Test number | Number of receive packets | Number of alert | Ratio (Packet/sec) | Number of packets captured by Snort |
|---|---|---|---|---|
| 1 | 3445263 | 76 | 300 | 3445112 |
| 2 | 9655422 | 102 | 500 | 9655315 |
| 3 | 2712657 | 52 | 200 | 2712645 |
| 4 | 6845795 | 151 | 350 | 6845710 |
| 5 | 8932698 | 134 | 400 | 8932624 |

**Table 3. Running Snort within ProDF component.**

| Test number | Number of receive packets | Number of alert | Ratio (Packet/sec) | Number of packets captured by Snort |
|---|---|---|---|---|
| 1 | 3445263 | 76 | 300 | 3445112 |
| 2 | 9655422 | 100 | 500 | 9655311 |
| 3 | 2712657 | 52 | 200 | 2712645 |
| 4 | 6845795 | 151 | 350 | 6845713 |
| 5 | 8932698 | 136 | 400 | 8932621 |

In the first case, the relevant information to the investigation was housed in the default backup directory of Snort. In the second case, the evidence is found in the evidence.ids file. It is a safe file containing data obtained in accordance with a chain of custody for the preservation and collection of digital evidence. Observing the number of packets received the number of alarms and the number of captured packets in both cases, the gap is negligible. This proves that the IDS Snort although its structure has been modified to output admissible digital evidence, has not seen its performance deteriorate as a tool for detecting intrusions.

# 4. CONCLUSION AND FUTURE WORKS

In this paper, it has been established that the IDS could be used as input to a digital forensics door. To carry out this study, a detailed research and cataloging of prior formal work in forensics and intrusion detection was performed. Next, the

general impact of forensic evidence management on IDS was presented. After analyzing and updating the basic intrusion detection system model, a combined model for intrusion detection in a forensic environment using the multiperspective cybercrime investigation process model was theorized. The designed architecture for IDS in a forensic environment using SNORT has been experimented and it has been showed how log files can be exploited in a forensic purpose. The results obtained in this paper are limited to a Network Intrusion Detection System (NIDS) environment. Be able to generalize a theory that supports intrusion detection and digital forensics in the same system remains a significant challenge. IDS can help investigators during a digital forensic process, but computing forensic cannot rely solely on the IDS otherwise, these would be subject to acute changing that could undeniably devious them to their primary goals.

## 5. REFERENCES

[1] Aleksandar Lazarevic, Vipin Kumar, and Jaideep Srivastava.: Intrusion Detection: A Survey, (2005).

[2] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt.: Network Intrusion Detection. IEEE Network 8, 3, 26–41, (1994).

[3] Rodney McKemmish.: What is Forensic Computing? Australian Institute of Criminology. http://books.google.pt/books?id=NoqGmgEACAAJ, (1999).

[4] Eoghan Casey.: Digital Evidence and Computer Crime, 3rd Edition, Forensic Science, Computers, and the Internet. Academic PressPrint Book, Baltimore, USA, (2011).

[5] George M. Mohay, Alison Anderson, Byron Collie, Rodney D. McKemmish, and Olivier de Vel.: Computer and Intrusion Forensics. Artech House, Inc., Norwood, MA, USA, (2003).

[6] Jim Yuill, Shyhtsun Felix Wu, Fengmin Gong, and Ming-Yuh Huang.: Intrusion Detection for an On-Going Attack.. In Recent Advances in Intrusion Detection (2002-01-03). http://dblp.uni-trier.de/db/conf/raid/raid1999.html#YuillWGH99, (1999).

[7] Peter Stephenson.: The Application of Intrusion Detection Systems in a Forensic Environment. Executive Office for United States Attorneys, Vol. 59. United States, Department of Justice, Washington, DC 20530, (2011).

[8] Golden G. Richard, III and Vassil Roussev.: Next-generation Digital Forensics. Commun. ACM 49, 2 (Feb. 2006), 76–80. DOI:http://dx.doi.org/10.1145/1113034.1113074, (2006).

[9] Thomas Scaria Nathan Balon, Ronald Stovall. : Computer Intrusion Forensics. (2004).

[10] Peter Sommer.: Intrusion detection systems as evidence. Computer Networks 31, 2324, 2477 – 2487. DOI:http://dx.doi.org/10.1016/S1389-1286(99)00113-9, (1999).

[11] Mark L. Krotoski and Jason Passwaters.: Obtaining and Admitting Electronic Evidence. Executive Office for United States Attorneys, Vol. 59. Washington, DC 20530. http://www.justice.gov/usao/eousa/foia reading room/usab5906.pdf, (2011).

[12] Roger Etoundi Atsa and Achille Moyo Mboupda.: Multi-perspective Cybercrime Investigation Process Modeling. International Journal of Applied Information Systems 2, 8 (June 2012), 14–20. Published by Foundation of Computer Science, New York, USA, (2012).

[13] Rafeeq Ur Rehman.: Intrusion Detection Systems With Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, And ACID. Prentice Hall PTR, Upper Saddle River, N.J. http://isbnplus.org/9780131407336, (2003).

[14] Kristin M. Finklea and Catherine A. Theohary.: Cybercrime: conceptual issues for congress and U.S. law enforcement. United States, Department of Justice, Washington, DC 20530, (2013).

[15] P.Lakshmi Prasanna D.R.Lavanya K.Rajasekhar, B.Sekhar Babu and T.Vamsi Krishna.: An Overview of Intrusion Detection System Strategies and Issues. InternatIonal Journal of Computer Science and technology 2, 4 (December 2011).

[16] Eoghan Casey.: Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. Digital Investigation 1, 1 (2004), 28 – 43. DOI:http://dx.doi.org/10.1016/j.diin.2003.12.002, (2004).

[17] Eugene H. Spafford and Diego Zamboni.: Data Collection Mechanisms for Intrusion Detection Systems. Technical Report. Cerias, Purdue University, 1315 Recitation Building, (2000).

[18] Andrew Case, Andrew Cristina, Lodovico Marziale, Golden G. Richard, and Vassil Roussev.: FACE: Automated digital evidence discovery and correlation. Digital Investigation 5, Supplement, 0 (2008), S65 – S75. DOI:http://dx.doi.org/10.1016/j.diin.2008.05.008 The Proceedings of the Eighth Annual DFRWS Conference, (2008).

[19] Talania Grobler, C. P. Louwrens, and Sebastian H. von Solms. 2010. A Multi-component View of Digital Forensics. In ARES (2010-03-22). IEEE Computer Society, 647–652. http://dblp.uni-trier.de/db/conf/IEEEares/ares2010.html#GroblerLS10