Journal of Computer Science 7 (7): 1094-1104, 2011
ISSN 1549-3636

Towards an Integrated Intrusion Detection Monitoring in High Speed Networks

Hassen Sallay
Department of Computer Sciences,
College of Computer and information Sciences
Imam Mohamad ibn Saud University, Riyadh, Saudi Arabia

Abstract: Problem statement: Security Management has become a critical aspect for large scale distributed systems. Particularly, recent Distributed Intrusion Detection Systems (DIDS) schemes in High Speed Networks (HSN) have raised new serious management problems and challenges. Increasing the effectiveness of IDS monitoring is primordial to satisfy the restrictive constraints in such large multi-domains environment for narrow context of HSN. **Approach:** We consider the intrusion detection monitoring as a two facets entity: one at local level (single domain) and another at the global one (multi-domains). Through the local level, evolution of single domain intrusion detection process (vulnerability data collection, alert generation and sensor configuration according to some improvement scenarios) can be monitored. The global level represents evolution of multi-domain intrusion detection process as well as the eventual security defending process through overall network (policy generation, load balancing operations and global alert correlation). Differentiating these two facets, leads to the design of a scalable intrusion detection management solution. **Results:** The effectiveness of DIDS management in HSN had been studied and an IDS scalable monitoring architecture for multi-domains had been proposed. Several scenarios of Snort IDS showed an improvement on the performance of real-time detection. An integration of a set of tools provided a convivial IDS monitoring platform. **Conclusion:** To satisfy the constraints of Intrusion detection process in term of real-time and efficiency in HSN we need to monitor efficiently the IDS process. In this context, the management framework outlined is more appropriate, convenient and efficient. The herein proposed architecture, the snort IDS improvement techniques and the integrated platform played a crucial role in improving of IDS real-time monitoring.

Key words: Intrusion Detection Systems (IDS), high speed networks, management architecture, PBNM monitoring, Snort Benchmarking, Integrated monitoring

INTRODUCTION

The goal of security management is to control the access to sensitive information and resources based on security policies to prevent and defend against intentional or unintentional attacks in the network. Among the well-known network defending techniques stay the Intrusion Detection Systems (IDS). The IDS scans the incoming or outgoing network traffic in order to detect the malicious or suspicious activities. As networks become faster such that the High Speed Networks (HSN), there is a need for IDS to perform security analysis techniques that can keep up with the increased network throughput otherwise it becomes a network bottleneck. Efficient management of Distributed IDS (DIDS) is both a crucial requirement and a major challenge for security services. Building management solutions which can address scalability, efficiency and real-time constraints for IDS is a key to their successful deployment. But, due to the differences between high speed networks and usual networks, the potentially huge number of packets evolved over time, the design and implementation of the afore mentioned management services are much more complex to achieve.

Related work: Several efforts are driven in the literature in the context of intrusion detection process. (Wang and Liu, 2008) establishes a whole work model on the basis of intrusion detection techniques and proposes to establish a data warehouse of intrusion detection to provide an efficient and stable system. (Roschke *et al*., 2009) presents an extensible IDS management architecture to manage security event and correlation. It is based on the virtualization concept which integrates and handles different types of sensors or collects and synthesizes alerts generated from multiple hosts located in a loosely coupled environment. Yu *et al*. (2004), an intrusion alert management system based on a collaborative architecture design for multiple intrusion detection systems to work together is developed. By aggregating alerts and correlating events based on logical relations they reduce the alert processing overload and generate a global and synthesized alert report. A model, architecture and an implementation of an information manager for IDS using the technology of cooperative Multi-agents Systems and Web Services are presented in (Claudino *et al*., 2006). The main goal of their information manager module is to keep safe and update of information that is necessary for the development of inherent functions of an IDS. The validation and correctness of security policies is discussed in (Blanc *et al*., 2006). The authors propose a global architecture, based on an original meta-policy approach for access control and intrusion detection, to guarantee global security properties. Their main idea is to apply verification techniques on the meta-policy while supporting local updates of the security policy so that the system can verify the respect of global security properties after meta or local modifications of the policy. Stakhanova *et al*. (2009) the problem of the IDS rule conflict is involved. Since any conflict that arises between rules in the signature-based IDS detection could put the system in a vulnerable position, the authors address this conflict in host and network-based intrusion detection and response devices. Besides, they present a rule management framework that allows rule set analysis for potential conflicts. Finally in (Betser *et al*., 2001) the ubiquitous interoperability of intrusion detection components across Internet is presented.

On other hand different standard management frameworks have been proposed and developed in the literature. SNMP and CMIP are generic standard management frameworks both work in a very similar manner. They allow administrators/managers to query agents monitoring devices such that bridges, hubs, routers, network servers (Kim *et al*., 2005). Policy-based Network Management (PBNM) is another emerging and promising technology enabling a manager to specify what he wants to do and the end result, without having to know how to accomplish it for the specific devices. It performs network management based on policies. A policy is a rule governing choices in behavior of the system. The goal is to change system behavior without modifying implementation. PBNM facilitates the dynamic change of behavior of a distributed management system and permits the reuse of the managers in different environments (Kim *et al*., 2005; Ok *et al*., 2006). DMTF defined the DEN approach developed as an extension to Common Information Model (CIM) by adding network devices and services to it. In this approach we use LDAP as new way to store management information. Mainly LDAP is used as a database for the configuration information. It provides secure access and simple querying features based on simple filters. Common Open Policy Service (COPS) defined by IETF is another possible implementation protocol of DEN approach. It is a common protocol between elements and policy server. It sends status updates and requests to remote policy decision server to get back policy decisions as well as it provides mechanisms to push or pull policies (Howes *et al*., 2001; Goncalves *et al*., 2009). The IETF through the IDWG (Intrusion Detection Working Group) defined a general IDS architecture through four boxes; Event-boxes which are the sensors, Database-boxes which store information from Event boxes for subsequent processing by Analysis-boxes to detect potential hostile behavior and Response-boxes to execute an intrusion reaction. Furthermore the IDWG specified Intrusion Alert Protocol (IAP) to facilitate an interoperability which is critical for intrusion detection for large networks.

In terms of management technologies, the active networks, JMX and P2P are the main ones. Active network technology is well-known for flexibility and extensibility. It enables to dynamically configure and operate network nodes. For example we can adapt rapid and online load balancing strategies, activate security components or more generally push event correlation and load balancing algorithms where they are needed (Ladner *et al*., 2007). Java Management eXtension (JMX) Framework can also be used to manage an IDS system. All objects will be defined as MBeans and the service communicates with the management application using Java Remote Method Invocation (RMI). Simple Object Access Protocol (SOAP) is another way to invoke methods on classes and objects that exist on remote servers; it is emerging alternative to other component based languages such as DCOM, CORBA. SOAP is used on top of HTTP and uses XML for the syntax. By using these two ubiquitous technologies, SOAP inherits many advantages among which are interoperability, ubiquity and pervasiveness. The intermediary format generates an overhead in the messaging that cuts down performance and therefore limits its usability for very systems requiring frequent changes and updates. Finally, since the DIDS process needs to share different types of files such that vulnerability report files, security policy rules, intrusion detection rules and attack alerts files and security reports about the network security status, Peer-to-Peer (P2P) technology which employs distributed resources to perform a critical function in a decentralized manner makes easy the exchange of these files and the sharing of the security information with other entities (Sallay, 2009). This will increase the effectiveness of the security monitoring and allows for greater automation to take action in real-time against intruders.

We observe that even if some interesting generic standard management frameworks and technologies exist; they need an instantiation for high speed networks through a specification of a dedicated management framework. The intended framework should couple both technical and management plans. In the technical plan we should: (1) design an underlying scalable and adaptive parallel and distributed IDS architecture, (2) develop algorithms and techniques improving the accuracy of alert generation and correlation. In the Management plan we should design and develop an efficient integrated management platform coupling these aforementioned algorithms and techniques to the underlying architecture. The goal of the work undertaken in our group is to build such a framework to manage DIDS entities. Ben *et al*. (2010a; 2010b), we exposed the different architectures used in High Performance Computing (HPC), the common high-speed networks, the programming models, the communications models, and the communication libraries. Rouached *et al*. (2010), we stated the need of HPC for DIDS and we discussed the design requirements of the system. We studied the mapping of the different requirements over the software and hardware features of HSN. We proposed several recommendations for the design of IDS over HSN, starting from the communication protocol and the programming model that should be adopted, to the way the system should handle the communication flow, the memory management and the data transfer between IDS sensors. We presented a formal based approach to intrusion detection in the context of high speed networks. We proposed the global architecture of our approach and detailed its components. We also showed how the proposed formalism can be used. Sallay *et al*. (2010), a security model as well as an architecture able to perform automated and procedural security safeguards are proposed. Sallay *et al*. (2009), we presented an optimized scalable distributed architecture which is about 10 times quicker than the centralized architecture. The solution is based on switch-based splitting approach that supports intrusion detection on high-speed links by balancing the traffic load among different sensors running Snort placed in each point of access to the Internet. Sallay (2009), we designed and implemented a P2P platform dedicated to share different types of files such as vulnerability report files, security policy rules, intrusion detection rules, attack alerts files and security reports about the network security status, in a decentralized manner making easy the exchange of these files and the sharing of the security information involved by the DIDS process.

MATERIALS AND METHODS

In this study we deal with some management plan issues. The main objective is to increase the efficiency, scalability and the effectiveness of DIDS security monitoring in the context of the HSN. This will allow for greater automation to take action in real-time against intruders. Mainly, this work investigates three specific points (1) DIDS management architecture dedicated to the multi-domain environment (its design challenges, description and implementation). (2) Efficiency improvement of Snort IDS according to four scenarios in order to enhance the real time of IDS process capability. (3) An integrated monitoring platform dedicated to manage IDS process in the context of single domain environment.

The DIDS management framework:

Design challenges: Efficient management of DIDS systems is both a crucial requirement and a major challenge for security services. Building management solutions which can address scalability, efficiency and real-time constraints for IDS is a key to their successful deployment. But, due to the differences of high speed networks with usual networks, namely, the potentially huge number of packets evolved over time, the design and implementation of the afore mentioned management services are much more complex to achieve.

To be efficient in the IDS management context, four factors need to be considered (1) the traffic and attacks behavior (2) the scale factor (3) the degree of specialization for the alert event correlation functions and, (4) the ease of deployment and integration with the legacy. In such an evolving environment, the signaling and management plan loops sometime share very close timing requirements, management being forced to follow near real-time constraints found in the control plane. The problem of scalability is due to the architectural design, its implementation and the associated management solutions.

For management solutions dedicated to DIDS systems, scalability is essentially a gradient of: (1) the size of the traffic and its nature (2) the number of domains to manage, (3) the number of IDS nodes of each managed domains, (4) the number of alert events for each sensor (5) the frequency of change in sensors groups (6) the overhead of management and signaling. The Design of efficient management solutions that can scale up to hundreds of domains and thousands of IDS nodes with millions of security alert events, while maintaining a limited management overhead, remains a very challenging task.

Event correlation functions and load balancing algorithms have to integrate the specific nature of IDS process in the definition of the corresponding management applications and related metrics. For example, it is not an easy task to maintain in real-time an up-to-date central knowledge of the alerts together with their distribution in a multi-domain environment. Thus, if one of the event correlation approaches relies on considering the exact number of alerts, these operations must be traced in a very precise way. Moreover new parameters like the number of sensors, the overall number of domains and sensors in a given network, traffic volume and memory usage within hosts must be considered in the load balancing process for DIDS systems.

To be well integrated, any management solution must provide gateways and interfaces to existent protocols and frameworks and take advantage of the most recent technologies to facilitate their deployment and configuration. Moreover the management platform must be sufficiently open and dynamic to adapt itself to changes and to be capable of supporting new types of management algorithms.

Management architecture: The main concept behind our architecture is to distribute as much as possible both management data and processing units to maintain a high degree of adaptability and ensure scalability. This distribution follows a pattern that enables composition and coupling of operations.

We consider the intrusion detection monitoring as a two facets entity: one at the local level and another at the global one. Through the local level, the evolution of the single domain intrusion detection process (vulnerability data collection, sensor configuration and alert generation) can be monitored. The global level represents the evolution of the multi-domain intrusion detection process as well as the eventual security defending process through the overall network (policy generation, load balancing operations and global alert correlation). Differentiating these two facets, leads to the design of a scalable intrusion detection management solution.

The overhead generated by the transport of management data related to an intrusion detection service can also be reduced by setting the granularity of the data to be exchanged. Thus, only data that is mandatory for a given management task is sent over the network. Management data storage becomes necessary at the places where these data are produced and tasks can be delegated to the various network nodes that participate in the intrusion detection service management chain. In this case, the processing distribution follows the data distribution providing an efficient solution for large security data and processing amount involved by the high speed network environments. The proposed architecture provides a 3-level hierarchy: the top management level, intermediate management level and the end management level. At each level of the hierarchy a dedicated management agent is operational.

At the top management level, a Multi Domain Manager (MDM) is in charge of the multi-domain intrusion detection management activities. The MDM hosts a data storage facility that has the entire data related to the overall intrusion service. At the intermediate level, a Single Domain Manager (SDM) in each network domain manages the local intrusion detection facet of the intrusion detection process. It maintains a local view of attack attempts performed by enemy users and holds the local database populated by the IDS sensors according to specific exchange policy. The SDM maintains an interaction with the underlying sensors as well as an interaction upward towards the MDM. At the end management level, IDS sensors are deployed. To this end, the sensors analyze the traffic and build a database which holds very detailed data about each security alerts.

Efficiency improvement of Snort IDS: To improve snort performance, we defined four main scenarios:

- Service Oriented Scenario: Optimization for a specific network service (i.e., HTTP)
- Platform-Oriented Scenario: Optimization using vulnerability assessment tools' results
- Real-Time-Oriented Scenario: Optimization for a specific HW/SW configuration
- Flow-Oriented Scenario: Optimization using filtered input network data-flow

Service-oriented scenario: We modify the Snort source code and configuration files by removing unused preprocessor for a specific sensor (i.e., HTTP) to treat only target packets. Main preprocessors included in snort-2.9.0 are: FRAG3, STREAM5, HTTP_INSPECT, RPC_DECODE, BO (Back Orifice detection), FTP_TELNET, SMTP, SSH, DCERPC2 (SMB / DCE-RPC normalization and anomaly detection), DNS, SSL. In this scenario, we remove the following preprocessors: FTP_TELNET, SMTP, SSH, DNS.

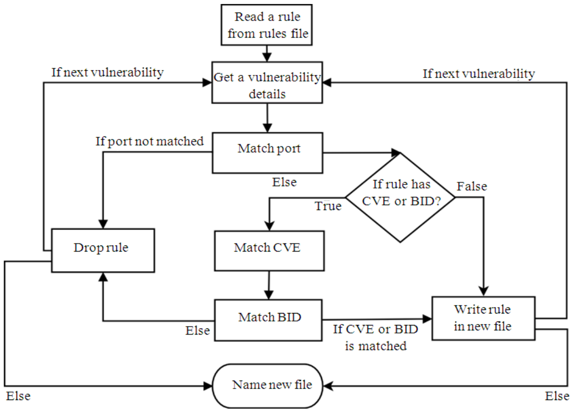


Fig. 1: Snort rules filtering process

Platform-oriented scenario: The main problem in NIDS based-rule is the generation of a large number of false alerts. Optimizing NIDS based-rule by decreasing the number of rules can reduce the amount of alerts, traffic scanning time and increase alerts accuracy. The basic idea of this approach is to efficiently correlate between NIDS rules and vulnerability assessment application report. This report contains information about the network security state such as open ports, vulnerabilities, etc. In this approach we used OpenVas as a Vulnerability Assessment Tool. The Open Vulnerability Assessment System (OpenVAS) is a framework of several services and tools offering a comprehensive and powerful vulnerability scanning and vulnerability management solution. After each network vulnerability a scanning report is generated with important information; host IP, port, some standard references such as CVE (Common Vulnerabilities and Exposures) and BID (BugTraq ID), description of this vulnerability and some information about it. Snort rules filtering process is described by Fig. 1.**Error! Reference source not found.** It is based on two principal parameters (1) Port Number an (2) Vulnerability Description Tag (CVE, BID). For more details about of the filtering process you can refer to ((http://www.amansystem.com.). After this process, a new rule-set is generated and used as input for snort.

Real-time oriented scenario: We plan to apply the OS Tuning to the Linux 2.6.33.7 Kernel. A previous version of the RT kernel patch rt29 was used and results were not accurate (Hard Disk Latency has a big influence on snort processing time, which increases quickly with RT patch activated). We are planning to use the latest version of the RT patch (http://www.kernel.org/pub/linux/kernel/projects/rt/patch-2.6.33.7.2-rt30.gz) released on 12/21/2010. This task is not yet finalized.

Flow-oriented scenario: In this scenario, we realize a benchmarking of snort with a filtered input stream without modifying its configuration nor the operating system. In this scenario, significant parameters are: CPU usage, CPU Time, Memory Usage (Virtual memory, Resident memory, Shared Memory). Other parameters may be added. Some measures were taken but this scenario is not yet finalized.

An integrated platform for DIDS monitoring in a single domain: There are different free and commercial tools used by many network administrators but not suitable in our context since they are not open source. To integrate our architectural improvement our platform is built on an integration of interesting free open source tools such as Snort, Nagios, Base, SnortCenter, Acid, Ignoramuse, and their plugins. The platform detects attacks by managing Snort sensors which analyze the network traffics continuously and convert it into specific events depending on certain rules and policies. Based on sensors collected data, the SDM console produces different types of reports to help the system administrators to specify a certain rules and policies. The administrator can view, add, update and maintain new policies as well as edit the IDS rules. He can prioritize and analyze these events to find the real threats that need taking actions and ignore false positive alerts.

RESULTS

Management architecture implementation: The previously presented architecture can be implemented in several ways. We should choose the appropriate technology for each layer. Mainly we have to distribute the data and the processing between the different architectural level components. The network administrators either those responsible for multi domains management or those responsible for single domain management should have a tools to exchange the needed data to or from the other components in the same or different layer. As mentioned before, the MDM is responsible for the specification and the distribution of the proactive or reactive security polices, define and apply the different event correlation algorithms and load balancing techniques.

For the internal or external attacks, the SDM aggregates and analyses the different alarm to detect an eventual local internal or external multi host attacks. It will execute some event correlation algorithms for that purposes and defending operations. Theses algorithms and operations are dictated by the MDM which define the security policies.

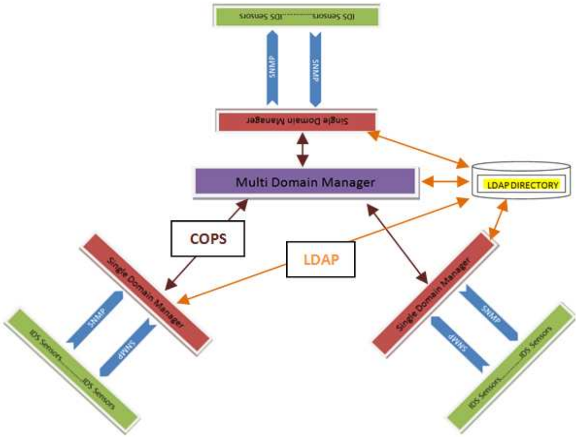


Fig. 2: Policy based management architecture

Thus, depending on the attack nature, the SDM executes an automatic security policy strategy issued from the MDM and sends also a notification to the local network security administrator user to do the right and corrective actions. This model is more scalable than a full source driven polling approach. Moreover it enables MDM to build service level statistics which are specific to the security forensics service (e.g., checking and collecting the attacks evidences) and produces some different security reports about the network security status.

Policy-based Network Management seems to be the appropriate technology to use since it enables manager to specify rules governing choices in behavior of the system without having to know how to accomplish it for the specific devices. Consequently we can change system behavior without modifying implementation and the reuse of the managers in different environments Fig. 2. These characteristics will guarantee the flexibility and the adaptability required by our management architecture. Figure 3 shows our policy based management architecture implementation.

In PBNM framework there are three main components which are: Policy Decision Point (PDP), Policy Consumer (PC) and Policy Target (PT). In our architecture the policy decision point will be the Multi-Domain Manager (MDM). It will make policy decisions based on policy conditions. The policy consumer will be the SDM. It will receive policy and translates it into format applicable to target since it knows about target capabilities. The policy target will be IDS Sensors which will execute the policy itself. The SDM can play the role of policy target for the MDM when it is needed for example if we need to balance load or correlate events between the different SDM in the intranet. Mainly, the management takes the following scenario. The administrator makes a new security policy rule or retrieves existing policy from directory service using LDAP and views or edits policy. He associates the policy with policy targets for example the SDM or IDS sensors. The policy and association with targets is stored in the repository via LDAP. The associated consumer for each target is notified that a new policy is available. The consumer obtains the policy from the repository via LDAP e.g., using query to find the policy. The consumer processes the policy and configures the targets using target-specific mechanism. Finally, for each target which received policy data, the consumer provides status information back to the policy management application. In order to exchange the polices between SDM and MDM we use COPS protocol. It is a lightweight and efficient protocol minimizing the signaling traffic involved by the management process. We can also pre-configure devices with policy data, so they do not have to query the MDM on every event-provisioning. SNMP can be used between the policy target and the policy consumer since the SNMP is commonly used in the network devices. We can also use other appropriate specific tools to manage the policy target depending the platform and sensors used in the single domain intrusion detection system.

Efficiency improvement of snort IDS:

Service-oriented scenario: Hardware used in this scenario is a computer with the following characteristics:

Parameter	Value
OS kernel	Ubuntu 10.10 (maverik) / Linux 2.6.35-22-generic-pae
CPU/ Memory	4 x Intel(R) Core(TM) i5 CPU 760 @ 2.80GHz / 3.9GiB
Hard Disk	500GB ATA WDC WD5000AAKS-00UU3A0

We run this test with the four different streams dumped from the “Capture the Flag” competition held at Defcon conference each year organized and run by the Ghetto Hackers. The traffic generated during DefCon days was dumped using the tcpdump program. For example, the dumped file RootFu!-11 which is a part of the Defcon-11 conference encompasses 10527588 packets in a 3.8 GB file. It contains about 31500 Snort alerts, from which more than 80% is HTTP. We evaluated the following parameters for each test:

- Processing Time in seconds
- Mean value of Processed Packet/s
- TCP Rebuilt packets
- Number of detected POST requests
- Number of detected GET requests

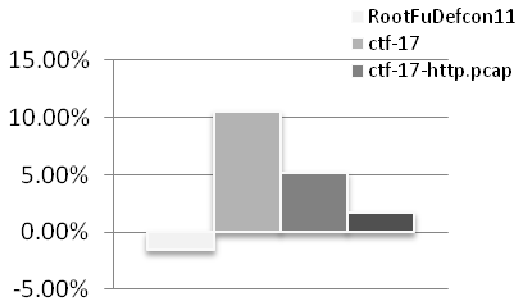


Fig. 3: Gain in processing time for different streams

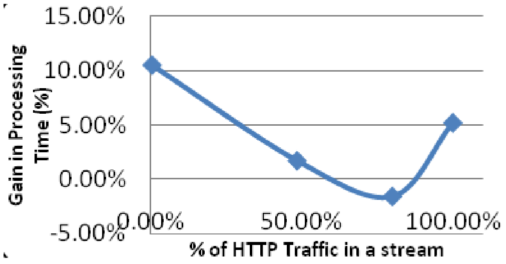


Fig. 4: Gain in processing time in function of % of HTTP packets in a stream

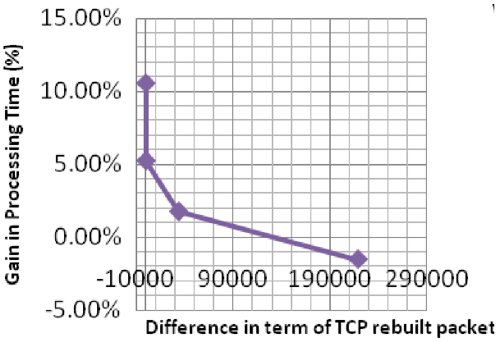


Fig. 5: Relation between Gain in processing time and the difference between TCP rebuilt packets numbers before and after modification

Table 1: HTTP stream

Service Oriented	HTTP Traffic			
Stream	RootFu Defcon11	ctf-17	ctf-17-http.pcap	ctf-08/eth1
Size(KB)	3871988	7598672	60016	2424536
NB Packets	10527588	38994342	459835	12155014
%HTTP Traffic	80.08%	0.79%	100.00%	48.65%

The results for each stream are shown by 0 3-5. Figure 3 shows the gain in processing time for the four different streams. Figure 4 shows the gain in processing time in function of % of HTTP packets in a stream. Figure 5 shows the relation between the gain in processing time and the difference between TCP rebuilt packets numbers before and after snort code modification Table 1. We can observe the following:

- In two of precedent four streams we have a difference in number of processed POST requests and GET requests which indicate that eliminated preprocessors influences snort behavior in term of TCP packet rebuilding and HTTP stream reassembling
- When the number of TCP rebuilt packets increases, the processing time increases and the P.T Gain decreases
- With all streams, except stream 1, we have a gain in processing time
- If we use all four streams results, we can calculate a global gain in processing time weighted by stream size:

$$G = \frac{\text{size}_1 \times G_1 + \text{size}_2 \times G_2 + \text{size}_3 \times G_3 + \text{size}_4 \times G_4}{\text{size}_1 + \text{size}_2 + \text{size}_3 + \text{size}_4} = 5.62\% \qquad (1)$$

Platform-oriented scenario: The following figures (Fig. 6-8) show the difference between Snort performance before and after rules filtering. We evaluated the following parameters for each test:

- Some global statistics before and after rules filtering
- Scanning time before and after rules filtering
- Gain in percentage for some different parameters after rules filtering process

The results are shown by Fig. 6-8. Figure 6 shows the difference in snort alerts when original rules and filtered rules are used. Values used in this comparison are as following:

- Total number of alerts
- Alerts with port number
- Alerts with CVE reference
- Alerts with CVE reference not detected in OpenVas report
- Alerts with port number not detected in OpenVas report

Figure 7 shows the gain in scanning time. We observe a good benefit in term of scanning time. This will increase the effectiveness of the intrusion detection and allows for greater capability to take action in real-time against intruders.

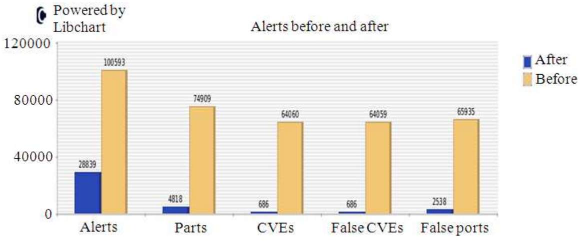


Fig. 6: Global Statistics before and after rules filtering

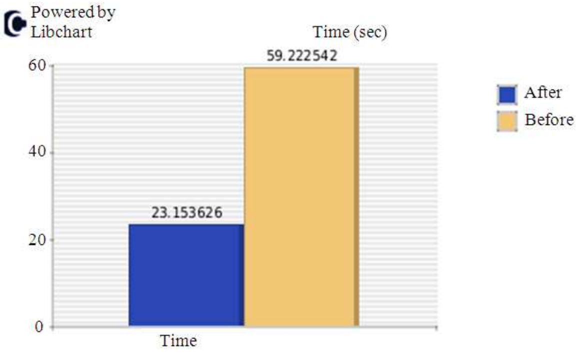


Fig. 7: Snort scanning time before and after rules filtering

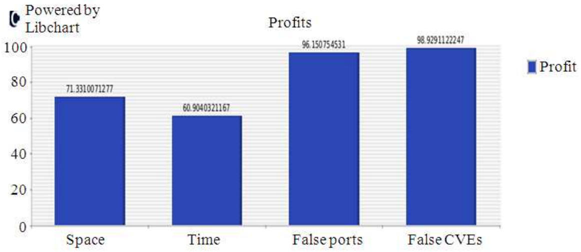


Fig. 8: Gain in percentage for different parameters after rules filtering process

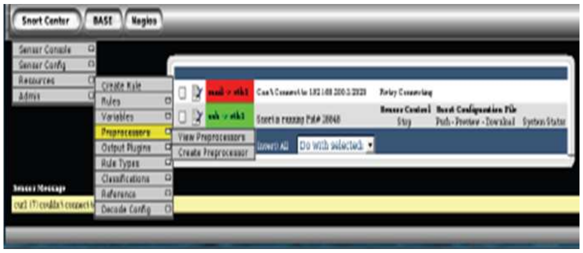


Fig. 9: The integrated platform main menus

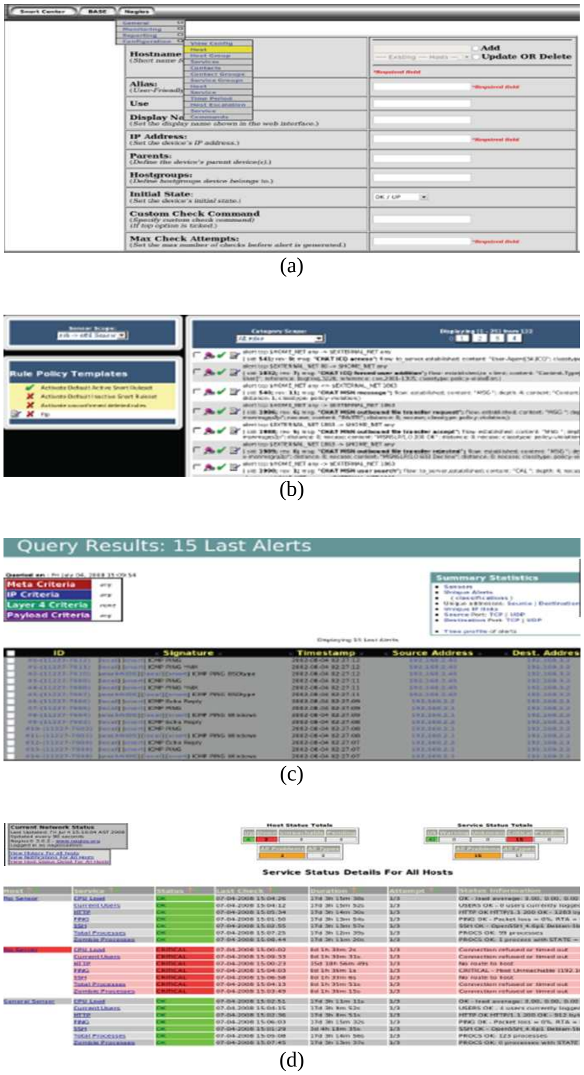
Figure 8 shows gain in percentage in term of:

- Hard Disk Space
- Snort Processing Time
- Number of alerts with ports not in OpenVas report (false ports)
- Number of alerts with CVE reference not in OpenVas Report (false CVE)

An integrated platform for DIDS monitoring in a single domain: The front end of the system is an integration of different tools in a well-designed web page to make managing and controlling the system easier. The main functionalities of the platform GUI are (1) the controlling of the snort sensors and the editing of their rules and configuration files by using SnortCenter, (2) showing the traffic analyzing graphs and producing the generated events in tables by using BASE (3) the network monitoring and checking of network hosts and services by using Nagios and its plugins (NDOUTIL, NRPE, IGNORAMUS). The management of the Nagios configuration files is done via our enhanced Ignoramus plugin to delete and Update the Nagios configurations files. Figure 9 present respectively the operations provided by SnortCenter, Base and Nagios. It is noted that all communication and data transfer between the sensors and the administrator are secure by using SSL and SSH protocols.

Figure 10 represents some policy based management operations provided to administrator to manage the distributed intrusion detection process. Basically the administrator is able specify policies generating alerts when communication is lost between components of network or when the traffic load on a host exceed its capacity and in this case a rule is activated to analyze traffic (Fig. 10a). He can also classify alerts as high, medium or low in real time, block traffic from certain IP address or ports and display the alert status report (Fig. 10b and c). Moreover, He can create reports based on traffic analysis and network monitoring including communication protocol, domain, IP address, port of source, domain, IP address, port of destination, rule violated, type of attack if possible and recommended fixes if any (Fig. 10d).

Another interesting feature for this platform is to provide the administrator with real snapshot of the attacks behavior Based on two criteria which are the IDS characteristics and the nature of the incoming traffic, the administrator will be more comprehensive of security risks and attacks his network is exposed to. Firstly the Snort rules files are studied to extract the most vulnerable ports (Fig. 5 First histogram). This gives the administrator valuable information on which services should be more careful. Secondly we studied the RootFu! dumped traffic which is the descendant of the Capture the Flag competition held at Defcon conference each year organized and run by the Ghetto Hackers.



Hostname

Alias:
(User-Friendly

Use

Display Na
(Set the display name shown in the web interface.)

IP Address:
(Set the device's IP address.)

Parents:
(Define the device's parent device(s).)

Hostgroups:
(Define hostgroups device belongs to.)

Initial State:
(Set the device's initial state.) OK / UP

Custom Check Command
(Specify custom check command)
(If top option is ticked.)

Max Check Attempts:
(Set the max number of checks before alert is generated.)

*Required field

(a)

(b)

(c)

Service Status Details For All Hosts

(d)

Fig. 10: Some IDS Policy based operations

The traffic generated during RootFu! days was dumped using the tcpdump program. For example the dumped file RootFu!-11 which is a part of the Defcon-11 conference encompasses 10527588 packets in a 3.7 GB file. It contains about 31500 Snort's alert. The main objective of this statistical study is to give the administrator the most used ports by the hackers (see Fig 11. RootFu!-11 of DefCon-11). Based on this information the administrator can adapt the IDS rules since he will be more aware of the hacker's behavior. These statistics are performed by a statistical component integrated in the platform which performs several statistics to help the admin to adapt the security policies according to the traffic behavior.

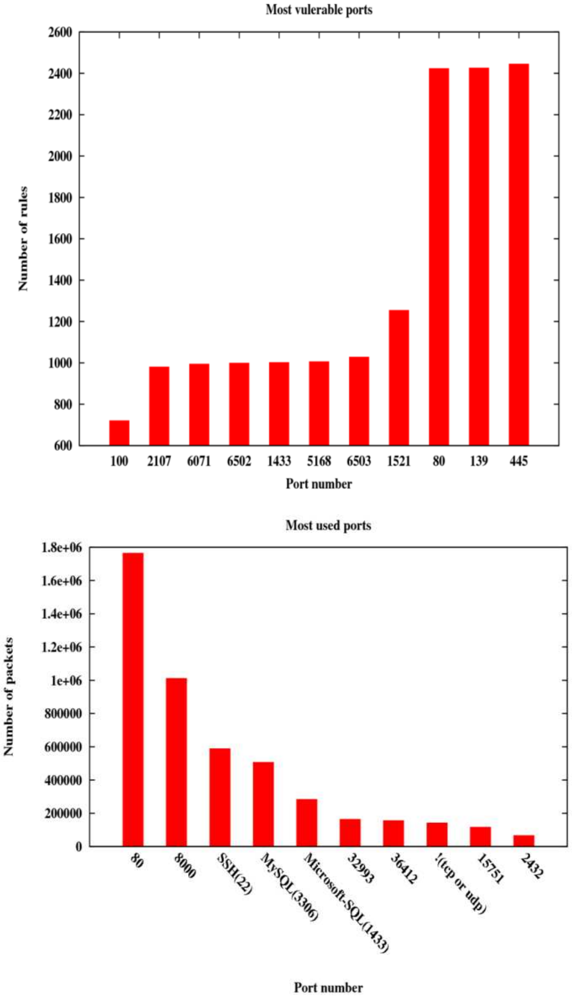


Fig. 11: Most vulnerable ports In Snort Rules and Most used port in DefCon11

In addition to these functionalities the platform specifies in detail the running services on each node in the network (sensors, servers, hosts), provides auto configuration for the network nodes and their service checking, displays them in a map and draws graphs and charts for the analyzed traffic and notifies the administrators of any odd activities via emails and SMS messages if he was far away.

CONCLUSION

Since traditional centralized approaches to traffic analysis cannot scale with the increase of bandwidth advances mainly due to their memory and computational requirements, a number of Distributed Intrusion Detection Systems (DIDS) architectures have been proposed for dedicated network monitoring tasks. However, they remain not scalable in the context of high speed networks. These tasks become more complex if we require a real-time security defending. Improving efficiency, real-time and scalability of IDS in HSN passes through specifying a dedicated integrated security management framework. Such framework makes it easier to work with complex technologies and it ties together a bunch of components into more useful integrated architectural solution as well as it ensures its good and flexible implementation easily tested and debugged.

We presented here an efficient dedicated IDS management framework improving efficiency, real-time and scalability of IDS management in High Speed Networks. A hierarchical distributed architecture and its policy based management implementation are described, some IDS real-time and efficiency improvement are evaluated and an integrated IDS monitoring platform is proposed. This platform can be used in forensic analysis purposes by tracking down the attack activities and intruder violations. As future work, we envisage implementing the overall architecture and extending it to multi domain environment. We intent also to illustrate its applicability and show how the framework can cope with different IDS security alerts correlation techniques and load balancing algorithms.

ACKNOWLEDGMENT

This study is a partial result of a project funded by King Abdul Aziz City for Sciences and Technologies (KACST) in the security research track.

REFERENCES

Ben, F.O., H. Sallay, A. Ammar, M. Rouached, K. Al-Shalfan, M. Ben Saad, 2010a. A Survey on Architectures and Communication Libraries Dedicated for High Speed Networks. accepted to appear COMPUTERS and SIMULATION in MODERN SCIENCE, Volume V, ISI book, 2010.

Ben, F.O., H. Sallay, A. Ammar, M. Rouached, K. Al-Shalfan, M. Ben Saad, 2010b. On Distributed Intrusion Detection Systems Design for High Speed Networks. Proceedings of the 9th international conference on Advances in e-activities, information security and privacy, ISPACT'10, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA., pp: 115-120. http://portal.acm.org/citation.cfm?id=1948855

Betser, J., A. Walther, M. Erlinger, T. Buchheim and B. Feinstein *et al*., 2001. GlobalGuard: Creating the IETF-IDWG Intrusion Alert Protocol (IAP). Proceedings of the DARPA Information Survivability Conference and Exposition, June, 12-14, IEEE Xplore Press, Anaheim, CA , USA, pp: 22-34. DOI: 10.1109/DISCEX.2001.932189

Blanc, M., J. Briffaut, P. Clemente, M.G.E. Rab and C. Toinard, 2006. A collaborative approach for access control, intrusion detection and security testing. Proceedings of the International Symposium on Collaborative Technologies and Systems, May, 14-17, IEEE Computer Society, Las Vegas, Nevada, USA, pp: 270-277. http://www.computer.org/portal/web/csdl/doi/10.1109/CTS.2006.1

Claudino, E.C., Z. Abdelouahab and M.M. Teixeira, 2006. Management and integration of information in intrusion detection system: Data integration system for IDS based multi-agent systems. Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops, Dec. 18-22, IEEE Xplore Press, Hong Kong, China, pp: 49-52. 10.1109/WI-IATW.2006.87

Goncalves, P., J.L. Oliveira and R.L. Aguiar, 2009. An evaluation of network management protocols, Proceedings of the IFIP/IEEE international Symposium Integrated Network Management, June 1-5, IEEE Xplore Press, Long Island USA., 537-544. 10.1109/INM.2009.5188859

Howes, T., M. Smith and G.S. Good, 2001. Understanding and Deploying LDAP Directory Services. 7th Edn., New Riders, USA., ISBN: 1578700701, pp: 846.

Kim, G., J. Kim and J. Na, 2005. Design and implementation of policy decision point in policy-based network. Proceedings of the 4th Annual ACIS International Conference on Computer and Information Science, July, 16-16, IEEE Xplore Press, USA., pp: 534-538. 10.1109/ICIS.2005.46

Ladner, R., E. Warner, U. Katikaneni, F. McCreedy and F.E. Petry, 2007. Active network architecture and management. Int. J. Intell. Syst. 22: 1123-1138. DOI: 10.1002/int.20242

Ok, K.-S., D.W. Hong and B.-S. Chang, 2006. The design of service management system based on policy-based network management. Proceedings of the International Conference on Networking and Services, July, 16-18, IEEE Xplore Press, Silicon Valley, CA., pp: 59-64. 10.1109/ICNS.2006.109

Roschke, S., F. Cheng and C. Meinel, 2009. An extensible and virtualization-compatible IDS management architecture. Proceedings of the 5th International Conference on Information Assurance and Security, Aug. 18-20, IEEE Xplore Press, Xi’an, China, pp: 130-134. 10.1109/IAS.2009.151

Rouached, M, H. Sallay, O. Ben Fredj, A. Ammar., K. Al-Shalfan, M. Ben Saad, 2010. Formal analysis of intrusion detection systems for high speed networks. Proceedings of the 9th WSEAS International Conference on Advances in E-activities, Information Security and Privacy, (ISPACT'10), World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA., 109-114. http://portal.acm.org/citation.cfm?id=1948854

Sallay H., 2009. An Efficient secure manageable P2P Framework. Proceedings of 5th International Computer Engineering Conference, Dec. 27-28, IEEE Press, Cairo, Egypt, 58-62. http://amansystem.com/people/sallay/rr.php

Sallay H., K. Al-Shalfan and O. Benfredj, 2009. A scalable distributed IDS architecture for high speed networks. Int. J. Comput. Sci. Network Sec., 9: 9-16. http://paper.ijcsns.org/07_book/200908/20090802.pdf

Sallay H., K. Al-Shalfan, 2010. A Standard-Compliant Integrated Security Framework, Proceedings of the 9th WSEAS international conference on Advances in e-activities, information security and privacy, (ISPACT'10), World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA, 77-84. http://portal.acm.org/citation.cfm?id=1948849

Stakhanova, N., Y. Li and A.A. Ghorbani, 2009. Classification and discovery of rule misconfigurations in intrusion detection and response devices. Proceedings of the World Congress on Privacy, Security, Trust and the Management of e-Business, August 25-27, IEEE Computer Society, New BrunsWick, Canada, pp: 29-37. http://www.computer.org/portal/web/csdl/doi/10.1109/CONGRESS.2009.12

Wang, W. and X. Liu, 2008. The model design of educational administration safety--based on intrusion detection technology. Proceedings of the International Seminar On Business and Information Management, Dec, 19-19, IEEE Xplore Press, Wuhan, pp: 281-284. 10.1109/ISBIM.2008.204

Yu, J., Y.V.R. Reddy, S. Selliah, S. Kankanahalli and S. Reddy *et al*., 2004. TRINETR: An intrusion detection alert management system. Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, June, 14-16, IEEE Xplore Press, USA., pp: 235-240. 10.1109/ENABL.2004.76