``

IMPLEMENTATION OF AN INTRUSION DETECTION SYSTEM ON MTU NETWORK

By

OLUSEYE-PAUL ISAAC

18010301032

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, COLLEGE OF BASIC AND APPLIED SCIENCES, IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

2022

``

DECLARATION

I hereby declare that this project has been written by me is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

OLUSEYE-PAUL ISAAC

Date

``

CERTIFICATION

This is to certify that the content of this project entitled **'IMPLEMENTATION OF AN INTRUSION DETECTION SYSTEM ON MTU NETWORK'** was prepared and submitted by **OLUSEYE-PAUL ISAAC** in partial fulfillment of the requirements for the degree of **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**. The original research work was carried out by him under by supervision and is hereby accepted.

-------------------------------------- (Signature and Date)

Dr. Akindele A. Onifade

B.Tech (Ogbomoso), MSc,Ph.D (Ibadan)

Supervisor

-------------------------------------- (Signature and Date)

Matthew O. Adewole, PhD

Coordinator, Department of Computer Science and Mathematics

Accepted as partial fulfillment of the requirement for the degree of BACHELOR of SCIENCE (Computer Science)

``

DEDICATION

This project is dedicated to the glory of Almighty God and the Holy Spirit, for being faithful and merciful, for seeing me through to the end of this project. I also dedicate this work to my father, Pst. Oyeranmi Oluseye Paul, my mother, Mrs. Oyeranmi Olayinka for being a major

Source of support in every way. I finally dedicate this work to Alewi Olamilekan Peter who has been supportive, encouraging, and present during the whole process of delivering this Work.

``

ACKNOWLEDGMENTS

Except the Lord builds the house, they labor in vain that build it. "Indeed, without the involvement of the Almighty God-Omnipotent, Omniscient, Omnipresent, this work would not have been a success." Great is your faithfulness, O Lord! I sincerely thank my parents who brought me to this school and to this world. I also want to thank them for their unconditional love and care for me, but physically, spiritually, and financially, without them I would not be able to start this project. And I also want to thank my best friend, Alewi Olamilekan Peter, who was with me throughout the journey. He helped in all aspects of my life. Mr. J.A Balogun to Dr. Funmilayo Kasali and a host of other lecturers. I would like to thank my supervisor, Dr. Akindele A. Onifade, for his counsel, advice, instructions, free access and all the encouragement given to make this work a success. May the Lord guide, support, keep, and promote him and his family.

``

ABSTRACT

Cyber-attacks are growing more complex, posing greater challenges in detecting breaches effectively. Failure to prevent breaches could jeopardize security services' credibility, including data confidentiality, integrity, and availability, and academic institutions' reputations have not been left behind in this. To combat computer security threats, a variety of intrusion detection methods have been proposed in the literature. The interest of this work is to adopt an Intrusion Detection System (IDS) for academic institutions to provide early detection and prevent network intrusion. The idea is to provide an integrated system that will minimize the weaknesses of the different intrusion prevention techniques while putting to the most effective use the strength of each of them. It also talks about future research challenges to stop these kinds of attacks and make computer systems safer by showing how attackers try to avoid being caught.

Malware, intrusion detection systems, and anomaly detection are some of the terms used in this paper.

Keywords: *Malware, Intrusion detection system, Anomaly detection, Snort, Pfsense*

``

TABLE OF CONTENTS

``

``

LIST OF TABLES

Table Page

``

LIST OF FIGURES

Figure Page

``

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

The evolution of malicious software (malware) poses significant challenges to intrusion detection system developers (IDS). Malicious attacks have become more advanced, and the most difficult task is detecting unknown and obfuscated malware since malware creators employ numerous evasion techniques for data concealment to evade detection by an IDS. Furthermore, security threats such as zero-day attacks meant to target internet users have increased. Because of this, computer security has become more important as information technology has become a bigger part of our daily lives.

As a result, these zero-day attacks have had quite a significant impact in countries such as Australia and the United States. According to the 2017 Symantec Internet Security Threat Report, more than three billion zero-day assaults were recorded in 2016, and the volume and severity of these attacks increased dramatically. The number of zero-day assaults was significantly higher than in the past. Previously Symantec (2017) as mentioned in the data, in 2017, there were around nine billion data breaches. Since 2013, hackers have lost or stolen records (Breach_LeveL_Index, 2017). According to a Symantec study, the number of security breach incidents is on the rise. In the past, cybercriminals primarily focused on stealing bank accounts or stealing bank customers' debit and credit cards (Symantec, 2017). In the current generation, on the other hand, malware has become more ambitious and is now tarnished. Obtaining access to banks and, in some cases, attempting to rob them of tens of millions of dollars in a single attack (Symantec, 2017). Because of this, finding zero-day cyberattacks has become one of the most important things to do.

``

The ease with which cyber threats can spread internationally has been highlighted by high-profile criminal incidents, as a simple compromise can disrupt a business' essential services or facilities. Around the world, many cybercriminals are motivated to steal information, get illegal cash, and identify potential targets. Malware is software that is designed to infiltrate computer systems and abuse vulnerabilities in intrusion detection systems. The Australian Cyber Security Centre (ACSC) performed a comprehensive analysis of the attackers' diverse levels of skill in 2017. (Australian, 2017) As a result, an effective IDS is required to detect novel, powerful malware.

Even though these measures provide some level of security, it has been determined that they are deficient in several ways.

1. A firewall is a hardware or software solution used to enforce security policies on a private network. It is primarily used to regulate traffic entering or leaving a private network. These are just a list of permits and denial rules, and therefore, they may not always can detect intrusions. Firewalls, user authentication, encryption keys, and Virtual Private Networks (VPN) offer some level of security, but cannot protect against malicious programs, inside attacks, or unsecured modems (Ajith, G, & C, 2001). They, therefore, would only be effective as one of the available lines of defense. Perfectly secure systems are difficult to come by for institutions that already have intrusion prevention systems. In addition to possible administrator configuration problems, there are always a few system faults. As a result, intrusion detection systems can be utilized to enhance current systems.
2. Intrusion protection systems are frequently installed on pricey routers. These are sometimes not flexible enough to allow changes, such as in network design

and topology. Intrusion detection systems give you the flexibility you need to keep your home or business safe without having to buy specialized hardware.

3. Cryptography hides information from people who shouldn't see it, but it's hard to tell if someone has attacked. In general, key management is a difficult task. Crypto systems may require special key management systems such as the use of a Terminal Access Controller Access System (TACACS) or Remote Authentication Dial in User Service (RADIUS) server. This could mean specialized hardware or configuration. Otherwise, hackers could gain access to these keys and break into the system.
4. Physical security for the network site or servers. But these aren't perfect because attackers who use telnet sessions to get into a network might not be able to be stopped by physical security.
5. Authentication A method for verifying users of a network resource. The effectiveness of this is hampered by the fact that many people continue to "use easy-to-crack passwords,"

And that some users are either untrustworthy or negligent. Oftentimes, unauthorized users can easily obtain the passwords of users.

Many organizations have also employed anti-virus software. However, this may not provide information as to whether there has been an intrusion or not. Anti-viruses also require frequent updates.

Denning also outlined four primary reasons for IDSs. He saw that most systems have security flows that are easy to break into, that most systems are hard to replace, that making completely secure systems is hard, and that users on the inside can abuse the systems that are already out there (Denning, 1967) This emphasizes the importance of intrusion detection systems.

``

1.2 Sources of intrusion data

The approaches used to identify intrusions were classified in the previous two sections. The input data sources utilized to detect abnormal activity can also be used to classify IDS. There are two sorts of IDS technology in terms of data sources: host-based IDS (HIDS) and network-based IDS (NIDS). HIDS examines data from the host system and audit sources such as the operating system, window server logs, firewall logs, application system audits, and database logs. Insider threats that do not require network traffic can be identified by HIDS (Creech & J, 2014) NIDS is a network traffic monitoring system that monitors network traffic retrieved from a network using packet capture, NetFlow, and other network data sources. A network-based IDS can keep track of many machines connected to a network. External harmful activities that could be triggered by an external threat can be monitored by NIDS at an early stage before the dangers spread to another computer system. Due to the volume of data traveling through modern high-speed communication networks, NIDSs have a limited ability to evaluate all data in a high bandwidth network (Bhuyan & Bhattacharyya DK, 2014). Together with HIDS and firewalls, NIDS is installed at multiple points within an internal attack (Sundaram, 1991).

1.3 History

The IDS journey began thirty years ago, when the need for user access and user monitoring arose because of an increase in enterprise network access. Levels of access to these systems and clear visibility into user activity were necessary to operate safely and securely as day-to-day operations became more dependent on the shared use of information systems. The United States Air Force made much of the initial progress on IDS. James P. Anderson, a pioneer in information security and a member of the Defense Science Board Task Force on Computer Security at the U.S. Air Force, published "Computer Security Threat Monitoring and Surveillance" in 1980. This report is generally credited with introducing automated IDS. The first model, which was developed not long after this report was made public, was based on rule-based systems that continuously scanned and compared network traffic against a list of known threats. These systems were born out of the same methods that are used by anti-virus applications. As the number of shared networks grew in the late 1980s, enterprise system administrators around the globe began adopting intrusion detection systems. However, IDS presented a few challenges. First, it could only flag known issues that had been identified as threats on a signature list; zero-day attacks could compromise the security of a network. The constant scanning and updating of a signature list were laborious and a significant drain on resources. The proliferation and sophistication of network attacks necessitated the development of more effective intrusion detection systems, which occurred primarily in the 1990s. This new method, nicknamed "anomaly detection," identifies unusual network behavior patterns and generates alerts for any recognized anomalies.

Due to the high number of false positives caused by the inconsistency of networks in the 1990s and early 2000s, many administrators began to believe that IDS was unreliable and doomed to a gradual decline. However, the advent of cloud computing has given IDS systems new relevance, resulting in a surge in the IDS market. IDS systems, an essential component of current security best practices, are designed to detect attacks that may occur despite prevention. In fact, IDS is now one of the most popular security technologies, and its growth is anticipated to continue. After all, the monitoring of security, and especially cloud security, manually is an impossible task given how complex the problems are. IDS's logic and strategies are more pertinent than ever before. IDS has found an environment where it can thrive and be most effective with cloud computing. Cloud computing has allowed the infrastructure to catch up with IDS technology. The standardization of cloud server infrastructure is best suited for IDS technology.

As a result, IDS is able to establish stronger and more accurate baselines than was previously possible on erratic on-premises network infrastructures. Today's growth and significance of intrusion detection are significantly influenced by big data. With cloud-hosted databases growing at an exponential rate and the world's data doubling every 20 months, it is not surprising that IDS is more important than ever.

1.4 Statement of the problem

Institutional resources become exposed to unwanted access as computer systems go online. On one hand, hacker tools have gotten better faster than the technical knowledge needed to stop them (Sundaram, 1991). This necessitates a technique for countering these dangers. Academic institutions are also coming online because of the benefits, but many do not prioritize network security in their budgets. As a result, they remain vulnerable to these dangers.

Furthermore, their networks are shared among students and professors, and vital information such as student academic records and financial reports is available. This needs much more stringent surveillance. If left uncontrolled, these networks may suffer significant financial losses due to intrusion and inefficient resource consumption, such as bandwidth waste due to unwanted server requests from network nodes. Using just one intrusion protection technique, such as a firewall, may not always be sufficient. All of this shows how important it is to have a system that can find intrusions in a way that can be controlled.

``

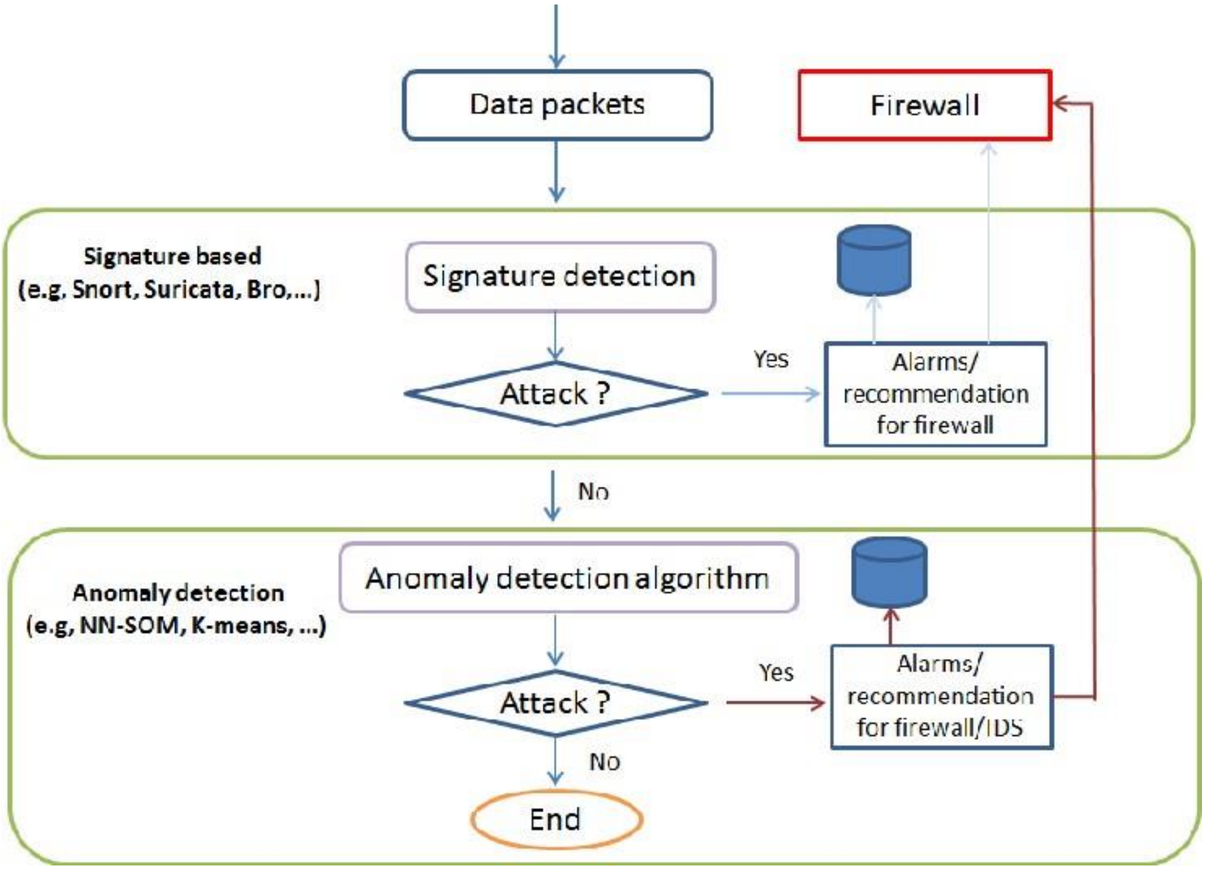

Data packets
Firewall
Signature based
(e.g. Snort, Suricata, Bro,...)
Signature detection
Attack ?
Yes
Alarms/ recommendation for firewall
No
Anomaly detection
(e.g, NN-SOM, K-means, ...)
Anomaly detection algorithm
Attack ?
Yes
Alarms/ recommendation for firewall/IDS
No
End

Fig 1. 1 IDS-process-flow-diagram

``

1.5 Aims and Objectives

The goal of the project is to adapt an intrusion detection system for academic institutions, highlighting the importance of these systems in network security. It emphasizes the need for diverse intrusion prevention measures to be integrated to strengthen the intrusion detection system.

This study looks at the limited resources that academic institutions have when it comes to network security. The research's specific objectives have been to investigate intrusion detection systems for networks (IDS), create a reliable intrusion detection system, implement a comprehensive and efficient intrusion detection system, and validate the intrusion detection system.

1.6 Methodology Proposal

Given that the goal of this study is to create a secure network from intrusions and attacks, the literature survey was the ideal tool for accomplishing this goal. The method is a signature-based methodology that compares observed activity to a baseline profile. The baseline profile is the learned normal behaviour of the monitored system, and it is generated throughout the learning period, during which the IDS learns the environment and generates a normal profile for the monitored system. This environment can include networks, users, and systems, among other things.

It is possible to have a fixed or dynamic profile. A fixed profile remains constant throughout time, but a dynamic profile varies when the systems being monitored change. A dynamic profile adds a lot of work to the system because the IDS keeps changing it, which makes it easy to get around.

By spreading the attack over a long period, an attacker can evade the IDS that uses a dynamic profile. As a result, her attack is incorporated into the profile, and the IDS treats her alterations as normal system adjustments. Any deviations that fall outside of a predetermined threshold are reported as violations. Because every deviation from regular behaviour is labelled as an anomaly, a fixed profile is (Agrawal, 2012) particularly successful at identifying new assaults.

Without any system changes, anomaly-based approaches can detect zero-day assaults in the environment. The statistical anomaly detection, knowledge/data-mining, and anomalous intrusion detection methodologies use three general strategies for detecting anomalies. For the systems to be effective, the threshold must be modified according to the requirements and behavior of the monitored environment.

1.7 Significance of study

IDS gives organizations greater visibility across their networks, making it easier to meet security regulations. Additionally, network administrators can use their IDS logs as part of the documentation to show they are meeting certain compliance requirements. The implementation of an intrusion detection system at mountain top university can also improve security responses.

1.8 Scope and limitation

The limitations of the system are as follows:

I. The system is only limited to the administrator.

II. The system can only be connected through LAN.

III. The system requires constant internet

IV. The administrator can only see the monitoring of the flow.

``

1.9 Definition of Terms

These refers to the terms that will be encountered in this report of the Intrusion detection system.

a. **DST:** also known as destination. It is where the network is heading.

b. **SID:** A security identifier, or SID for short, is a value that is completely unique and can be used to identify any security entity that the Windows operating system (OS) is capable of authenticating. SRC: stands for position of source, this is where the networking is coming from.

c. **PRI:** To carry multiple DS0 voice and data transmissions between a network and a user, a telecommunications interface standard known as PRI (Primary Rate Interface) is utilized on an Integrated Services Digital Network (ISDN). The Primary Rate Interface (PRI) is the industry standard for the delivery of telecommunications services to businesses and offices.

d. **TCP/UDP:** Transmission Control Protocol and User Datagram Protocol are both abbreviations for the Transmission Control Protocol, which is a communications standard that enables application programs and computing devices to exchange messages while connected to a network. Its primary functions are to transmit packets across the internet and to ensure that data and messages are successfully delivered over network connections. And UDP is a communications protocol that is utilized primarily for the purpose of establishing connections between applications on the internet that are both low-latency and loss-tolerant. UDP allows data to be sent before the receiving party gives its agreement, which significantly speeds up the transmission process.

``

e. **WAN/LAN:** A wide-area network, also known as a WAN or LAN, is a type of computer network that links together multiple local area networks. Localized networks are able to communicate with one another despite the great distances involved because wide area networks (WANs) are not associated with a particular location.

f. **HTTP/HTTPS:** The Hypertext Transfer Protocol, abbreviated as HTTP and HTTPS, is a rule set that governs the transfer of files such as text, images, sound, video, and other types of multimedia files over the internet. In addition to HTTP, which stands for the Hypertext Transfer Protocol (HTTP). It allows for safe communication to take place over a computer network and has found widespread application on the Internet.

g. **GPLv2:** is an abbreviation for "General Public License Version 2.0," which is also referred to as "GPL v2." The GNU Public License version 2 (GPL 2) was first made available to the public in 1991. As a copyleft license, it stipulates that users must abide by several stringent rules and requirements.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The chapter gives an outline of computer attacks as well as some of the tactics used to combat them. The architectures, models, and implementations of intrusion detection systems are also explored. An intrusion occurs when a network's security policy is successfully violated (Zhou, Carlson, & Bishop, 2005). An Intrusion Detection System (IDS) is a temporary security solution for protecting computer systems that have been compromised. If an attacker tries to bypass a security feature such as authentication or a firewall, the system sends a notification or acts (Lee & Stolfo, 1998). (Sodiya & Akinwale, 2004) define intrusion Detection Systems (IDS) as systems that can detect both internal and external intrusions on a computer system and undertake some measures to eliminate them. (Kendall, 1999) Also identifies failed intrusion attempts, allowing for preventive steps and, in some cases, counter-intrusions. Intrusion detection systems look for signs that an intrusion or attempted intrusion has occurred or is about to occur.

Data obtained from system probes, such as file system alteration monitors, or audit trails established by the operating system, network traffic flowing between systems, application logs, or data collected from system probes, could also be used to detect infiltration. (Kim, Lee, & Kim, 2014). A new hybrid intrusion detection technique within a decomposed structure: this method combines a misuse detection model and an anomaly detection model in a way that enables them to collaborate with one another. (Syed, Raza, & Biju, 2018) Analyzes the effectiveness of two open-source intrusion detection systems (IDSs), Snort and Suricata, in identifying malicious traffic on computer networks. (Alazab A. , 2014)Presented the Intelligent Intrusion

Detection and Prevention System, which is a hybrid of SIDS and AIDS (IIDPS). The IIDPS is linked to a reaction action in this study utilizing fuzzy logic, which is a novel approach. (Galal, Mahdy, & Atiea, 2016) Proposed a behavior-based characteristics model to define malware instances' destructive behavior. To get the proposed model, run a dynamic analysis on a recent malware dataset in a controlled virtual environment, capturing traces of API calls made by malware entities. In a signature-based network intrusion detection system (Hubballi & Vinoth, 2014) they analyzed existing false alarm minimizing strategies (NIDS). The authors suggest a taxonomy of ways to reduce false alarms in signature-based IDS, along with the pros and cons of each group.

2.2 Overview of Computer Attacks

Any harmful behavior directed at a computer system, or the services supplied by the system is referred to as a "computer attack." Examples of possible attacks include malware, denial of service (DoS), and flaw exploitation, unauthorized use of services, phishing, and even physical attacks against computer hardware. An understanding of the various approaches that have been employed by crackers or even script kiddies will go a long way in helping develop a system that will safeguard institutions against the incursions of crackers. Hackers have deployed a broad array of tactics to break into systems, and these include, but are not limited to:

a. Phishing

Frequent phishing attacks involve sending a substantial percentage of fraudulent emails to unwitting recipients while posing as reliable sources. Often, the fake emails look real, but they contain a web file or code that has been hacked. This gives attackers access to your device so they can control it, collect information, inject malicious scripts or files, or access information like user information, financial data, and more. Phishing attacks can also happen on social media and other online communities when people send each other messages with a secret goal.

b. Zero-day vulnerability

The term "zero-day exploit" describes the practice of taking advantage of a network flaw immediately after it has been discovered but before a patch has been released and/or installed. When a new vulnerability is found, zero-day attackers take advantage of it during a short window when there are no fixes or safety measures. Thus, combating zero-day threats needs ongoing monitoring, proactive detection, and adaptive threat management strategies.

c. Configuration mistakes

Configuration errors may allow an attacker to obtain access to a network. These flaws could allow access without requiring any kind of authentication, such as when users are given guest accounts with no password restrictions. Users may be given administrative privileges inadvertently. A person can also gain the rights of other users by getting their passwords or breaking rules that keep them from getting in (Root attack) (Joseph, 2003).

d. Masquerading

After observing the flow of traffic, the attacker can change a TCP packet or originate one but send it with a falsified source address to make it appear to be from a trusted source. The attacker can then send a Trojan, stop the service from working, or ask for sensitive information.

e. Malware

The term "malware" refers to a wide variety of computer hazards, such as spyware, viruses, and worms. When a user engages in malicious behavior, such as clicking on a malicious link or opening an email attachment, malware can exploit a vulnerability in the network and enter the system. Malware and malicious files on a computer system can deny entry to the network's vital components, retrieve data from the hard drive, and potentially render the system inoperable.

Malware is so pervasive that its methods of operation are diverse. Viruses are the most prevalent type, infecting applications by attaching themselves to the initialization procedure. The virus spreads across the computer system, infecting other programs. There are other ways that viruses can attach themselves to executable code or files. They can do this by making a decoy virus file with the same name but an.exe extension.

f. Trojan horse

It is malicious software concealed within a legitimate program. A Trojan, unlike viruses, does not spread itself. It is usually used to make a backdoor that can be used by hackers. As for worms, unlike viruses, worms are self-contained programs that do not attack the host. They spread through networks and computers, not the host. Worms are frequently distributed using email attachments, which send a copy of themselves to every contact on the infected computer's email list. They're typically used to cause a denial-of service attack by overloading an email server.

2.3 Why IDS?

While most of the above mentioned are prudent steps towards network security, an IDS provides an in-depth view by detecting and logging hostile activities. It can watch over the network whenever other protective measures are bypassed.

We also need an IDS despite having a firewall because firewalls simply shut off all communication and turn on only a small amount of well-chosen traffic. It does not have the capability to detect internal attacks.

``

Firewalls are typically used at the network's perimeter, where they can only monitor and potentially restrict incoming and outgoing traffic. However, a larger proportion of malicious attackers may exist within the network. Protecting the network from internal attacks through a boundary firewall will thus be almost impossible.

2.4 Intrusion Detection System (IDS)

An intrusion detection system (IDS) is a defensive measure component that protects computer systems and networks from being attacked. Intrusion is defined as the process of an information system user performing an action unlawfully. Inside or outside, intruders who go over their authorized limits to execute an action are considered intruders. Even if the behaviour does not cause harm, it is cause for concern because it may harm the service provided by the system or the system's health. Intrusion detection entails explaining the harm or gain sought by an invader to a single entity attempting unauthorized access to a system. None of the automated detection approaches now available to us is sufficient to recognize an intruder before they start interacting with the system used. Meanwhile, regular precautions are taken to avert intrusion.

The system administrators are in charge of this. These can include things like it requires that the user give the password ahead of time. Gain access to any system you want, barring all or some access. The familiar connection to the network and physical access.

To gain access, an intruder may exploit weaknesses that an intruder may exploit.

The goal of an IDS is to identify types of malicious network traffic and computer usage that even a traditional firewall cannot identify. This is essential to achieving effective protection against actions that potentially ruin computer systems' availability, integrity, or confidentiality.

``

Signature-based IDS (SIDS) and anomaly-based IDS (AIS) are two broad categories of intrusion detection systems (AIDS). Based on their structure, IDSs can be roughly divided into the following five categories: network-based IDSs (also known as NIDS), host-based IDSs (also known as HIDS), distributed IDSs (also known as DIDS), protocol-based IDSs (also known as PIDS), and application-based IDSs (also known as APIDS) (APIDS).

2.4.1 Network-Based Intrusion Detection System (NIDS)

A whole network segment, or subnet, is monitored by the NIDS. This is accomplished by altering the NIDS network interface card's mode (NIC). Normally, a network interface card (NIC) works in a non-promiscuous mode, listening only to packets destined for its own MAC address. Other packets are ignored rather than being pushed up the stack for analysis. The NIDS must accept all packets and forward those up the stack to monitor all traffic on the subnet, not only those addressed to the NIDS machine. Promiscuous mode is the term for this. The NIDS can listen in on all communications on the network segment in promiscuous mode. That is not, however, all that is required to ensure that your NIDS can listen to all traffic on the network. All packets on the subnet must be sent to your NIDS by the network device directly upstream of your NIDS.

Network intrusion detection systems (NIDSs) monitor network traffic for specific network segments or devices and analyze network and application protocol activity to detect suspicious activity. It can recognize a wide range of occurrences of interest. A NIDS has the advantage of having no impact on the systems or networks it monitors. It doesn't put any additional strain on the hosts, and an attacker who compromises one of the monitored systems can't access the NIDS and may not even be aware of it.

2.4.2 Host-Based Intrusion Detection System (HIDS)

The network card of a system with a HIDS installed generally runs in non-promiscuous mode, as it is primarily responsible for protecting the system on which it resides, not the entire subnet. HIDSs are systems that look for suspicious activity by monitoring the characteristics of a single host and the events that occur within that host. Network traffic (just for that host), system logs, ongoing processes, application activity, file access and modification, and system and application configuration changes are examples of the types of features a host based IDPS might monitor.

Most host based IDSs go after critical hosts, like servers that anyone can access and servers that hold sensitive data.

2.4.3 Distributed Intrusion Detection System, or DIDS

It consists of a network of NIDS, HIDS, or both sensors that are spread throughout your organization and report to a central correlation system. A DIDS system with four sensors and a centralized administration station. The public servers are protected by the NIDS 1 and NIDS 2 sensors, which operate in stealth promiscuous mode. The sensors NIDS 3 and NIDS 4 protect the host systems in the trustworthy computing base. One benefit of using DIDS for analysis is that you can keep an eye overall system, the whole network, or just one host.

2.4.4 Protocol-Based Intrusion Detection (PIDS)

Protocol-based intrusion detection systems (PIDSs) are systems or agents that are permanently installed at the server's front end and are responsible for controlling and translating the protocol used by clients to communicate with the server. It does this by monitoring the HTTPS protocol live feed on a continuous basis and recognizing the HTTP protocol that is associated with it. This is done to secure the web server. For this system to make use of HTTPS, it would need to remain in this interface. This is what happens because HTTPS data isn't encrypted until it gets to the web presentation layer.

2.4.5 Application Protocol-based Intrusion Detection System (APIDS)

A system or agent that, in most cases, resides within a collection of servers is referred to as an Application Protocol-based Intrusion Detection System, or APIDS. It detects attacks by keeping tabs on network traffic and understanding the context of those messages according to predefined protocols for applications. This would, for example, keep an eye on the middleware's SQL protocol as it talks to the web server's database.

2.4.6 Signature intrusion detection systems (SIDS)

(SIDS) use pattern matching techniques to detect known attacks; they're also known as Knowledge-based Detection or Misuse Detection (Khraisat, A, I, & P, 2018). Matching algorithms are employed in SIDS to locate a previous intrusion.

To put it another way, an alarm signal is dispatched whenever the signature of an attack matches the signature of a previous attack that is already stored in the signature database. To detect malicious activity, SIDS analyzes a host's logs for specific patterns of behavior or command execution. In the literature (Modi, et al., 2013), SIDS has also been called Knowledge-Based Detection or Misuse Detection.

``

The fundamental concept is to create an intrusion signature database, compare current activities to existing signatures, and raise an alarm if a match is detected. One possible result of a rule with the form "if: antecedent-then: consequence" is the phrase "if (source IP address=destination IP address), then label as an attack. “For previously known incursions, SIDS usually provides good detection accuracy (Kreibic & Crowcroft, 2004). SIDS, on the other hand, has trouble identifying zero-day attacks since no matching signature exists in the database until the new attack's signature is extracted and stored. SIDS is used in a variety of standard tools, like Snort (Roesch, 1999) and netstat (G & RA, 1999).

2.4.7 An Anomaly Intrusion Detection System

Intrusion detection system based on anomalies (AIDS) Because of its ability to surpass the limitations of SIDS, AIDS has piqued the interest of many academics. A normal model of a computer system's behavior is built-in AIDS by utilizing machine learning, statistical-based, or knowledge-based methods. An anomaly is defined as a significant difference between observed behavior and the model, which can be construed as an incursion.

This set of approaches is based on the notion that harmful conduct varies from normal user behavior. Intrusions are anomalous user behaviors that differ from usual behavior. The training phase and the testing phase are the two stages in the development of AIDS. The usual traffic profile is utilized in the training phase to build a model of normal activity, and a new data set is used in the testing phase to determine the system's ability to generalize to previously unseen intrusions.

AIDS can be classed into several groups based on the training approach employed, such as statistical, knowledge-based, and machine learning-based (I, SD,

& R, 2014). The key advantage of AIDS is the ability to detect zero-day attacks because it does not rely on a signature database to detect aberrant user behavior (Alazab, Hobb, Abawajy, & Alazab, 2012). When the examined behavior varies from the norm,

AIDS sends out a warning signal. AIDS also provides several advantages. To begin with, they can detect internal harmful activity. It sounds an alarm if an intruder starts making transactions in a stolen account that can be seen in normal user activity. Second, a cybercriminal's ability to distinguish between a normal user and a malicious user is quite strong. Because the system is made up of different profiles, it can act in a way that doesn't raise an alarm.

2.5 Comparison of IDS with Firewalls:

IDSs and firewalls both try to keep a network safe, but an IDS is different from a firewall in that it looks outside the network for intrusions to prevent them from happening. Firewalls prevent intrusions by preventing access between networks. However, if an attack originates from within the network itself, the firewall will not detect it. An intrusion detection system looks for it after something suspicious has happened and sounds an alarm

``

FIREWALL	**IDS**
A firewall is a piece of hardware or software that works in a networked environment to prevent unwanted access while allowing permitted traffic.	An intrusion detection system, or IDS, is a piece of software or hardware that watches a network (in the case of a HIDS) or a host (in the case of a NIDS) for attempts to break in and then alerts the right people.
A firewall can prevent unwanted network access (e.g., a watchman standing at a gate can block a thief).	An intrusion detection system (IDS) can only report an intrusion; it cannot stop it (e.g., a CCTV camera can alert about a thief but cannot stop it).
A firewall cannot identify security breaches in traffic that does not travel through it.	IDS can protect the inside of a computer or network by gathering information from a wide range of system and network resources and looking for signs of security problems.
Permitted traffic is not inspected by the firewall.	An IDS monitors the entire network.
To an outsider, the firewall is the most visible aspect of the network. As a result, they are more likely to be assaulted first.	IDS are notoriously difficult to detect in a network (especially in stealth mode).

Table 2. 1 Difference between Firewall and IDS

``

2.6 IDS-related features.

1. Searches are done based on many things, such as the destination and source addresses, the ports, and the time.
2. Packet viewing is used to display distinct elements of a packet. You can also display various payload and header items.
3. Alerts can be handled by creating alert classes, deleting them, exporting them, and forwarding them to an email address.
4. For graphical analysis, protocol, time, port numbers, classes, and IP addresses are used as inputs to make charts.
5. Snapshots of the Alerts database are also taken. You can, for example, see the last 24 hours' worth of notifications, as well as frequent and distinctive alerts.
6. The owner of a certain IP address that is causing an assault on your network might be discovered by utilizing several databases on the Internet. Based on this information, we can communicate with that person to put a stop to it. The databases contain information about the owner's IP addresses and domain names.

2.7 Snort

Snort is a lightweight and widely used network intrusion detection and prevention system (IDS/IPS) for defending against attackers. Open-source software was created in 1998 by Martin Roesch using the C programming language and has undergone continuous revision and improvement for more than a decade. Snort is compatible with practically every computer architecture and operating system. It has now evolved into a global network intrusion detection and prevention system.

``

Furthermore, Snort-IDS can do an in-depth analysis of data flow and protocol, as well as create real-time alerts. It scans and compares the data packets in network traffic with the rules for detecting anomalous data packet traffic. Snort-IDS rules are written in a one-line format. It's simple to read and understand, and it's also adaptable. Packet Decoder, Preprocessor, Detection Engine, Logging, and Alerting are the fundamental components of Snort-IDS.

Snort's flow work is divided into six sections: catching data packages; examining data code; preparing data packages; parsing rules; detecting engines; and logging. Snort works with rules, which are detection signatures.

Snort is configured to log traffic streaming into a private network. The information gathered is then utilized to determine whether an IDS system is necessary on the protected network. The intrusion detection system, Snort, was chosen because:

1. It's an open-source intrusion detection system, for starters. As a result, it's useful in situations where NIDS sensors aren't cost-effective.
2. It's a small and light application. When it comes to resource use, it is cost-effective.
3. Snort may be set up to not only detect infiltration but also prevent it.
4. Snort uses open-source flexible rules as well. Snort's database contains over 2400 attack signatures (Hwang, 2007). The rules can be readily tweaked to match the requirements of the user.
5. Snort is compatible with both Linux and Windows systems. It's also one of the most popular IDSs (Hwang, 2007), and I found a lot of information about it.

There are two types of basic criteria for snort configuration: functional and non-functional needs. The following are the functional requirements:

``

a. To improve the effectiveness of the IDS, the configuration is connected to some intrusion prevention systems. Firewalls, Virtual Private Networks (VPNs), and even subnetted networks are examples of these.

b. Although IDSs come with up to 256MB of random-access memory (RAM), this limited RAM can cause packets to be dropped by the IDS. Sensors now have a total of 1024 MB of RAM when they leave the factory.

c. The network is segregated using a layer two switch. To switch packets, the switch uses MAC addresses. A layer three switch, on the other hand, would be preferable for granule growth.

d. Layer 3 switches (VLANs) could also be integrated with a layer three switch (VLANs). VLANs make networks safer by separating them in a way that lets different groups, like students, administrators, and support staff, use the same infrastructure.

2.8 PfSense

PfSense is a free, open-source FreeBSD customized distribution designed for use as a firewall and router, with everything managed using a simple web interface. The web-based GUI configurator is the name of this online interface. To deploy and use pfSense, no prior knowledge of FreeBSD is required; in fact, the bulk of the user base has never used FreeBSD outside of pfSense. Snort is a collection of additional extension features for pfSense that allows you to set up and customize signature rules. pfSense is a well-known project with hundreds of installations ranging from modest home networks covering a single machine to huge enterprises, universities, and other institutions. Any hardware that is supported by the FreeBSD version in use is compatible with pfSense.cal

``

CHAPTER THREE

METHODOLOGY

3.1. Method of Identification of User and System Requirements

Using a systematic review method for identifying, the user and system requirements of the software system were identified during this project. The techniques used in highlighting the system requirements of the software includes; Nonfunctional requirements that specify criteria that can be used to judge the operation of the system, functional requirements which describe the services the software must offer, Hardware requirements, and the Software requirements.

During the course of the development of the system, the user requirements were identified and they include; basic user requirements. In this chapter, the following models; Use case, Sequence, Activity, class, and System architecture, that were used in the system design would be adequately discussed. This chapter would also talk about the method of system implementation. Finally approaches to testing the system would be highlighted.

3.1.1 Identification of users' requirements

This section discusses the functional and non-functional requirements of the system being developed. It highlights the requirements for system implementation. Also described in this section are the hardware and software requirements for the development of the system.

a. **The non-functional requirements**

The non-functional requirements of the system include:

i. Providing physical security to network devices

ii. Develop best practices for utilizing networked resources. For instance, password selection and use.

``

iii. Establishing a network of educational opportunities for students and employees

b. Functional requirements:

i. Users will be seeing the firewall logs and alerts

ii. Users will be seeing the snort alerts and blocked host

c. Hardware requirements:

The desktop for the server should be at least 64 bits, have a minimum of 2GB of memory, a minimum of 10GB of storage space, at least a 4-port switch, and standard internet flow from an Ethernet port network, either from a router or from modem.

d. Software requirements:

The software required to implement this system includes Pfsense, a system with a good bios for installation, a snort package, and oinkcode.

3.1.2 Identification of user requirements

This section highlights the system and user requirements of the system.

a. System admin requirements

The network administrator has been assigned the role of administrator for this project.

The admin can scan through the network and check the alert log entries. The admin is also responsible for any information seen on the logs and is responsible for the management of the network and the system.

3.2 System Design Methods

The system design was specified using relevant UML diagrams such as use case diagram, sequence diagram and activity diagram. The system architecture was also designed and described in this section. The UML diagrams used in designing the system represent various functions that are available on the system and how they are used by the users of the system as identified in the previous section.

3.2.1 Diagram of use-cases

The figure 3.1, described below, presents the admin of this system and the various actions that they can perform on the system. It also talks about their different roles and conditions for the admin to perform various activities and for those activities to be termed as successful.

3.2.2 Activity flowchart

Figure 3.2 below describes the flow of activity in the system. Once the user requests to log in, if the request is successful, the user logs in to the server. If the login was unsuccessful, the user is prompted to try again. If a user logs in as a network administrator, then the user can perform the following activities: change configurations, check the traffic graph, and view the firewall logs and alert log entries.

3.2.3 Classification diagram

Figure 3.3 below describes the class diagram of the system. The class diagram shows the relationship between the system entities. The attributes of each of these entities represent columns of the tables that they model in the database. The user class, which comprises properties that define the user of the system (Network administrator), also contains methods that are used for performing different operations and objects of this class.

3.2.4 System structure

The system architecture in Figure 3.4 below depicts that the system presented is navigated to the IDS server. The server talks to the routes that have been set up to serve the right page and do what needs to be done.

``

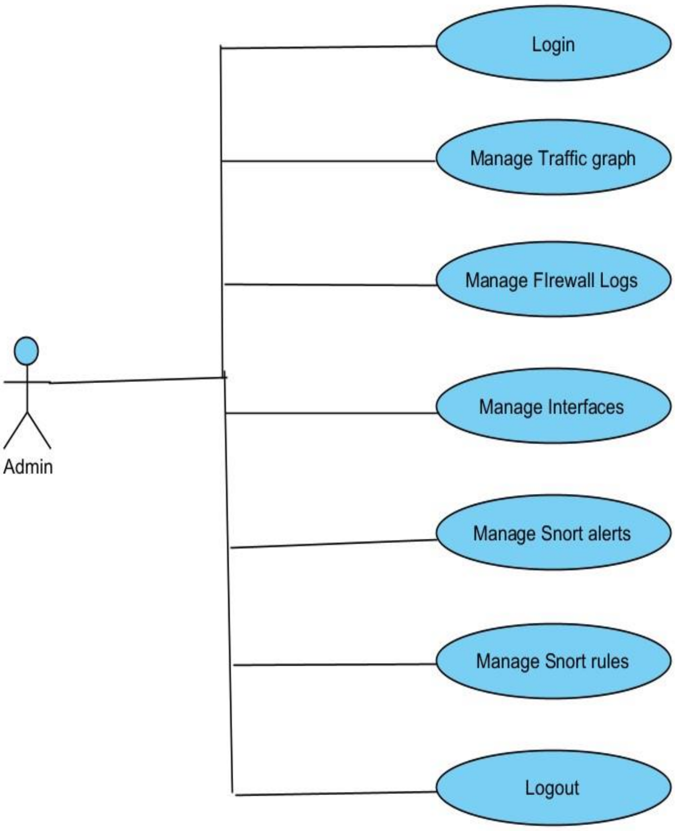


Figure 3. 1 System use case diagram

``

Use case Name	Check alerts
Actors	Admin
Flow of Events	Admin navigates to the snort Admin selects a alert and block page Admin selects scrolls through the alerts
Entry Condition	Admin must be logged in to the server
Exit condition	The admin checks for rules and alerts.
Quality Requirements	The admin must successfully examine the alerts and block the hot who are tagged dangerous or suspicious.

Table 3. 1 Assign role use case

``

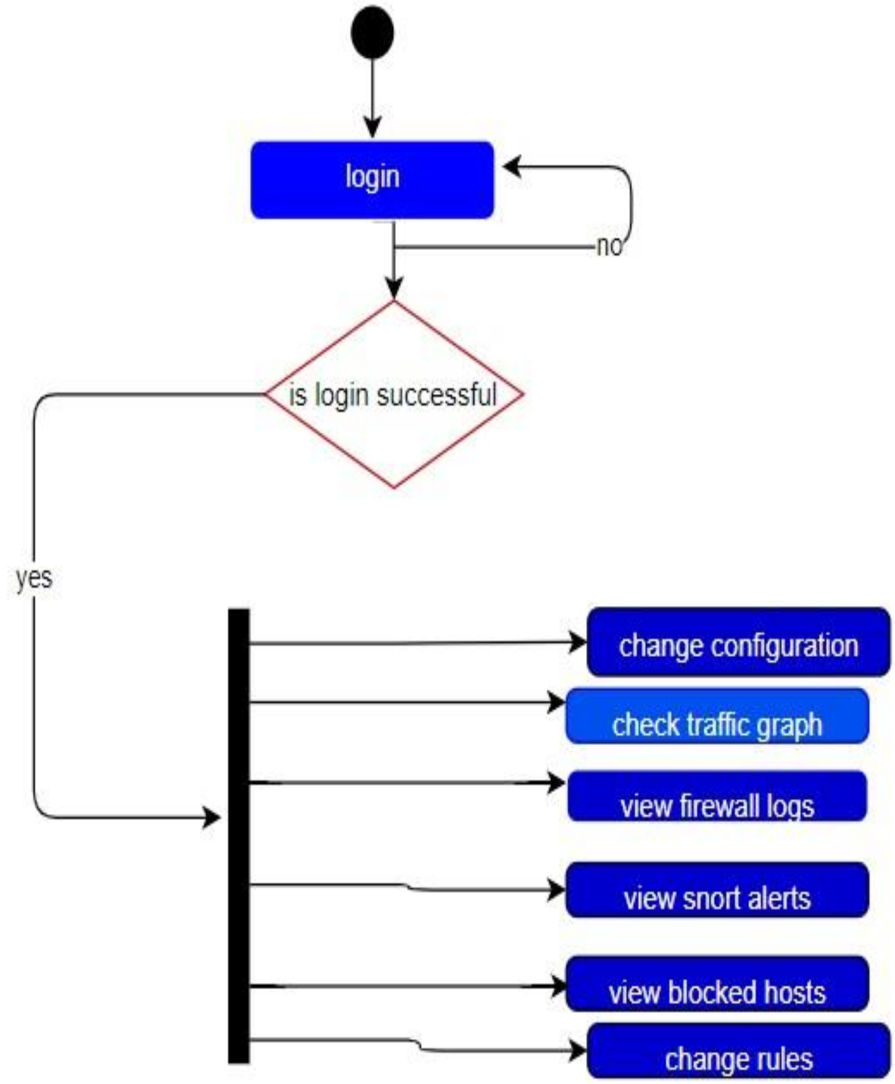


Figure 3. 2 Activity flowchart of the system

``

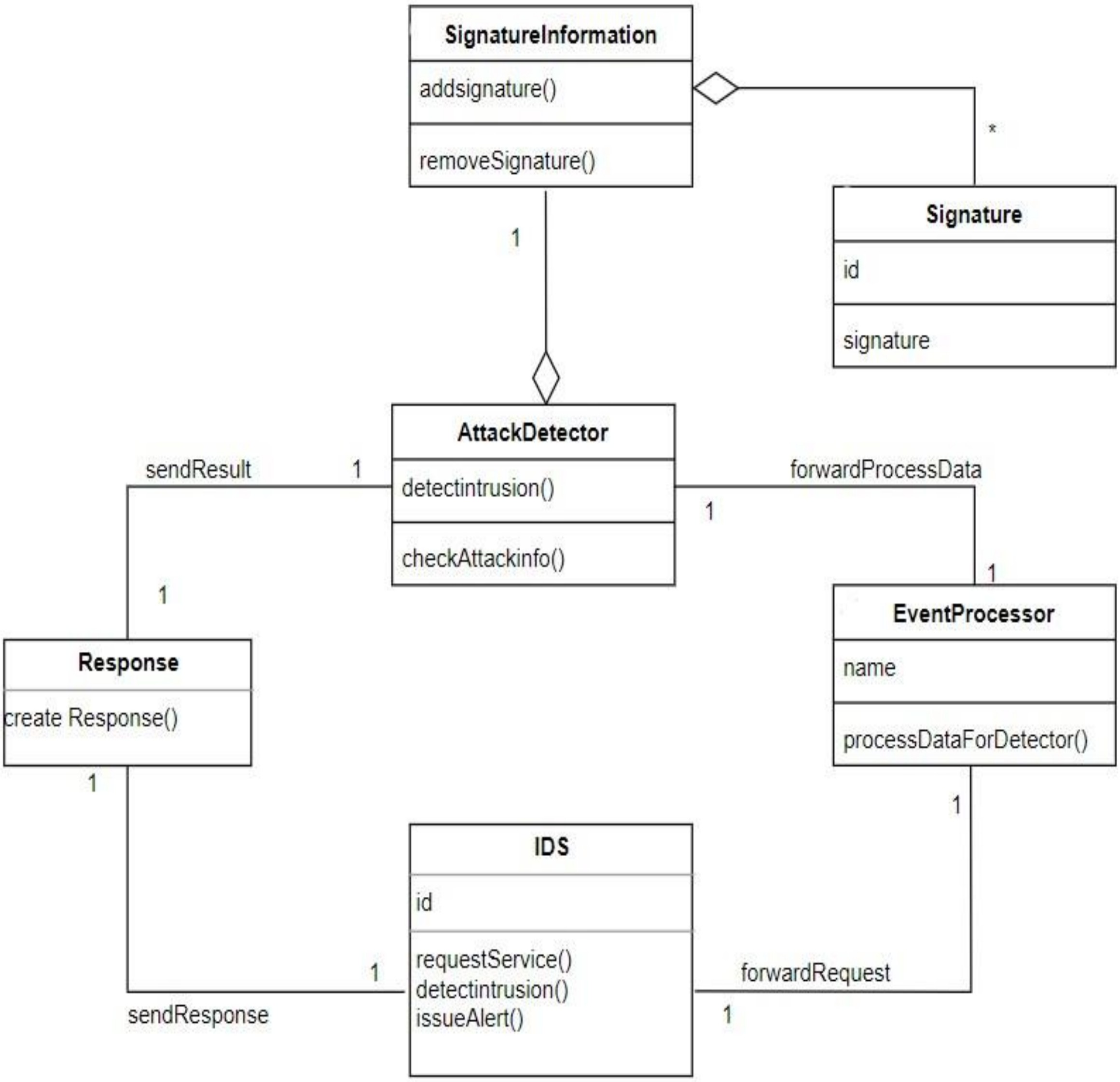


Figure 3. 3 Classification diagram of the system

``

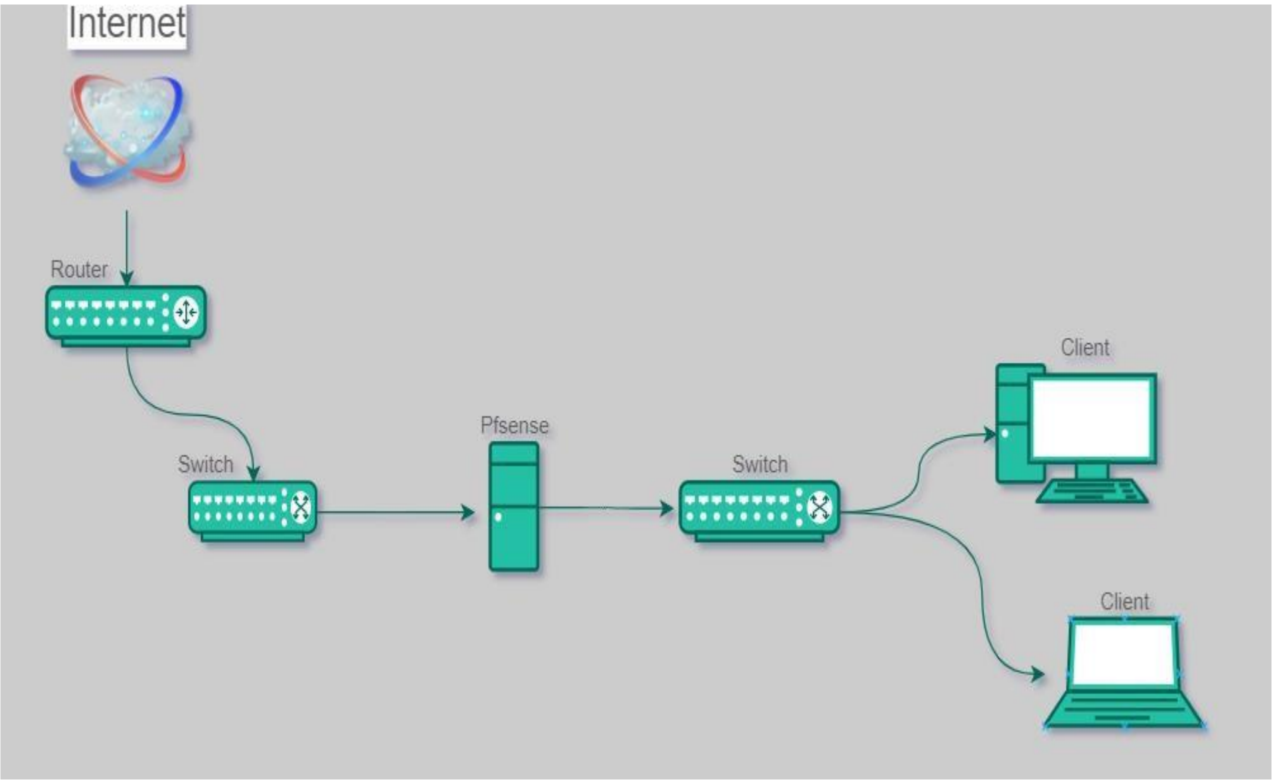


Figure 3. 4 System structure of the system

``

3.3 System Implementation

In practice and experimentation, I use Snort packages as an extension feature of the pfSense system, which runs on Bare Metal. The code name of this project is "IsaacIDSsever". The structure of the network.

The configuration and setup steps are as follows:

- Install pfSense
- Configure the WAN interface and enable the LAN interfaces.
- Enable the routing of internet traffic through a private IP (192.168.168.2)
- Install the Snort packages and their dependencies.
- Create a license agreement.
- Get Oinkcode
- Install and update the rules.
- Create policy rulesets.
- Monitoring alerts, reports, and activity blocks

Furthermore, pfSense does not have its own IDS feature, but it does have a package system using which other software packages can be integrated with pfSense. It uses the Snort package for using IDS services.

Snort provides IDS/IPS services. It is used for blocking and creating log information about ongoing network activity. To install snort on pfSense, the user can locate it under System/Packag Manager/Available packages. Search for snort and hit the Install button. It will be installed after confirmation. After installation, users can manage snort services under Services/Snort. Snort VRT Rules require a paid subscription, but users can also register for a free trial. The GPLv2 Community Rules and Emerging Threat Open Rules are available for free.

Users must configure the package to use Snort Services.

``

A. Global Settings

Global Settings let users choose IDS package rules. Navigate to Services/Snort/Global Settings. The user must enable one IDS rule and set an update interval. An update interval is a timer that is used for checking whether the package is up to date or not.

B. Snort Interfaces

Navigate to Services/Snort/Snort Interfaces. Click the Add button to add an interface for implementing the Snort service. This will lead to a new setting tab where the user can choose an interface. Once the interface has been added, users can set policies under Services/Snort/Snort Interfaces/Categories/Categories. VRT offers three preconfigured IPS policies (Connectivity, Balanced, and Security) that facilitate user implementation. Interface Rules can be managed under Services/Snort/Snort Interfaces.

C. Updates

The Update tab is useful for managing updates. It contains information about the rules, such as the MD5 signature hash and the MD5 signature date. It shows the last updated details. The user can update the rule or force an update to download and enable the rule package. A user can also view or clear the rule log. The path to the Update Tab is Services/Snort/Updates.

D. Alerts

The Alert Tab contains Alert Log View Settings, Alert Log View Filter, and Last Alert Log Entries. Alerts provide notification services for any faulty network events. A user can limit the number of log entries and can download or view the alert log at any time. The path to the Alerts Tab is Services/Snort/Alerts.

``

E. Blocked

The Blocked Tab is used for blocking a logical address that the admin does not want to allow network access. The path to the Blocked Tab is Services/Snort/Blocked.

F. The Pass List

The Pass List functions similarly to the whitelist. It contains a list of IP addresses that should not be blocked by the IDS. Navigate to Services/Snort/Pass Lists. A user can create a pass list by clicking on the Add button. The General Information (Name, Description), Auto-Generated IP Addresses, and Custom IP Addresses can be found on the Pass List Edit Tab.

3.3.1 The Network Setup

Different intrusion detection systems can be deployed to protect different sections of the network. The WAN, which is the private network, will be connected to the server, which is the IDS. Then it will be connected to the second server that controls the bandwidth (pfense) and then to the server that distributes the bandwidth (radius manager).

``

CHAPTER FOUR

IMPLEMENTATION AND RESULT

This chapter presents the results of the intrusion detection system and a description of the results obtained. This section covers the result of the system, implemented with Pfsense and Snort.

4.1 The result of the successful installation of pfsense

The result of installing pfsense server on the system. The pfsense page is access by using the default LAN IP address of 192.168.1.1, which will come on the pfsense server. The computer connected to the server via a LAN cable needs to change the IP address to 192.168.1.3 to be on the same network, and if you change the IP address on the server, you must put it in the corresponding network and subnet mask. Then the page is access by writing the address into a browser. Figure 4.1: The server page and admin login page pop up when you tap on advanced.

Fig. 4.2: The admin login page is linked to the home page, which contains system information such as name of the system; user; BIOS; version of the pfsense software and when it was installed; CPU type; hardware crypto; MDS Mitigation; Uptime; and more. The most recent configuration change, the size of the state table, MBUF usage, load average, CPU usage, Memory Usage, SWAP Usage, Interfaces, Firewall, Logs, Traffic Graphs, and the current date and time are displayed.

4.2 Result of the Configuration of the Network Interfaces and Firewall

Figure 4.3 and 4.4 shows the result of the configuration page of the WAN and LAN interfaces. This page consists of the available options to pick from to get the WAN interface to receive internet from the source. The WAN is DHCP, which allows

``

the system to give an IP address to the devices connected to it by LAN. The same thing applies to the LAN interface, but the LAN interface will get a static IP address (192.168.168.2 with a subnet mask of 255.255.255.0), and it is important to disable bogus and do port forwarding using NAT (network address translation).

Figure 4.5 and 4.6 shows the result of the configuration page of the firewall rules for wan and LAN. This allows the admin to pick the connection from which he wants to allow access to the network and can block traffic from going from the LAN or wan to a particular IP address. In addition, the admin can pick the protocol that is most convenient, i.e., TCP, UDP, or any of the protocols. The same process will be done for WAN and LAN. You can also block some ports from receiving traffic.

Figure 4.7 depicts the outcome of the firewall log. The firewall logs shown on the home page at the side of the system information tab. The table shows the time in real time, and the interface is either LAN or WAN. In addition, it shows the source IP, which is the IP where the traffic is coming from, and the destination IP, which is the IP where the traffic is going too

.

4.3 Result of the successful installation of Snort

Figure 4.8 and 4.9 shows the result of a snort in pfsense on the server. The package can be gotten from the system icon above. Tap on the advanced package manager and move to the package manager. There you will see the package manager and available packages. Search for Snort and install it. The snort will take a few minutes to install, after which it will ask for an oinkcode, which can be gotten from the snort website after you sign up and have an account. Snort is accessible via the service icon at the top of the tab.

``

4.4 Results of the configuration of the network interfaces

Figure 4.10 and 4.11 shows the result of the snort interfaces and global settings. The interfaces are configured on a page, which is where the rules will be picked. It is also where most of the rules depend on the interface. There are a series of rules and configurations that can be picked depending on how the admin wants the server to be. The second page after the interface is Global Settings. These are settings like OpenID detectors, rules update intervals, general settings, blocked hosts, and the place where you input your oinkcode.

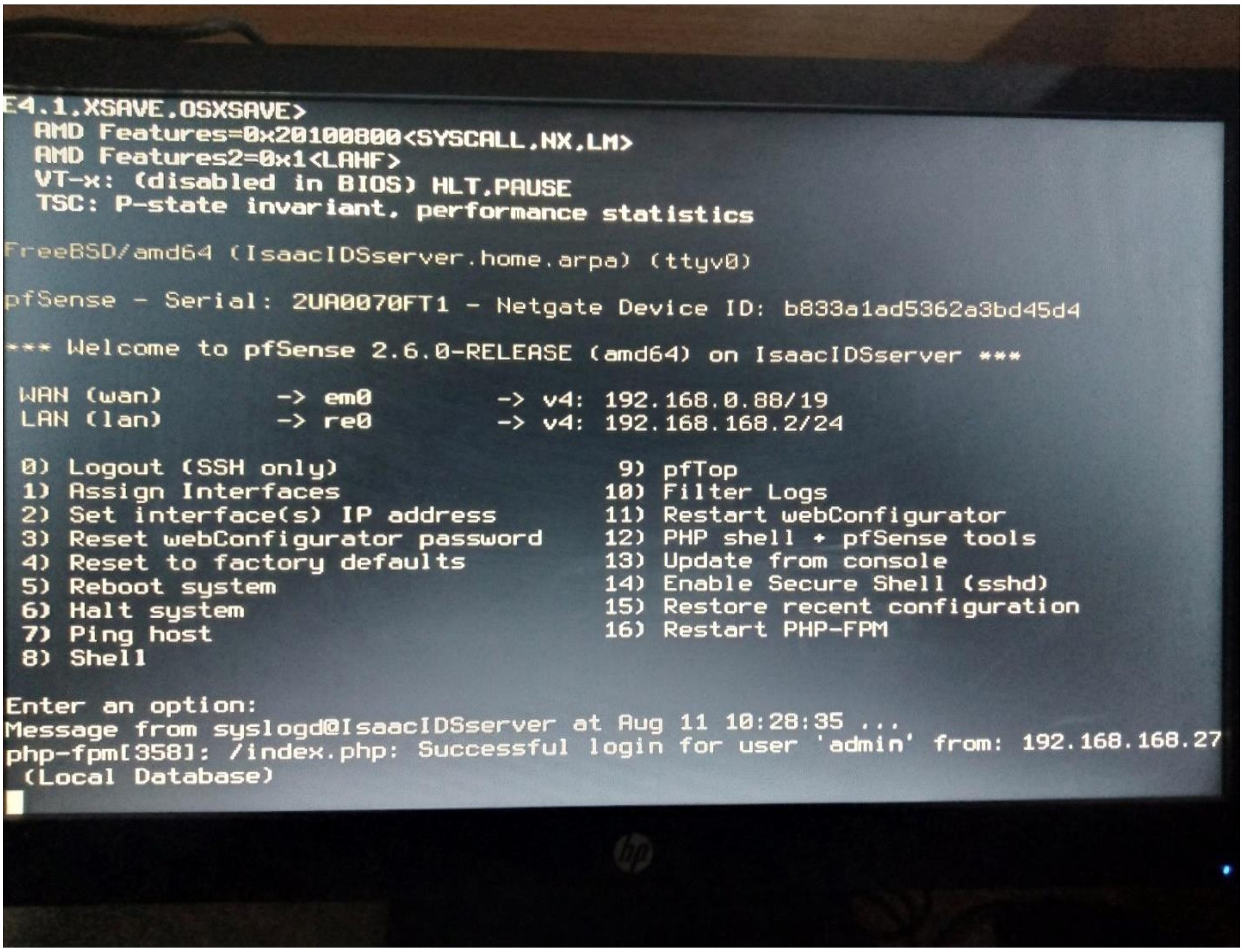
Figure 4.12 shows the result of the update rules and interface rules. The update rules page is the page where the rules enabled in the global settings are downloaded. This is an important part of the IDS. After the rules have been updated, the rules will come, and you can also add more rules on the interfaces depending on the options you want to do.

Figure 4.13 shows the result of the WAN AND LAN interface activation page, and this is where the rules are being pushed on to the network to start scanning and taking logs of the activities on the network. Also, the rules add to the memory usage.

4.5 Results of the IP addresses and host alerted and block by snort

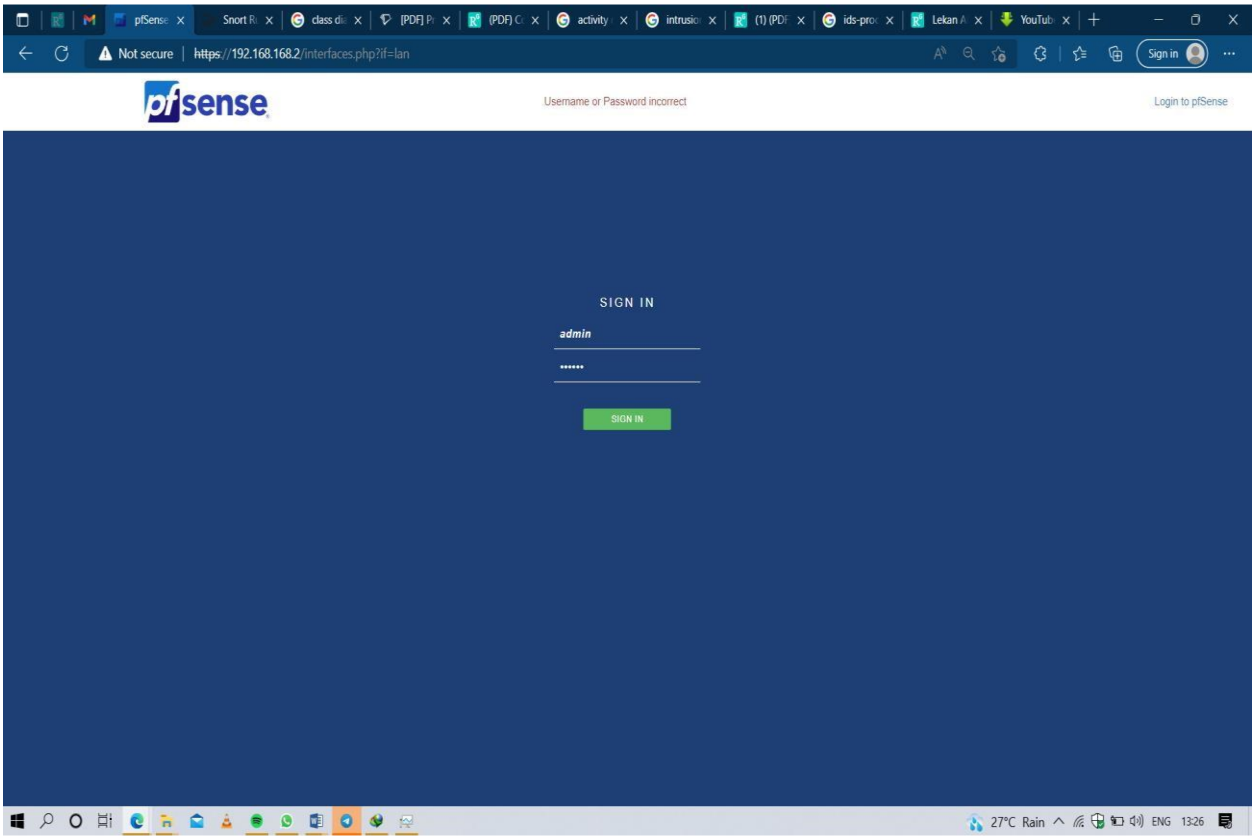
Figure 4.14 shows the result of the alerts and the blocked host. It displays the list of activities shown on each scanned device on the network. As it scans through the network, it keeps a list of alert log entries which consist of the date, pri (primary rate interfaces), protocol, class, dport, SID (security identify), destination ip, and description. Also, you can change the alert settings from both to DST (destination) or SRC (position of source). This will show the log view of the blocked hosts.

``



```
E4.1.XSAVE.OSXSAVE>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
  VT-x: (disabled in BIOS) HLT,PAUSE
  TSC: P-state invariant, performance statistics

FreeBSD/amd64 (IsaacIDSserver.home.arpa) (ttyv0)

pfSense - Serial: 2UA0070FT1 - Netgate Device ID: b833a1ad5362a3bd45d4

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on IsaacIDSserver ***

 WAN (wan)        -> em0        -> v4: 192.168.0.88/19
 LAN (lan)        -> re0        -> v4: 192.168.168.2/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
Message from syslogd@IsaacIDSserver at Aug 11 10:28:35 ...
php-fpm[358]: /index.php: Successful login for user 'admin' from: 192.168.168.27
 (Local Database)
```

``



Not secure | https://192.168.168.2/interfaces.php?if=lan

Username or Password incorrect

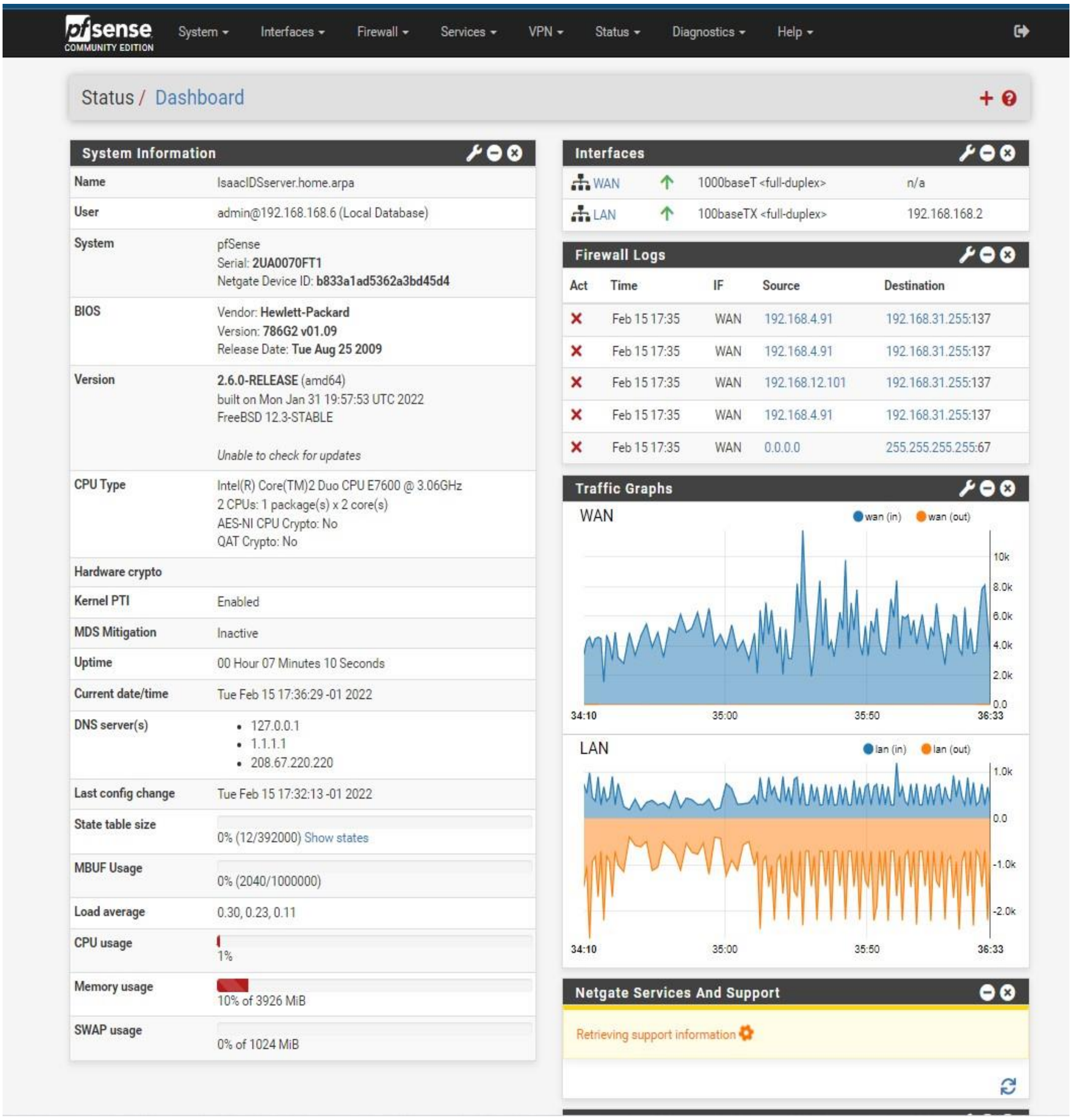
SIGN IN

admin

••••••

Figure 4. 1 (Server page and Login Page)

``



Status / Dashboard

System Information

Name	IsaacIDSserver.home.arpa
User	admin@192.168.168.6 (Local Database)
System	pfSense Serial: **2UA0070FT1** Netgate Device ID: **b833a1ad5362a3bd45d4**
BIOS	Vendor: **Hewlett-Packard** Version: **786G2 v01.09** Release Date: **Tue Aug 25 2009**
Version	**2.6.0-RELEASE** (amd64) built on Mon Jan 31 19:57:53 UTC 2022 FreeBSD 12.3-STABLE *Unable to check for updates*
CPU Type	Intel(R) Core(TM)2 Duo CPU E7600 @ 3.06GHz 2 CPUs: 1 package(s) x 2 core(s) AES-NI CPU Crypto: No QAT Crypto: No
Hardware crypto	
Kernel PTI	Enabled
MDS Mitigation	Inactive
Uptime	00 Hour 07 Minutes 10 Seconds
Current date/time	Tue Feb 15 17:36:29 -01 2022
DNS server(s)	• 127.0.0.1 • 1.1.1.1 • 208.67.220.220
Last config change	Tue Feb 15 17:32:13 -01 2022
State table size	0% (12/392000) Show states
MBUF Usage	0% (2040/1000000)
Load average	0.30, 0.23, 0.11
CPU usage	1%
Memory usage	10% of 3926 MiB
SWAP usage	0% of 1024 MiB

Interfaces

WAN	↑	1000baseT <full-duplex>	n/a
LAN	↑	100baseTX <full-duplex>	192.168.168.2

Firewall Logs

Act	Time	IF	Source	Destination
✗	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✗	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✗	Feb 15 17:35	WAN	192.168.12.101	192.168.31.255:137
✗	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✗	Feb 15 17:35	WAN	0.0.0.0	255.255.255.255:67

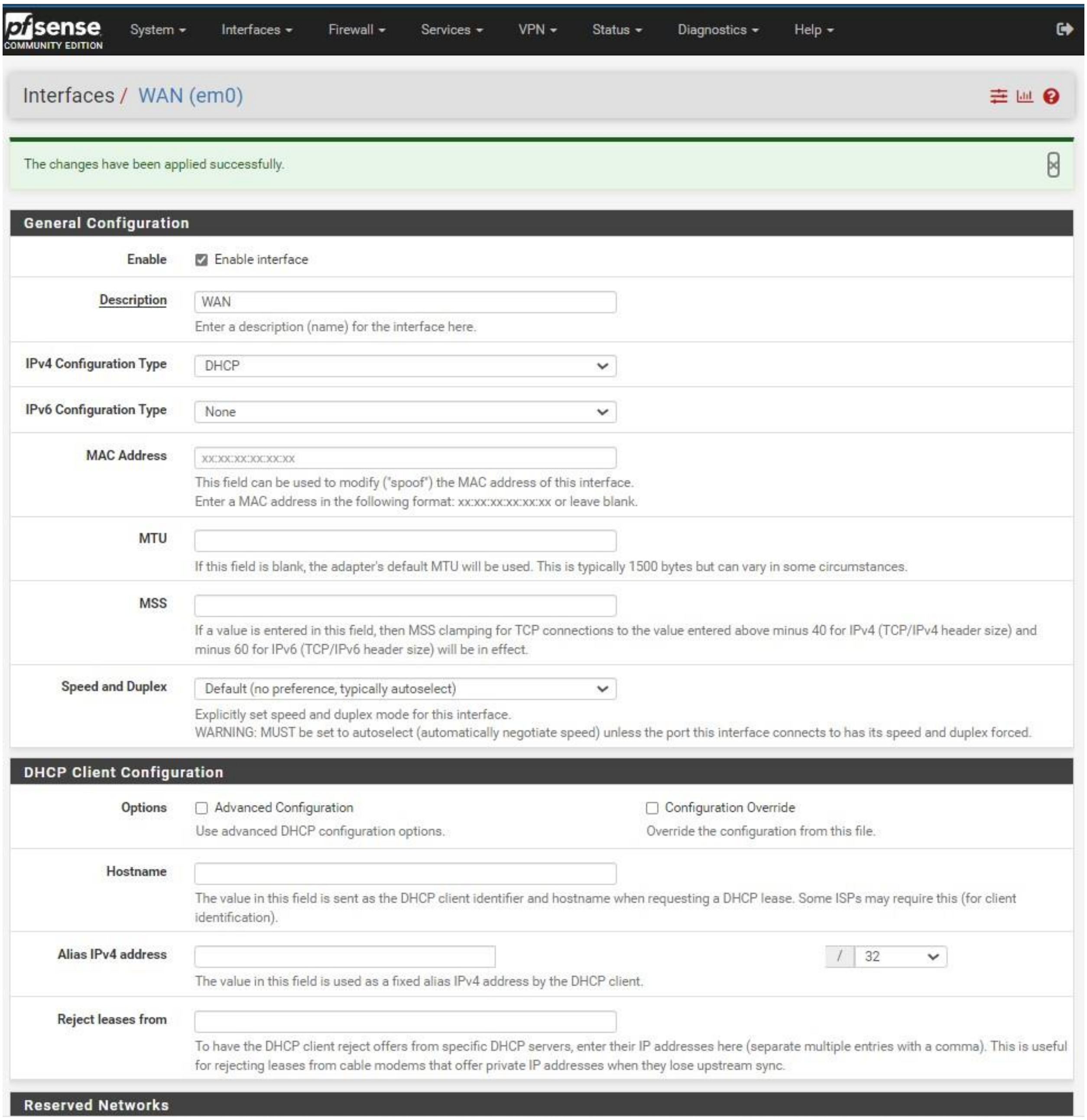
Traffic Graphs

Netgate Services And Support

Retrieving support information

Figure 4. 1 (Homepage)

``



Interfaces / WAN (em0)

The changes have been applied successfully.

General Configuration

Enable	☑ Enable interface
Description	WAN Enter a description (name) for the interface here.
IPv4 Configuration Type	DHCP
IPv6 Configuration Type	None
MAC Address	xx:xx:xx:xx:xx:xx This field can be used to modify ("spoof") the MAC address of this interface. Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank.
MTU	If this field is blank, the adapter's default MTU will be used. This is typically 1500 bytes but can vary in some circumstances.
MSS	If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 for IPv4 (TCP/IPv4 header size) and minus 60 for IPv6 (TCP/IPv6 header size) will be in effect.
Speed and Duplex	Default (no preference, typically autoselect) Explicitly set speed and duplex mode for this interface. WARNING: MUST be set to autoselect (automatically negotiate speed) unless the port this interface connects to has its speed and duplex forced.

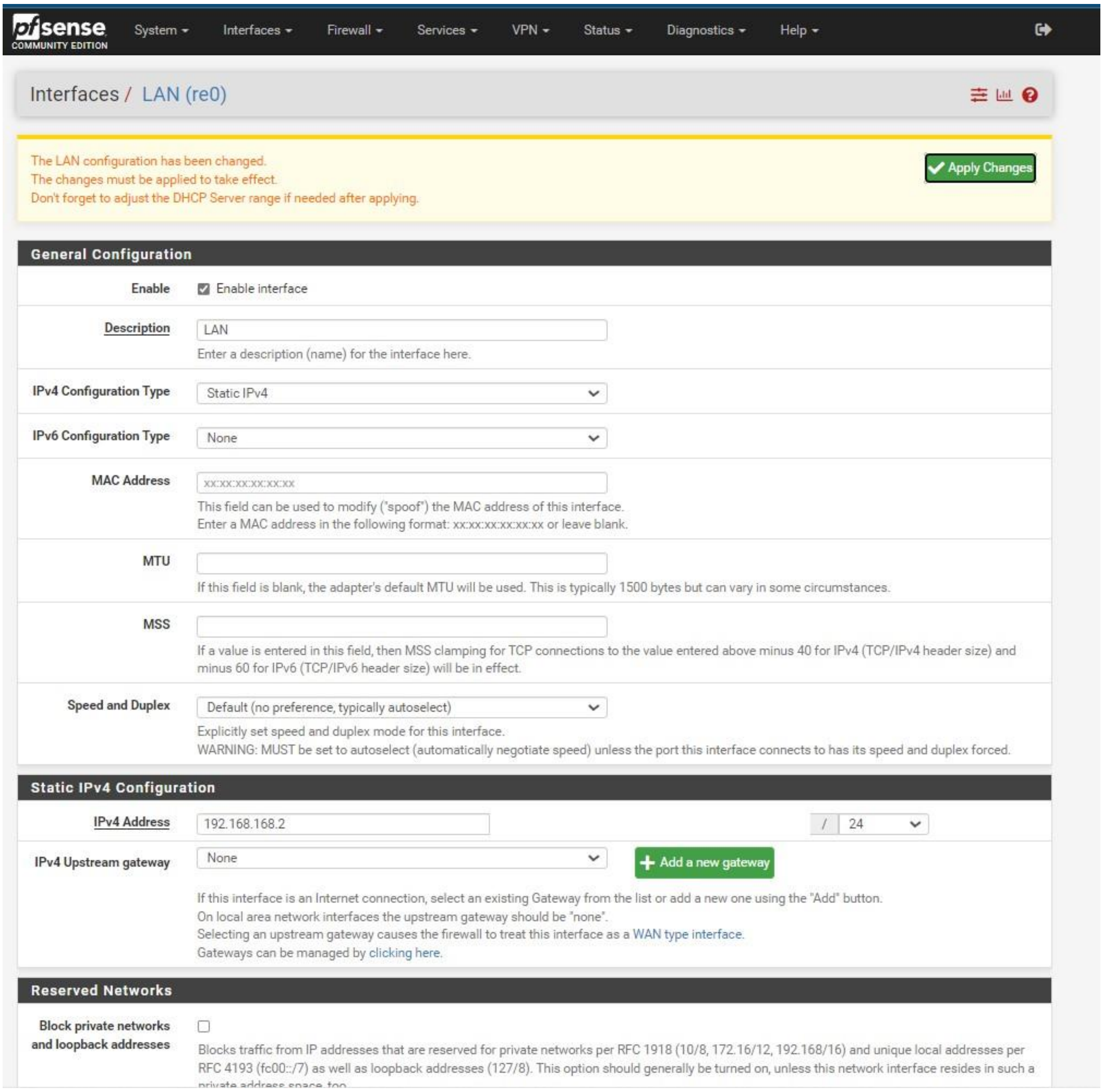
DHCP Client Configuration

Options	□ Advanced Configuration Use advanced DHCP configuration options. □ Configuration Override Override the configuration from this file.
Hostname	The value in this field is sent as the DHCP client identifier and hostname when requesting a DHCP lease. Some ISPs may require this (for client identification).
Alias IPv4 address	/ 32 The value in this field is used as a fixed alias IPv4 address by the DHCP client.
Reject leases from	To have the DHCP client reject offers from specific DHCP servers, enter their IP addresses here (separate multiple entries with a comma). This is useful for rejecting leases from cable modems that offer private IP addresses when they lose upstream sync.

Reserved Networks

Figure 4. 2 (Wan interface)

``



Interfaces / LAN (re0)

The LAN configuration has been changed.
The changes must be applied to take effect.
Don't forget to adjust the DHCP Server range if needed after applying.

General Configuration

Enable	☑ Enable interface
Description	LAN Enter a description (name) for the interface here.
IPv4 Configuration Type	Static IPv4
IPv6 Configuration Type	None
MAC Address	xx:xx:xx:xx:xx:xx This field can be used to modify ("spoof") the MAC address of this interface. Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank.
MTU	If this field is blank, the adapter's default MTU will be used. This is typically 1500 bytes but can vary in some circumstances.
MSS	If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 for IPv4 (TCP/IPv4 header size) and minus 60 for IPv6 (TCP/IPv6 header size) will be in effect.
Speed and Duplex	Default (no preference, typically autoselect) Explicitly set speed and duplex mode for this interface. WARNING: MUST be set to autoselect (automatically negotiate speed) unless the port this interface connects to has its speed and duplex forced.

Static IPv4 Configuration

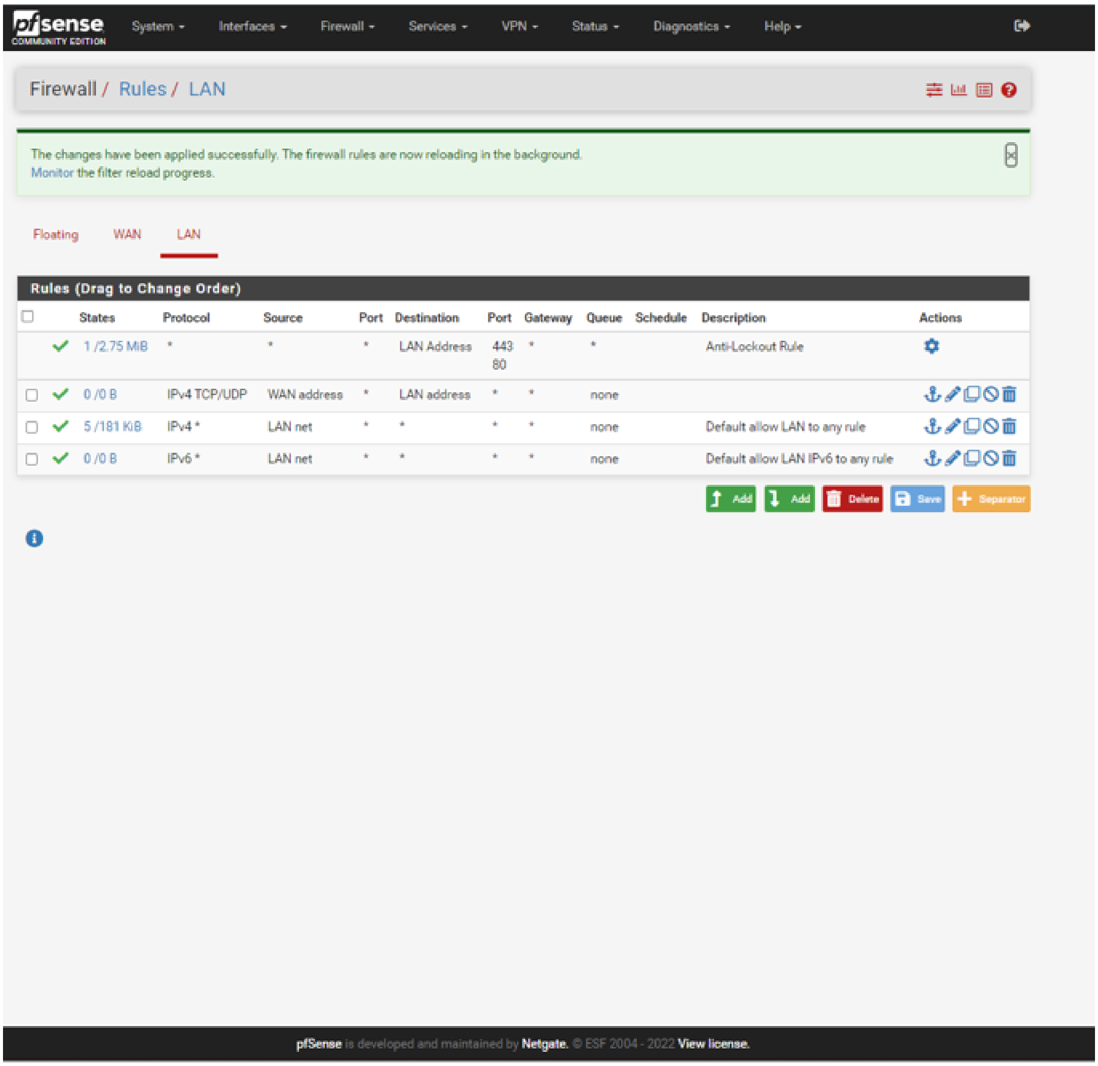
IPv4 Address	192.168.168.2 / 24
IPv4 Upstream gateway	None If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button. On local area network interfaces the upstream gateway should be "none". Selecting an upstream gateway causes the firewall to treat this interface as a WAN type interface. Gateways can be managed by clicking here.

Reserved Networks

Block private networks and loopback addresses	□ Blocks traffic from IP addresses that are reserved for private networks per RFC 1918 (10/8, 172.16/12, 192.168/16) and unique local addresses per RFC 4193 (fc00::/7) as well as loopback addresses (127/8). This option should generally be turned on, unless this network interface resides in such a private address space, too.

Figure 4. 3 (LAN interface)

``



Firewall / Rules / LAN

The changes have been applied successfully. The firewall rules are now reloading in the background.
Monitor the filter reload progress.

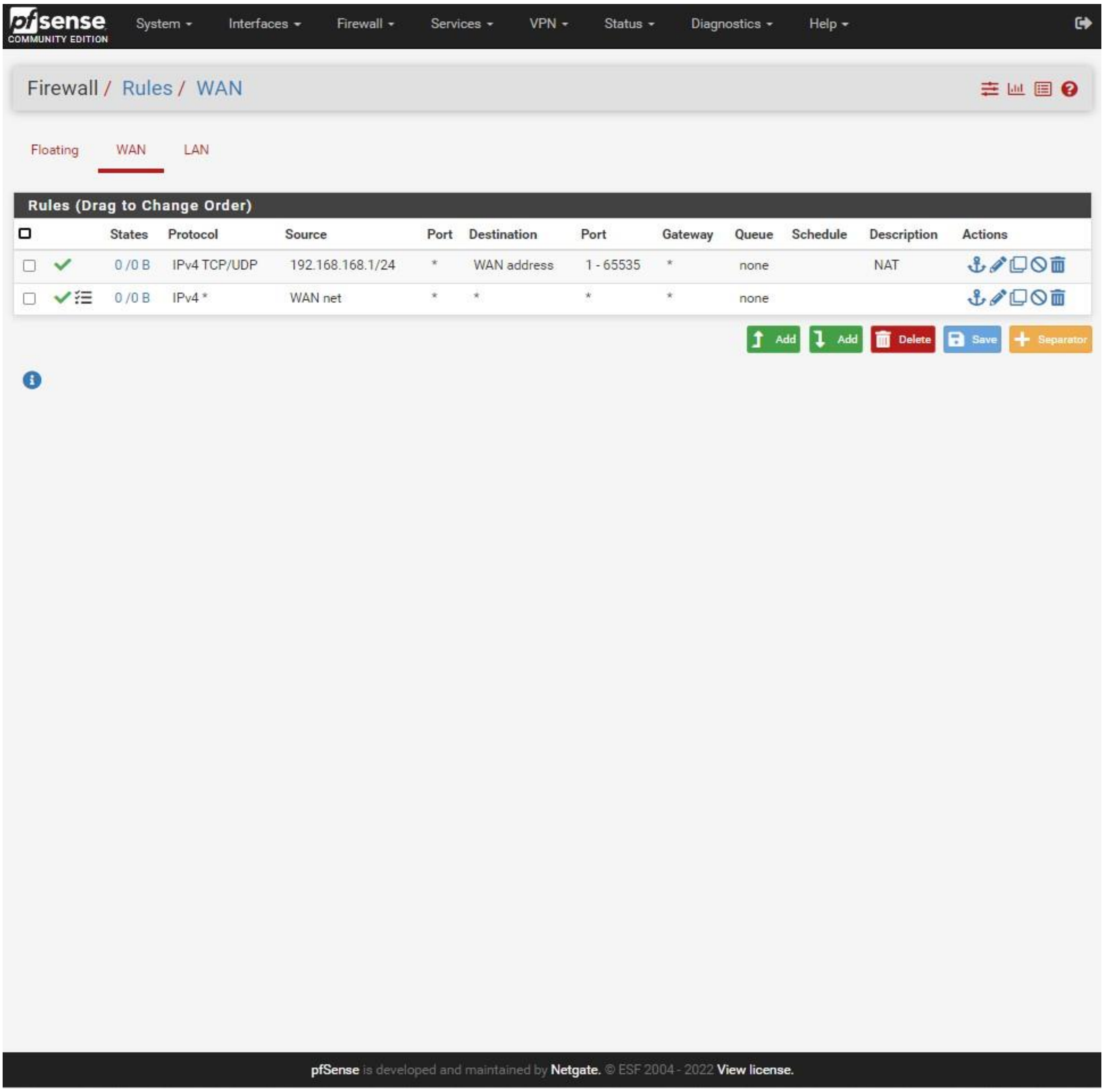
Rules (Drag to Change Order)

□		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✓	1 /2.75 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	
□	✓	0 /0 B	IPv4 TCP/UDP	WAN address	*	LAN address	*	*	none			
□	✓	5 /181 KiB	IPv4 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	
□	✓	0 /0 B	IPv6 *	LAN net	*	*	*	*	none		Default allow LAN IPv6 to any rule	

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

Figure 4. 4 (Rules for LAN)

``



Firewall / Rules / WAN

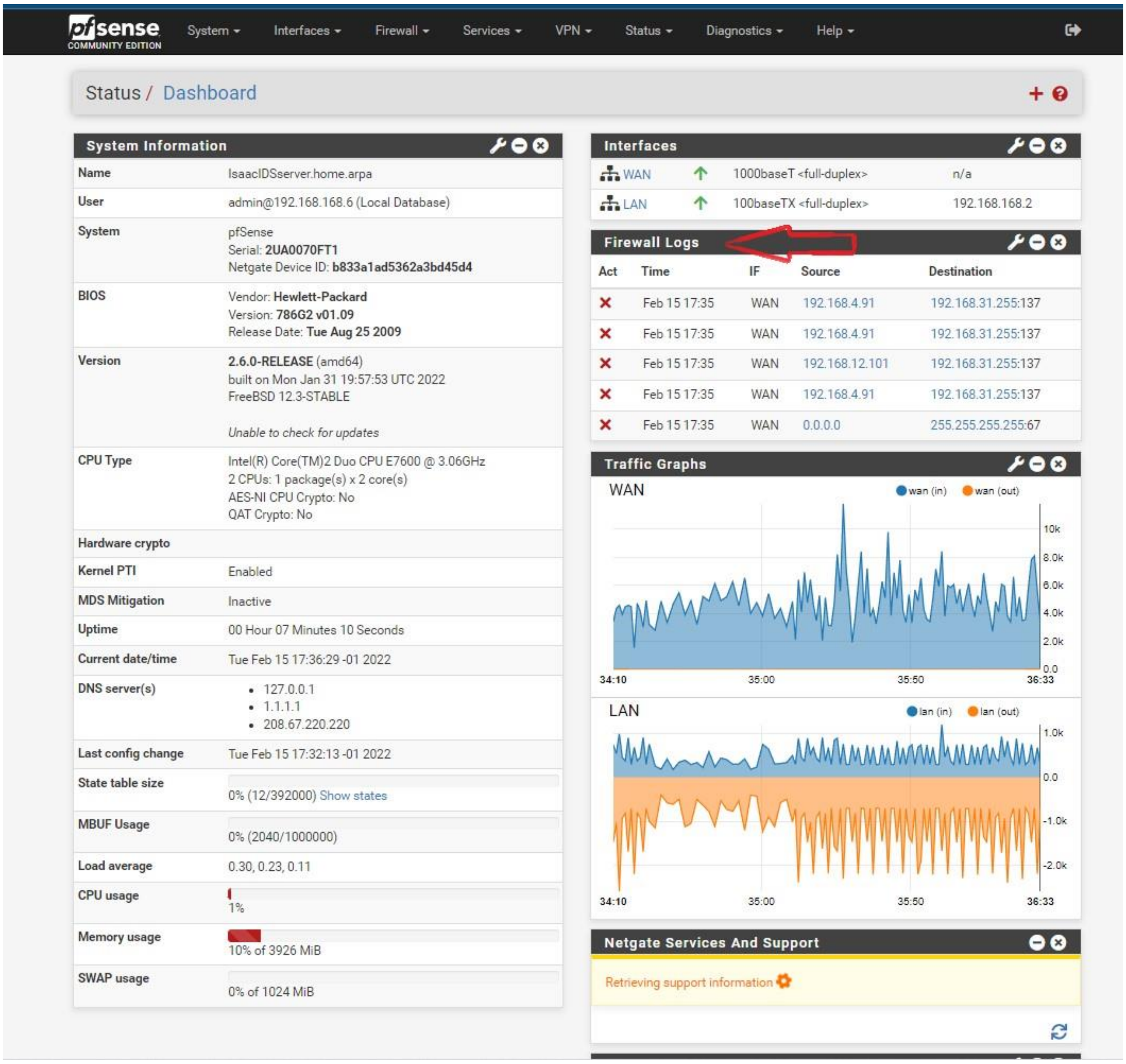
Rules (Drag to Change Order)

□		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
□	✓	0 /0 B	IPv4 TCP/UDP	192.168.168.1/24	*	WAN address	1 - 65535	*	none		NAT	
□	✓	0 /0 B	IPv4 *	WAN net	*	*	*	*	none			

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

Figure 4. 5 (Rules for WAN)

``



Status / Dashboard

System Information

Name	IsaacIDSserver.home.arpa
User	admin@192.168.168.6 (Local Database)
System	pfSense Serial: **2UA0070FT1** Netgate Device ID: **b833a1ad5362a3bd45d4**
BIOS	Vendor: **Hewlett-Packard** Version: **786G2 v01.09** Release Date: **Tue Aug 25 2009**
Version	**2.6.0-RELEASE** (amd64) built on Mon Jan 31 19:57:53 UTC 2022 FreeBSD 12.3-STABLE *Unable to check for updates*
CPU Type	Intel(R) Core(TM)2 Duo CPU E7600 @ 3.06GHz 2 CPUs: 1 package(s) x 2 core(s) AES-NI CPU Crypto: No QAT Crypto: No
Hardware crypto	
Kernel PTI	Enabled
MDS Mitigation	Inactive
Uptime	00 Hour 07 Minutes 10 Seconds
Current date/time	Tue Feb 15 17:36:29 -01 2022
DNS server(s)	• 127.0.0.1 • 1.1.1.1 • 208.67.220.220
Last config change	Tue Feb 15 17:32:13 -01 2022
State table size	0% (12/392000) Show states
MBUF Usage	0% (2040/1000000)
Load average	0.30, 0.23, 0.11
CPU usage	1%
Memory usage	10% of 3926 MiB
SWAP usage	0% of 1024 MiB

Interfaces

WAN	↑	1000baseT <full-duplex>	n/a
LAN	↑	100baseTX <full-duplex>	192.168.168.2

Firewall Logs

Act	Time	IF	Source	Destination
✕	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✕	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✕	Feb 15 17:35	WAN	192.168.12.101	192.168.31.255:137
✕	Feb 15 17:35	WAN	192.168.4.91	192.168.31.255:137
✕	Feb 15 17:35	WAN	0.0.0.0	255.255.255.255:67

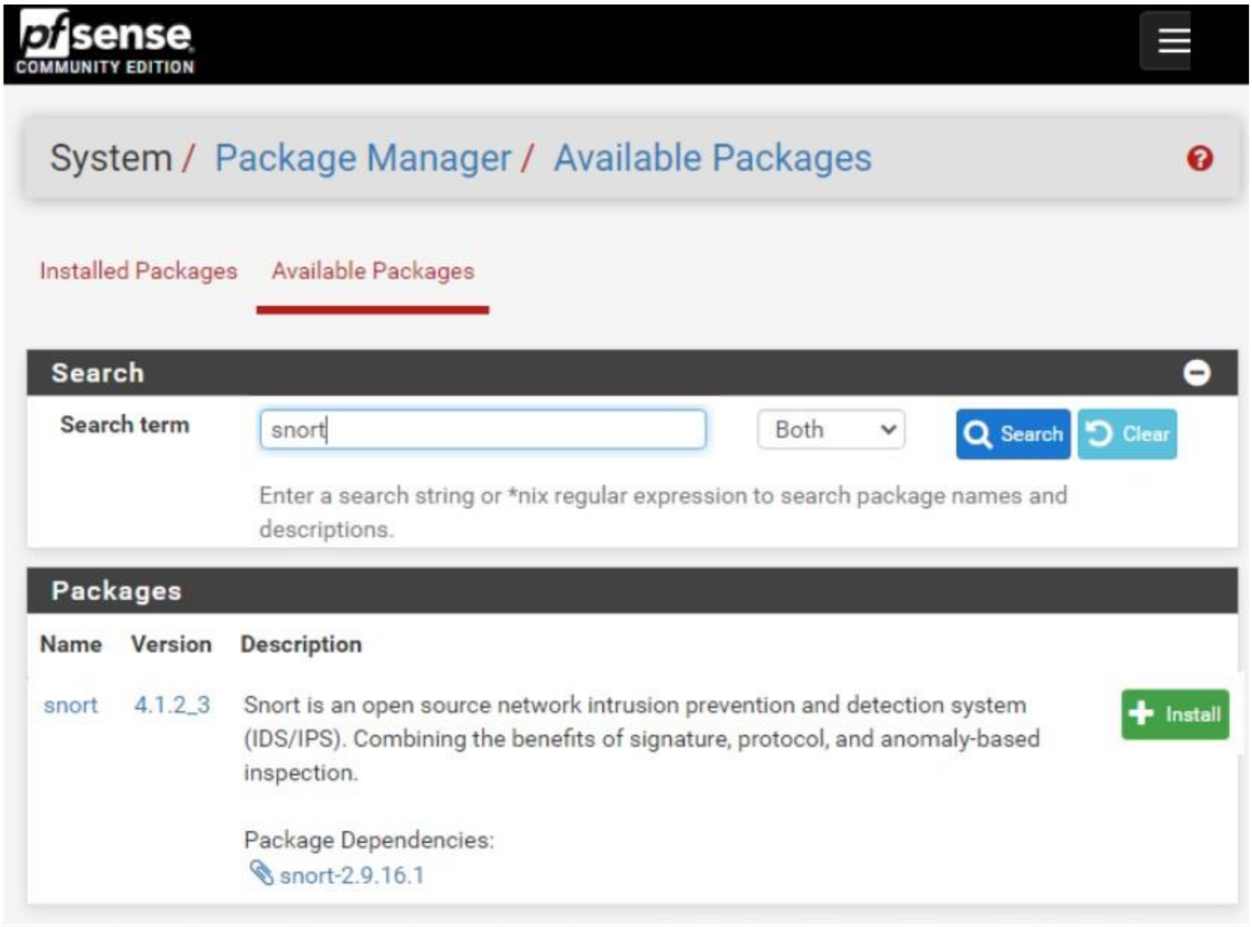
Traffic Graphs

Netgate Services And Support

Retrieving support information

Figure 4. 6 (Firewall logs)

``



System / Package Manager / Available Packages

Installed Packages | Available Packages

Search

Search term: snort | Both | Search | Clear

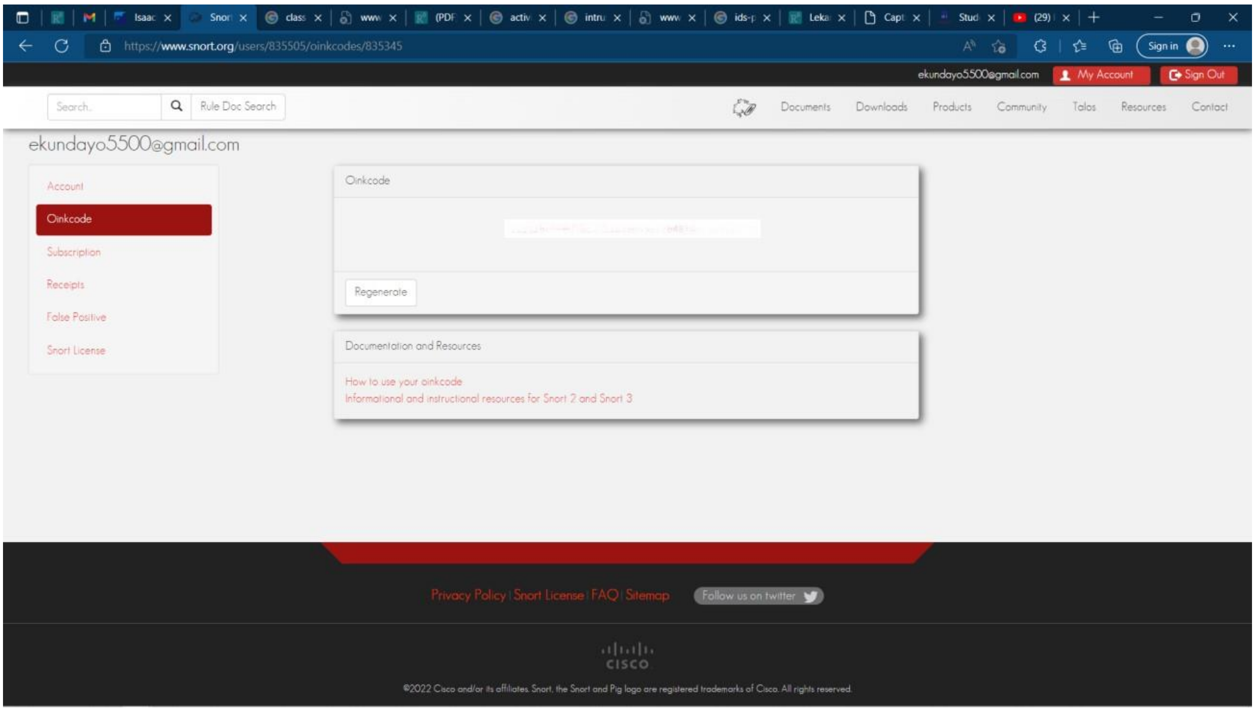
Enter a search string or *nix regular expression to search package names and descriptions.

Packages

Name	Version	Description
snort	4.1.2_3	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. Package Dependencies: snort-2.9.16.1

Figure 4. 7 (snort package)

``



ekundayo5500@gmail.com

Oinkcode

[redacted]

Documentation and Resources

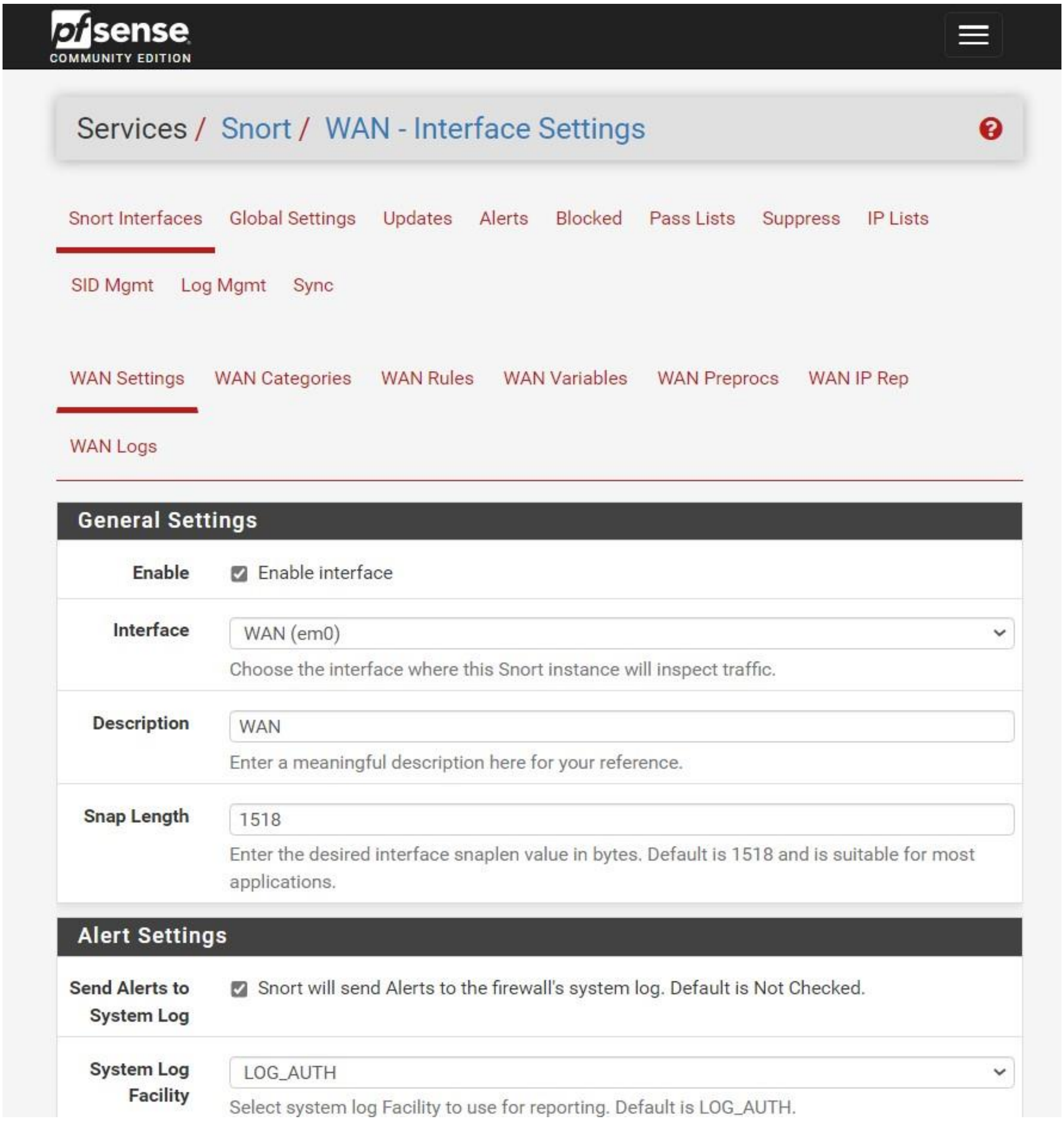
How to use your oinkcode

Informational and instructional resources for Snort 2 and Snort 3

©2022 Cisco and/or its affiliates. Snort, the Snort and Pig logo are registered trademarks of Cisco. All rights reserved.

Figure 4. 8 (Oincode)

``



General Settings

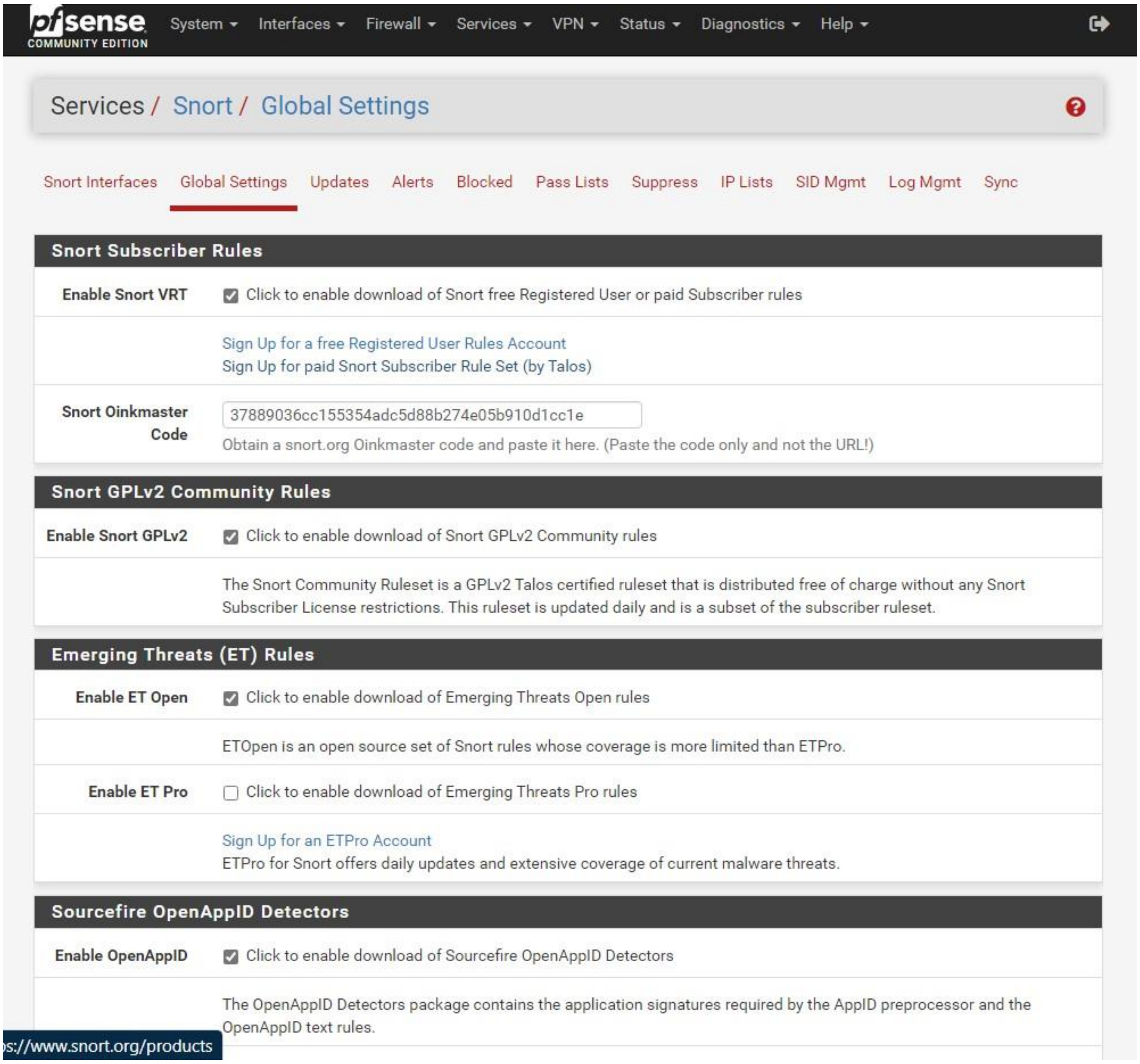
Enable	☑ Enable interface
Interface	WAN (em0) Choose the interface where this Snort instance will inspect traffic.
Description	WAN Enter a meaningful description here for your reference.
Snap Length	1518 Enter the desired interface snaplen value in bytes. Default is 1518 and is suitable for most applications.

Alert Settings

Send Alerts to System Log	☑ Snort will send Alerts to the firewall's system log. Default is Not Checked.
System Log Facility	LOG_AUTH Select system log Facility to use for reporting. Default is LOG_AUTH.

Figure 4. 9 (snort interfaces)

``



Snort Subscriber Rules

Enable Snort VRT ☑ Click to enable download of Snort free Registered User or paid Subscriber rules

Sign Up for a free Registered User Rules Account
Sign Up for paid Snort Subscriber Rule Set (by Talos)

Snort Oinkmaster Code: 37889036cc155354adc5d88b274e05b910d1cc1e
Obtain a snort.org Oinkmaster code and paste it here. (Paste the code only and not the URL!)

Snort GPLv2 Community Rules

Enable Snort GPLv2 ☑ Click to enable download of Snort GPLv2 Community rules

The Snort Community Ruleset is a GPLv2 Talos certified ruleset that is distributed free of charge without any Snort Subscriber License restrictions. This ruleset is updated daily and is a subset of the subscriber ruleset.

Emerging Threats (ET) Rules

Enable ET Open ☑ Click to enable download of Emerging Threats Open rules

ETOpen is an open source set of Snort rules whose coverage is more limited than ETPro.

Enable ET Pro □ Click to enable download of Emerging Threats Pro rules

Sign Up for an ETPro Account
ETPro for Snort offers daily updates and extensive coverage of current malware threats.

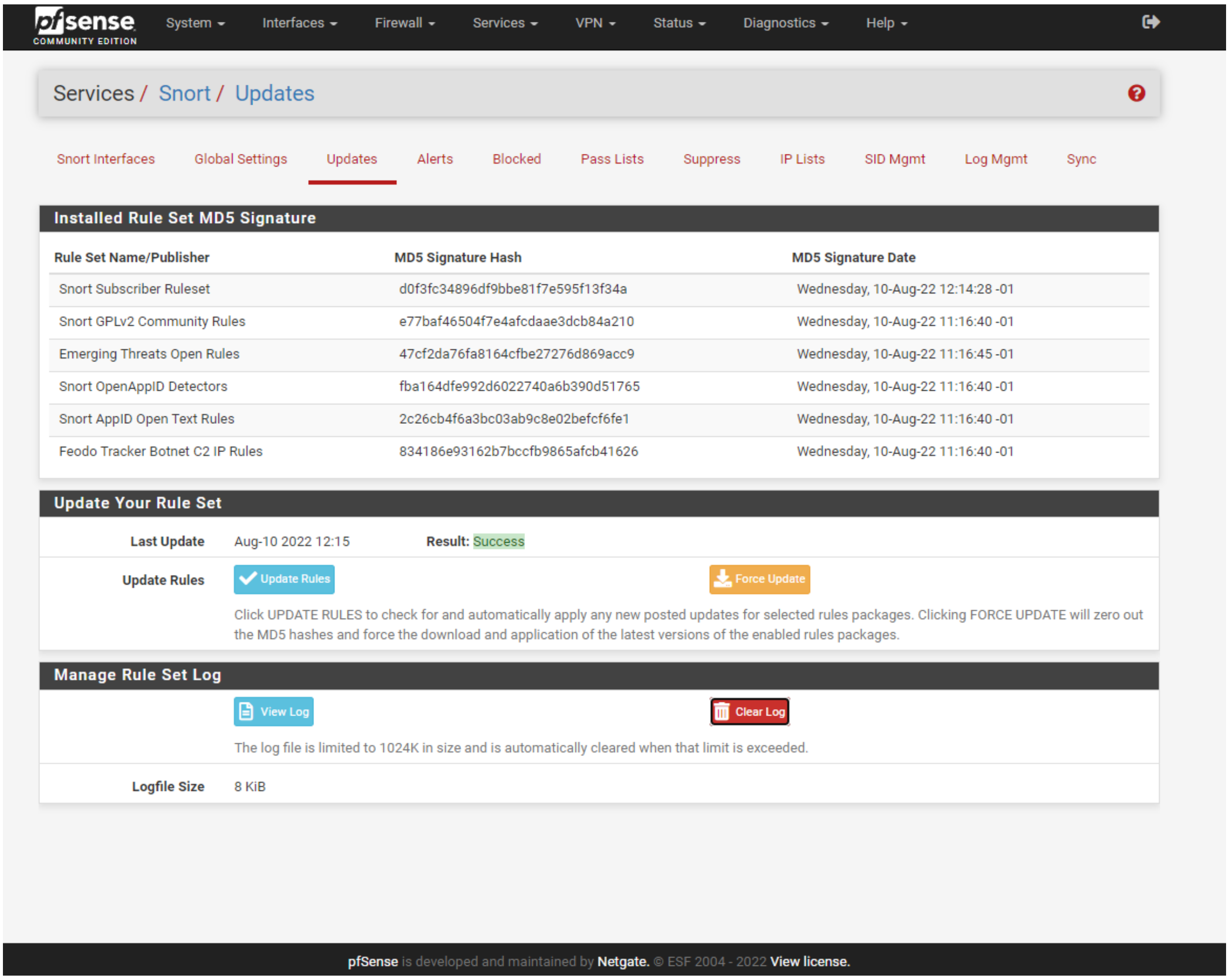
Sourcefire OpenAppID Detectors

Enable OpenAppID ☑ Click to enable download of Sourcefire OpenAppID Detectors

The OpenAppID Detectors package contains the application signatures required by the AppID preprocessor and the OpenAppID text rules.

Figure 4. 10 (Global settings)

``



Services / Snort / Updates

Installed Rule Set MD5 Signature

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Snort Subscriber Ruleset	d0f3fc34896df9bbe81f7e595f13f34a	Wednesday, 10-Aug-22 12:14:28 -01
Snort GPLv2 Community Rules	e77baf46504f7e4afcdaae3dcb84a210	Wednesday, 10-Aug-22 11:16:40 -01
Emerging Threats Open Rules	47cf2da76fa8164cfbe27276d869acc9	Wednesday, 10-Aug-22 11:16:45 -01
Snort OpenAppID Detectors	fba164dfe992d6022740a6b390d51765	Wednesday, 10-Aug-22 11:16:40 -01
Snort AppID Open Text Rules	2c26cb4f6a3bc03ab9c8e02befcf6fe1	Wednesday, 10-Aug-22 11:16:40 -01
Feodo Tracker Botnet C2 IP Rules	834186e93162b7bccfb9865afcb41626	Wednesday, 10-Aug-22 11:16:40 -01

Update Your Rule Set

Last Update Aug-10 2022 12:15 **Result:** Success

Update Rules ✓ Update Rules Force Update

Click UPDATE RULES to check for and automatically apply any new posted updates for selected rules packages. Clicking FORCE UPDATE will zero out the MD5 hashes and force the download and application of the latest versions of the enabled rules packages.

Manage Rule Set Log

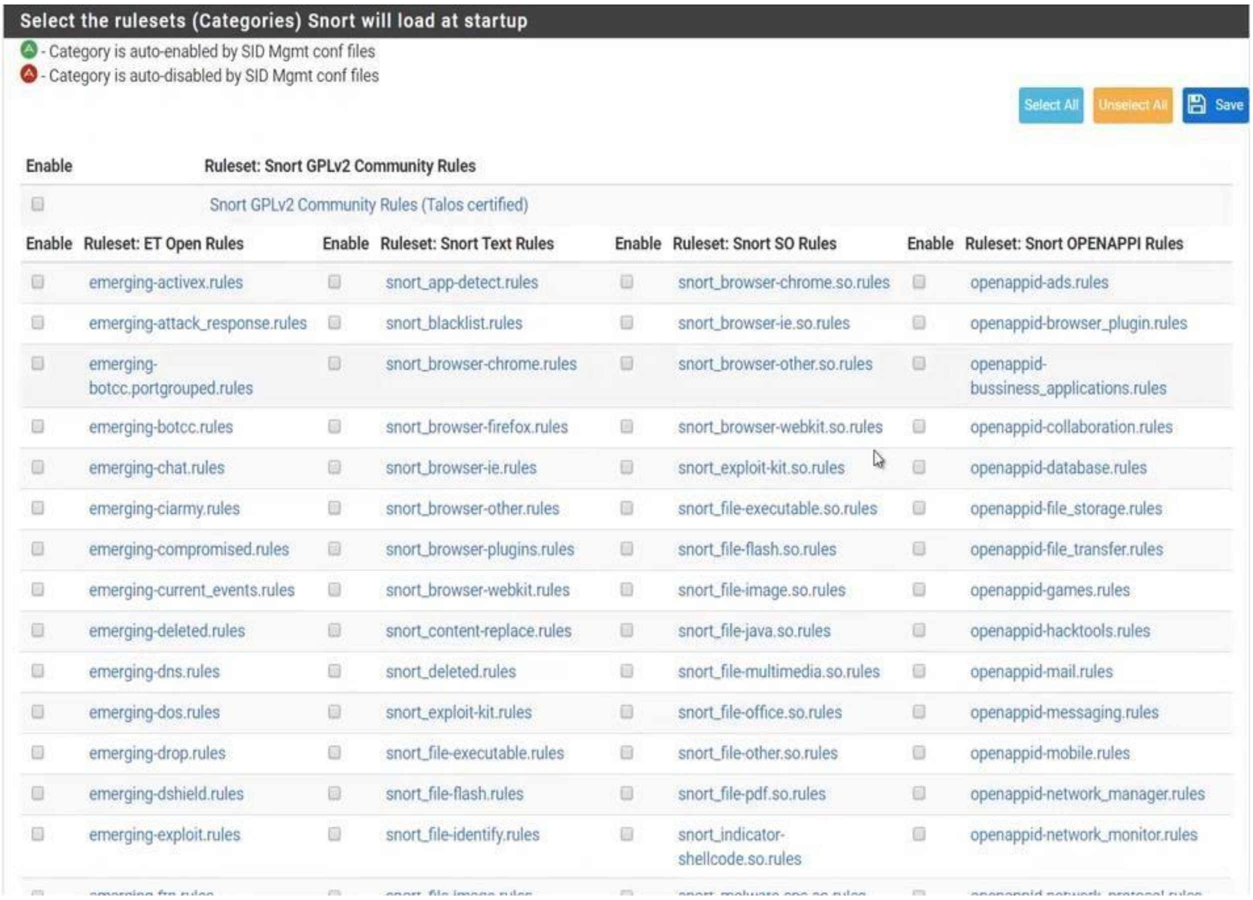
View Log Clear Log

The log file is limited to 1024K in size and is automatically cleared when that limit is exceeded.

Logfile Size 8 KiB

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

``



Select the rulesets (Categories) Snort will load at startup

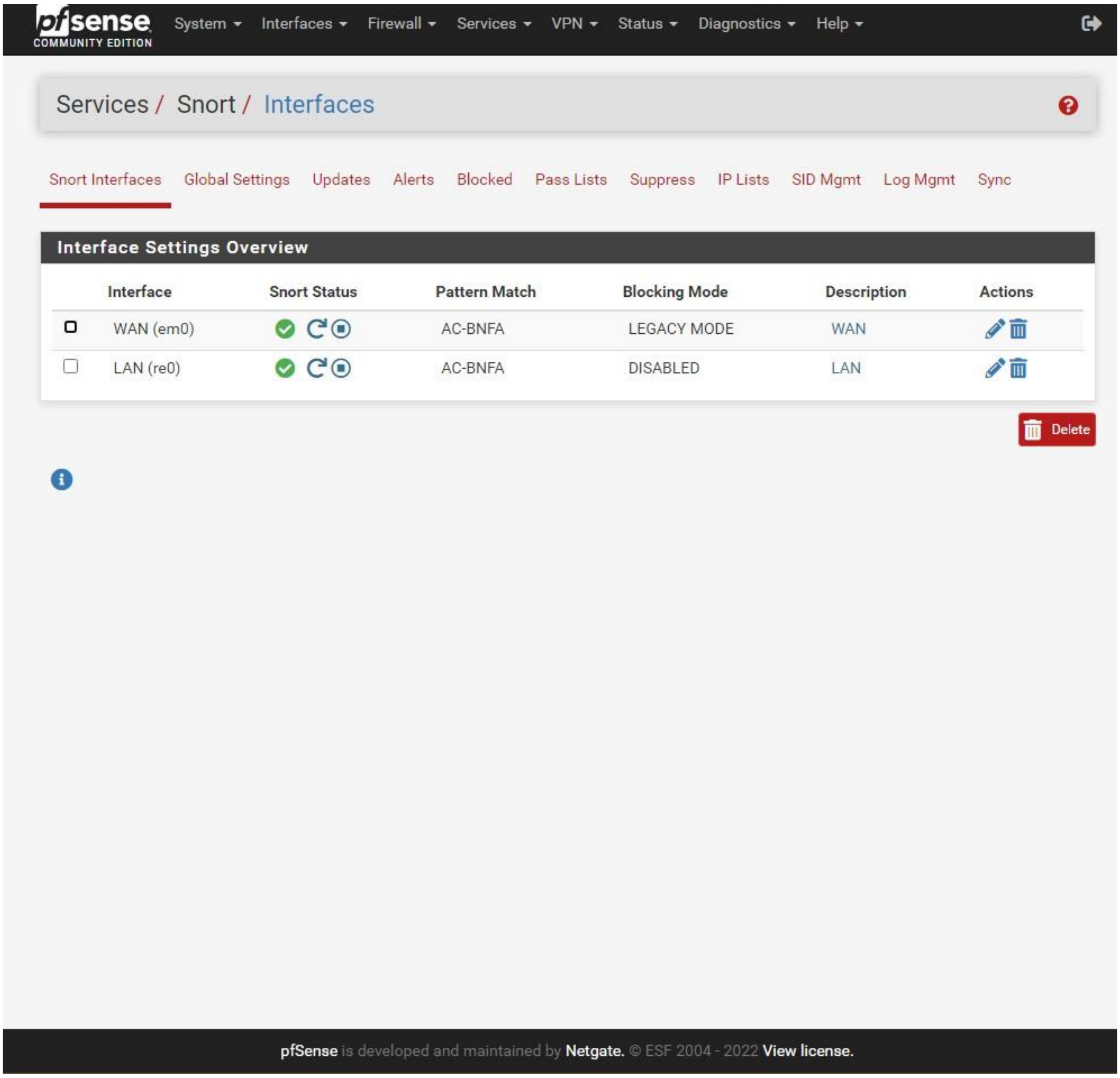
🟢 - Category is auto-enabled by SID Mgmt conf files
🔴 - Category is auto-disabled by SID Mgmt conf files

Enable	Ruleset: Snort GPLv2 Community Rules
□	Snort GPLv2 Community Rules (Talos certified)

Enable	Ruleset: ET Open Rules	Enable	Ruleset: Snort Text Rules	Enable	Ruleset: Snort SO Rules	Enable	Ruleset: Snort OPENAPPI Rules
□	emerging-activex.rules	□	snort_app-detect.rules	□	snort_browser-chrome.so.rules	□	openappid-ads.rules
□	emerging-attack_response.rules	□	snort_blacklist.rules	□	snort_browser-ie.so.rules	□	openappid-browser_plugin.rules
□	emerging-botcc.portgrouped.rules	□	snort_browser-chrome.rules	□	snort_browser-other.so.rules	□	openappid-bussiness_applications.rules
□	emerging-botcc.rules	□	snort_browser-firefox.rules	□	snort_browser-webkit.so.rules	□	openappid-collaboration.rules
□	emerging-chat.rules	□	snort_browser-ie.rules	□	snort_exploit-kit.so.rules	□	openappid-database.rules
□	emerging-ciarmy.rules	□	snort_browser-other.rules	□	snort_file-executable.so.rules	□	openappid-file_storage.rules
□	emerging-compromised.rules	□	snort_browser-plugins.rules	□	snort_file-flash.so.rules	□	openappid-file_transfer.rules
□	emerging-current_events.rules	□	snort_browser-webkit.rules	□	snort_file-image.so.rules	□	openappid-games.rules
□	emerging-deleted.rules	□	snort_content-replace.rules	□	snort_file-java.so.rules	□	openappid-hacktools.rules
□	emerging-dns.rules	□	snort_deleted.rules	□	snort_file-multimedia.so.rules	□	openappid-mail.rules
□	emerging-dos.rules	□	snort_exploit-kit.rules	□	snort_file-office.so.rules	□	openappid-messaging.rules
□	emerging-drop.rules	□	snort_file-executable.rules	□	snort_file-other.so.rules	□	openappid-mobile.rules
□	emerging-dshield.rules	□	snort_file-flash.rules	□	snort_file-pdf.so.rules	□	openappid-network_manager.rules
□	emerging-exploit.rules	□	snort_file-identify.rules	□	snort_indicator-shellcode.so.rules	□	openappid-network_monitor.rules
□	emerging-ftp.rules	□	snort_file-image.rules	□	snort_malware-cnc.so.rules	□	openappid-network_protocol.rules

Figure 4. 11 (update rules and interface rules)

``



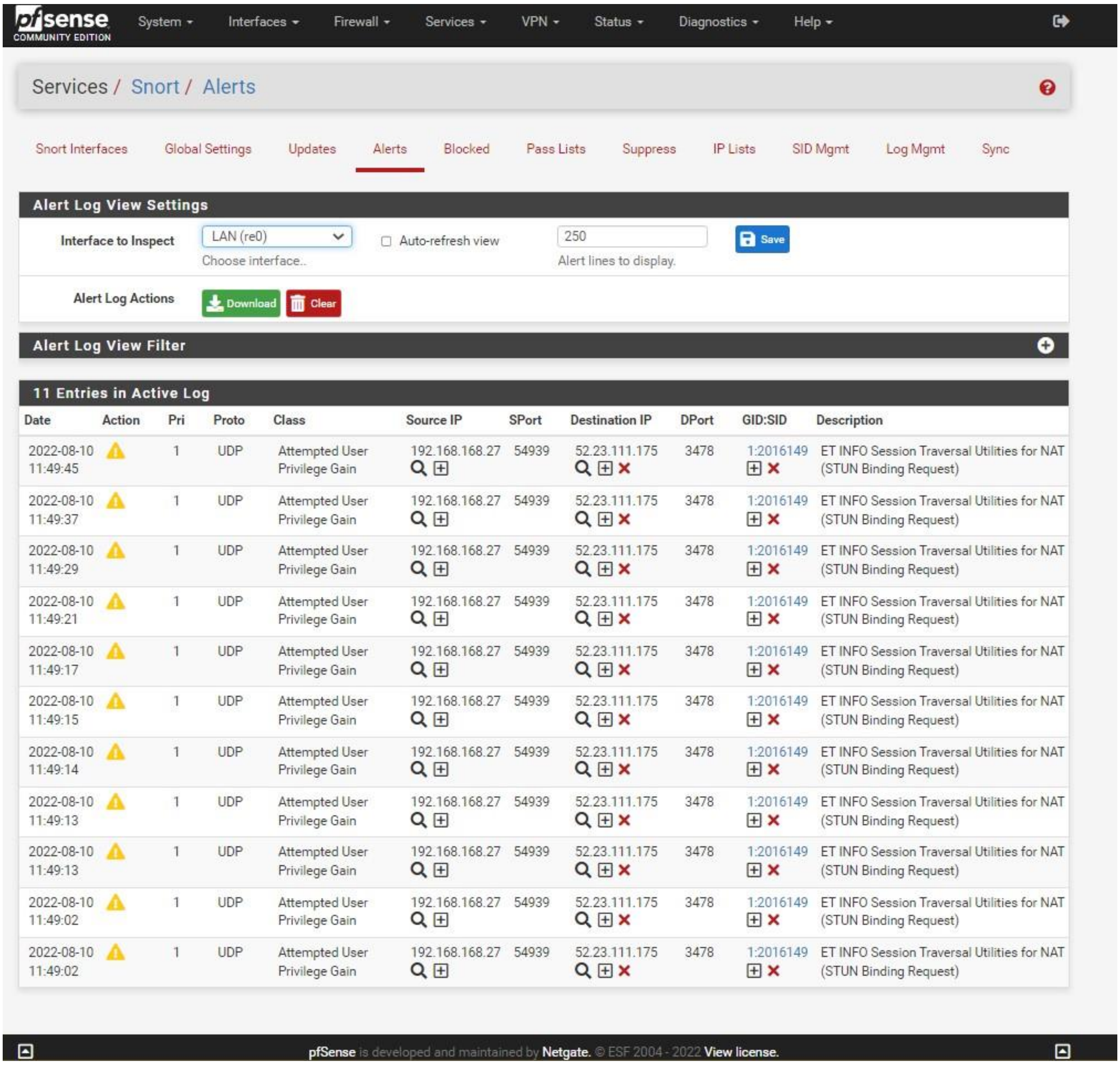
Interface Settings Overview

	Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
□	WAN (em0)	✓	AC-BNFA	LEGACY MODE	WAN	
□	LAN (re0)	✓	AC-BNFA	DISABLED	LAN	

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

Figure 4. 12 (WAN and LAN interface activation page)

``



Services / Snort / Alerts

Alert Log View Settings

Interface to Inspect: LAN (re0) — Choose interface..

□ Auto-refresh view

250 — Alert lines to display.

Alert Log Actions

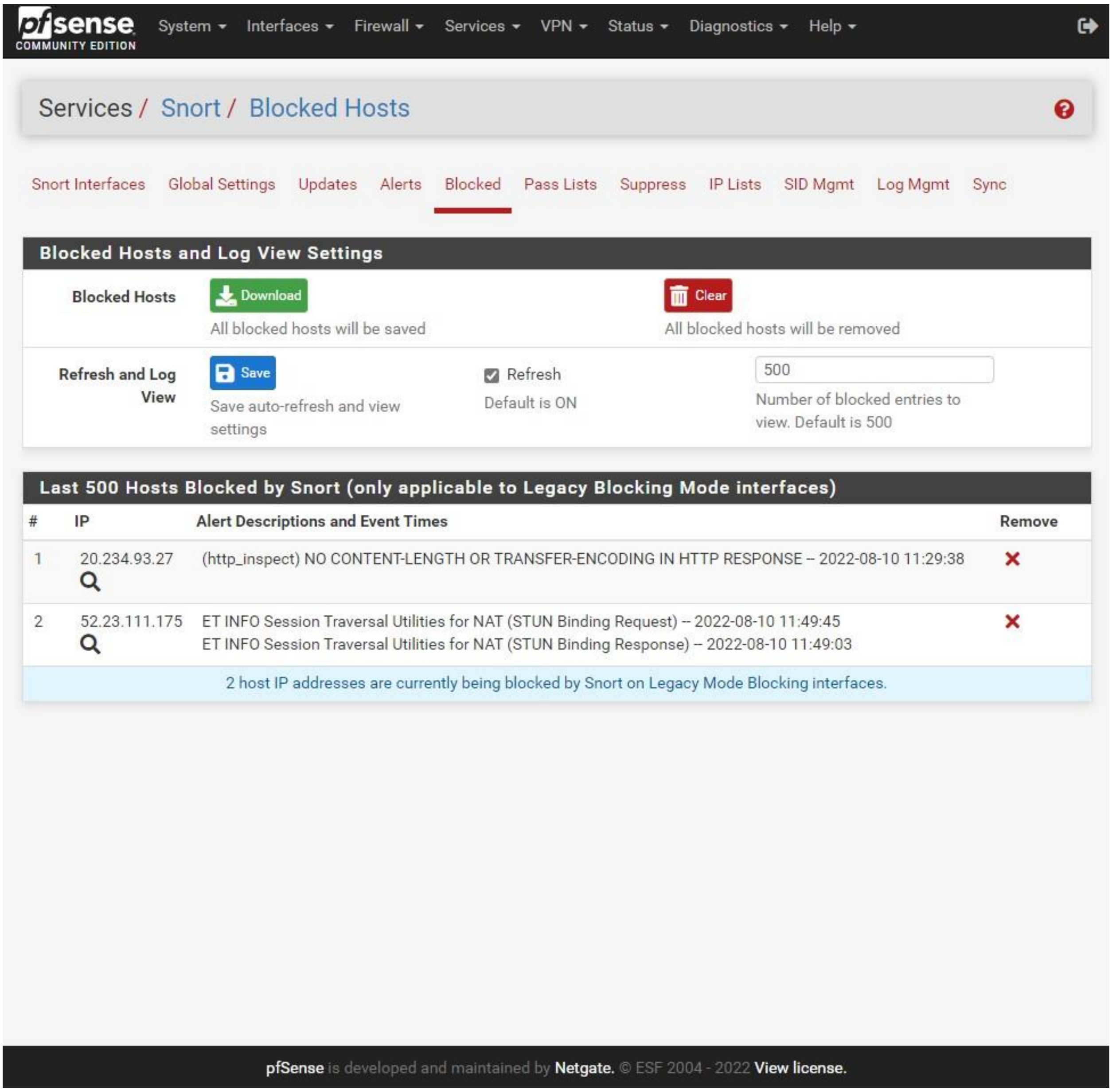
Alert Log View Filter

11 Entries in Active Log

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2022-08-10 11:49:45	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:37	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:29	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:21	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:17	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:15	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:14	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:13	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:13	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:02	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)
2022-08-10 11:49:02	⚠	1	UDP	Attempted User Privilege Gain	192.168.168.27 🔍 ⊞	54939	52.23.111.175 🔍 ⊞ ✕	3478	1:2016149 ⊞ ✕	ET INFO Session Traversal Utilities for NAT (STUN Binding Request)

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

``



Services / Snort / Blocked Hosts

Blocked Hosts and Log View Settings

Blocked Hosts: Download — All blocked hosts will be saved; Clear — All blocked hosts will be removed

Refresh and Log View: Save — Save auto-refresh and view settings; ☑ Refresh — Default is ON; 500 — Number of blocked entries to view. Default is 500

Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)

#	IP	Alert Descriptions and Event Times	Remove
1	20.234.93.27	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE -- 2022-08-10 11:29:38	✕
2	52.23.111.175	ET INFO Session Traversal Utilities for NAT (STUN Binding Request) -- 2022-08-10 11:49:45 ET INFO Session Traversal Utilities for NAT (STUN Binding Response) -- 2022-08-10 11:49:03	✕

2 host IP addresses are currently being blocked by Snort on Legacy Mode Blocking interfaces.

pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 View license.

Figure 4. 13 (alerts and the blocked host)

``

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1. Summary

This study developed an intrusion detection system that allows the network administrator to effectively monitor access of intrusions or suspicious acts using alerts as well as check traffic. The user and system requirements that were necessary to be met by the system were identified alongside the software and hardware requirements of the system during this study.

5.2. Conclusion

In conclusion, this study has designed and implemented an intrusion detection system that solves the challenges of an intrusion or suspicious activity. The study was able to identify the various constraints that characterized this process and, by this define the system and user requirements.

The designs of this system were also adequately specified with relevant diagrams so as suit to the expected functions of the proposed system. The system itself has some limitations on its implementation due to the numerous malware and malicious software being created nowadays. More rules need to be created and updated.

5.3. Recommendation

This system's identification mechanism can be enhanced; the system uses signature-based software, which might not see some activity if it is malicious. If this happens, it is advisable that the administrator add an anomaly-based system to make it a hybrid system which combines the two so it can detect other malicious attacks.

``

References

Aggrey, O. (2009). An Intrusion Detection System For Academic Institutions. 5.

Agrawal, R. (2012). A study of methodologies used in intrusion detection and prevention system (IDPS). 3.

Ajith, A., G, C., & C, Y. (2001). Cyber Security and the Evolution of Intrusion Detection Systems. *Information Management and Computer Security*, 175-182.

Alazab, A. (2014). Using response action with intelligent intrusion detection and prevention system against web application malware. *Information Management & Computer Security,*, 431-449.

Alazab, A., Hobb, M., Abawajy, J., & Alazab, M. (2012). Using feature selection for intrusion detection system. *International symposium on communications and information technologies (ISCIT)*, 296-301.

Australian. (2017, November). .*Australian cyber security center threat report.* Retrieved from https://www.acsc.gov.au/publications/ACSC_Threat_Report_2017.pdf

Bhuyan, M., & Bhattacharyya DK, K. J. (2014). Network anomaly detection: methods system and tools. *IEEE Communications Surveys & Tutorials*, 303–336.

Breach_LeveL_Index. (2017, November). *Data breach statistics.* Retrieved from http://breachlevelindex.com/

Creech, G., & J, H. (2014). A semantic approach to host-based intrusion detection. 807–819.

Denning, D. (1967). An intrusion detection model. *IEEE Transactions on Software Engineering*, 222-32.

G, V., & RA, K. (1999). NetSTAT: a network-based intrusion detection system. *J Comput Secur* , 37-72.

Galal, H. S., Mahdy, Y. B., & Atiea, M. A. (2016). Behavior-based features model for malware detection. *Journal of Computer Virology and Hacking Techniques*, 59-67.

Hubballi, N., & Vinoth, S. (2014). False alarm minimization techniques in signature-based intrusion detection systems. *A survey," Computer Communications*, 1-17.

Hwang, K. Q. (2007). Hybrid Intrusion Detection with Weighted Signature Generation over

Anomalous Internet Episodes. *IEEE Transactions on Dependable Computing*, 41-55.

I, B., SD, M., & R, S. (2014). A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 266–282.

Jiujiu, Y. (2018). Research Process on Software Development Model. IOP Conference Series.

Materials Science and Engineering. IOP publishing Ltd.

Joseph, S. a. (2003). Intrusion detection:methods and systems. Part II. *Information Management and Computer Security*, 11(5):222-229.

Kendall, K. (1999). A Database of Computer attacks for Evaluation of Intrusion Detection Systems. *PhD Thesis, Massachusetts Insitute of Technlogy,Boston.*

Khraisat, A, G., I, V., & P. (2018). An anomaly intrusion detection ystem using C5 decision tree classifier. *Trends and applications in knowledge discovery and data mining*, 49–155.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems:techniques, datasets and challenges. *Cybersecurity*, 1-22.

Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 1690-1700.

Kreibic, C., & Crowcroft, J. (2004). Honeycomb: creating intrusion detection gnatures using honeypots. *SIGCOMM Comput Commun,* 51–56.

Langsari, K. (n.d.). Intrusion Detection System (IDS). *Institut Teknologl Sepuluh Nopember*, 810.

Lee, W., & Stolfo, S. (1998). Data mining approaches for intrusion detection. *Proceedings of the 1998 USENIX Security symposium*.

Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). "A survey of Intrusion detection techniques in cloud. *J Netw Comput App*, 42–57.

Munassar, N. M. (2010). Comparison Between Five Models Of Software. *International Journal Of Computer Science Issues(Ijcsi)*, 7(5), 94-101.

Roesch, M. (1999). Snort-lightweight intrusion detection for networks. *In Proceedings of the 13th USENIX conference on system administration*, 229–238.

Sodiya, A., & Akinwale, A. (2004). A new two - tiered strategy to intrusion detection. *Information Management and Computer Security*, 27-44.

Sundaram, A. (1991). An introduction to intrusion detection,Crossroads. *The ACM Student Magazine*, 2. Retrieved from acm.org/Crossroads.

Syed, A., Raza, S., & Biju, I. (2018). Performance comparison of intrusion detection systems and application of machine learning to Snort system. *Future Generation Computer Systems*, 157-170.

Symantec. (2017). Internet security threat report .

Thakur, A. S. (2015). Comparing Various SDLC Models On The Basis Of Available Methodology. *International Journal Of Modern Engineering Research (IJMER)*, 5(3), 34-39.

Youssef, B. (2012). A Simulation Model for the Waterfall Software Development. *International Journal of Engineering & Technology(iJET)*.

``

Zhang, Y. S. (2006). Software Development Models. *A survey*. *Journal of Computer Engineering and Applications*, 109-110.

Zhou, J., Carlson, A., & Bishop, M. (2005). Verify Results of Network Intrusion Alerts Using Lightweight Protocol Analysis. *Proceedings of the 21st Annual Computer Security and Applications Conference(ACSAC 2005)*.