Digital Communications and Networks 8 (2022) 540–551

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

Intrusion detection and prevention system for an IoT environment

Ajay Kumar [a], K. Abhishek [a], M.R. Ghalib [b], A. Shankar [c,*], X. Cheng [d]

[a] Department of Computer Science and Engineering, NIT Patna, Bihar, India
[b] Faculty of Science, Engineering Computing(SEC), De Montfort University, Dubai, United Arab Emirates
[c] Department of CSE, Amity School of Engineering and Technology, Amity University, Uttar Pradesh, India
[d] Department of Computer Science, Middlesex University, London, UK

ARTICLE INFO

Keywords:
IDS and IPS
Cloud
Firewall
Denial of service
Smart environment
Internet of things
NBIPS

ABSTRACT

Internet of Things (IoT) security is the act of securing IoT devices and networks. IoT devices, including industrial machines, smart energy grids, and building automation, are extremely vulnerable. With the goal of shielding network systems from illegal access in cloud servers and IoT systems, Intrusion Detection Systems (IDSs) and Network-based Intrusion Prevention Systems (NBIPSs) are proposed in this study. An intrusion prevention system is proposed to realize NBIPS to safeguard top to bottom engineering. The proposed NBIPS inspects network activity streams to identify and counteract misuse instances. The NBIPS is usually located specifically behind a firewall, and it provides a reciprocal layer of investigation that adversely chooses unsafe substances. Network-based IPS sensors can be installed either in an inline or a passive model. An inline sensor is installed to monitor the traffic passing through it. The sensors are installed to stop attacks by blocking the traffic using an IoT signature-based protocol.

1. Introduction

Hacking is a major problem in today's world connected by the Internet and other secure networks [1–6]. Hackers keep finding new ways to breach networks and steal valuable information about the victim or install malicious software to monitor the financial activity of the victim [7–9]. A hacker searches for and manipulates inadequacies in a Personal Computer(PC) or PC arranged associations. Hackers may be software engineers driven by various motivations such as financial gain, differences with the victim, testing hacking ability or network security, or the joy of hacking [4,10–13]. A hacker can gain control or intrude into a user's system only through bypassing the network security set up by the remote user, which can be done using specific software and stealing identity or damaging the security system by launching massive Denial-of-Service attacks. Security and data breaches cause massive financial losses of billions of dollars to industries [14–17]. According to a recent report by Inc. magazine, companies lose $400billion to hackers each year, and the loss is estimated to have grown to $1 trillion by 2020. Hence, it is essential to study effective mechanisms that stop network attacks and secure the work environment [10,18–20].

A PC system involves several PCs and other equipment segments interconnected through correspondence channels for sharing assets and data. The gadgets inside the system are alluded to as hubs [21–23]. In our study, neighborhood Local Area Networks (LANs) are concentrated, and the system activity is checked for recognizing any instance of interruptions [24]. LANs are sent inside the associations, instructive organizations, etc., which mainly deal with asset sharing and building up viable correspondence among the hubs available inside the system. A wide variety of security approaches is adopted when building up LANs [25,26]. Still, extremely advanced hacking assaults are considered a risk for LANs.

A firewall is the fundamental level of safety measures created before the stages. With advances in systems for identifying interruption in networks, the number of assaults has also increased. For example, neighborhood incorporates various hubs associated together to share the assets or data. Any type of system is defenseless to pernicious assaults [11,27]. The underlying level of security for the hubs inside a system constitutes firewalls, antivirus programming, and so forth; however, these safety efforts themselves provide an escape path for assailants attempting to break into the framework [24,28].

A passive IoT-based sensor is installed to monitor a copy of actual traffic in the network; no traffic passes through the IoT-based sensor. The passive IoT-based sensors monitor traffic through a switch-spanning port; thus, all traffic passes through the switch.

* Corresponding author.
E-mail address: ashankar2711@gmail.com (A. Shankar).

https://doi.org/10.1016/j.dcan.2022.05.027
Received 23 September 2020; Received in revised form 27 May 2022; Accepted 29 May 2022
Available online 16 June 2022

Intrusion Prevention Systems (IPSs), also called IDPSs, are organized security frameworks that screen the network organization, and framework exercises for pernicious action [29]. The primary elements of IPSs are to recognize malicious action, log data about the action, attempt to stop it, and report it [30]. We intended to design IPS that can run on a host machine and help prevent network intrusion attacks on the host machine [31,32]. The goals of the host-based IPS system are to design a lightweight intrusion prevention software for an Ubuntu Linux OS- based system with a management console, introduce network-monitoring capabilities in the software, design statistical and signature-based security features that allow early detection of network attacks [33,34], implement response mechanisms against network attacks, and provide users with the facility to design their own network security rules and implement them via the IPS software.

This paper proposes two types of Network-based Intrusion Detection Systems (NIDSs): promiscuous and network system modes. This Intrusion Detection System (IDS) reacts just to the mark-based identified assaults, which is a significant disadvantage of this strategy, and yet, the user is expected to understand the issue. A novel string coordinating strategy algorithm, which is an enhancement of other coordinating algorithms, is proposed. The proposed string coordinating algorithm breaks a string into little arrangements of state machines. The state machines perceive the subset of the string. If there is any sign of suspicious conduct, data about the interloper is communicated to each module that holds the database for characterizing the tenets, and the mark of the interloper is compared with the predefined identified marks. This algorithm is exceptionally productive, and it is ten times faster than the other calculations and expends fewer assets. An appropriated IDS is utilized to investigate the framework as per Mrdović and Zajko's paper using numerous sensors arranged in sections to monitor the system conduct. SNORT tool was used in the investigation. The logs are created in MySQL.The conveyed IDS is overseen by the administration that is in charge of observing and designing the IDS for reassuring. This IDS provided important security against assaults. It can be built using numerous PCs for observing and safeguarding the arrangement from malicious attacks. This framework requires high memory and experienced security examiners and administration, which is troublesome.

2. Literature review

An IPS has different configurations based on the location of the installed software, the application domain and the network area of jurisdiction. Interruption counteractive action frameworks such as IPSs have two fundamental types, namely, network-organized and host-based [35,36]. The differences and similarities between these types are described here.

2.1. IoT based network-organization handling IPS

Network system-based interruption aversion frameworks are typically rack-mounted equipment or frameworks connected to information arrangement. The network is designed to send a duplicate of all the movement in the system through the IPS for the IPS to inspect it to identify potential interruptions [37,38]. There are four main types of IPSs, each with its own particular Four-Letter Acronym (FLA). They are Host-based IPDs (HIPDs), which are introduced on the host to identify attacks against host frameworks; Host-based IPSs (HIPSs), which is an interruption aversion framework, is introduced on a host and is intended to stop assaults against the host framework; Network-based IDSs (NIDSs), check a system to recognize assaults; Network-based IPSs (NIPSs) check a system to stop assaults.

A Network-based IPS (NBIPS) is used to monitor a network and protect its confidentiality, integrity, and availability. Its primary function includes protecting the network from threats, such as DoS and unauthorized usage. NBIPS monitors the network for malicious or suspicious traffic by analyzing the protocol activity. The proposed system uses signature-based detection, anomaly-based detection, and protocol state analysis detection to detect and prevent attacks on the network.

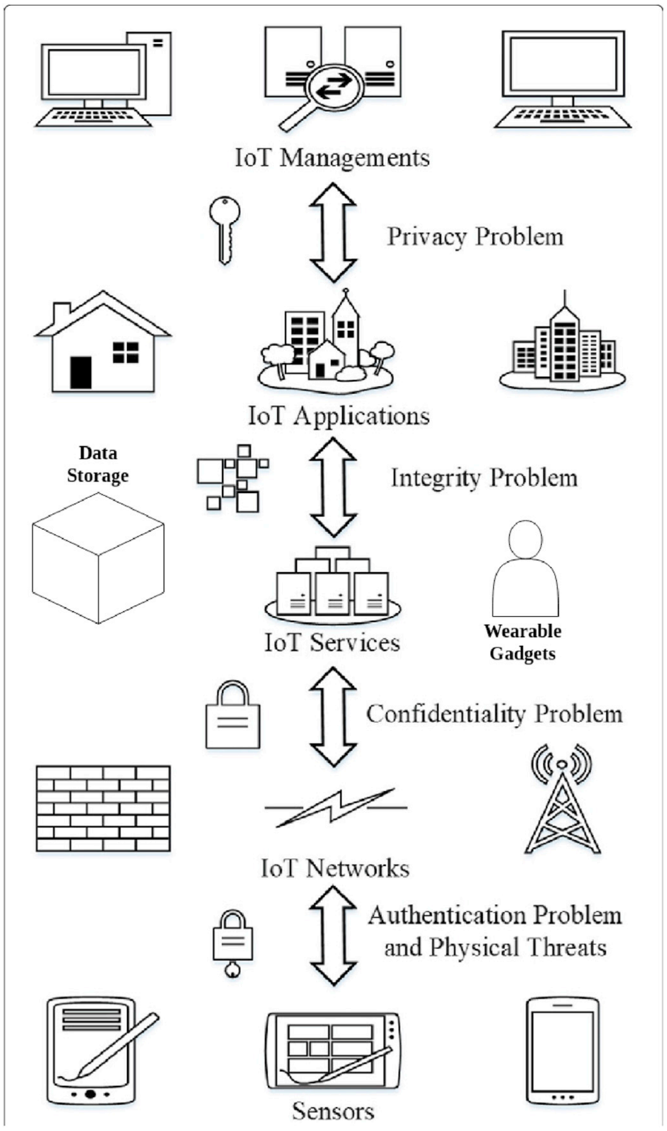


Fig. 1. Security challenges in IoT layers [44].

APTs regularly target frameworks that store, transmit, or process information [39,40]. Hence, it will be preferable to perform a host-based recognition and counteractive action [41]. However, host-based discovery involves the problem that once the aggressor possesses the capacity to trade off a framework, they can see the nearness of HIDS or HIPS on the framework [38,40,42,43]. This is similar to a thief who detects a video reconnaissance camera after breaking into a home or office (see Fig. 1). The NIDS, NIPS, or camcorder will fundamentally frighten away the interloper, yet it might compel the aggressor to change their strategies considering the goal of making their activities less recognizable [32,38, 43].

2.2. Regular identification methodologies

IPSs utilize distinctive techniques to recognize security episodes. The developers of IPSs found that no technique is viable for identifying and halting most sorts of occurrences; instead, they have settled on various well-known approaches to achieve this [45,46].

2.3. Rule-based identification

IPSs can distinguish occurrences by comparing perceptions against the already characterized episodes and known vulnerabilities. This requires identification of known and unknown dangers. A few guidelines (otherwise called marks) are attacks focusing on vulnerabilities in working frameworks and applications. For example, botnets focused on DoS attacks via Personally Identifiable Information (PII). Unusually substantial ping bundles may indicate a ping of deadly assault.

Because new types of assaults against data frameworks are constantly produced, IPSs need to update their standards frequently. The creators of IPS produce guidelines and randomly "network" of governing journalists, which are conveyed to run IPSs utilizing the Insightful Web interlopers to know how signature-based recognition functions; accordingly various methods were derived for dodging location, mostly by presenting unobtrusive variations in their assaults. Consequently, driving IPS producers typically distribute powerlessness-based tenets (rather than abuse-based marks) to recognize every single conceivable variation of an assault. They may likewise offer inconsistency-based location strategies [47].

2.4. Anomaly-based detection

IPSs can recognize episodes by looking at movement designs that an IPS considers "ordinary" with new activity examples and by choosing whether the new activity designs are indeed worthy examples. A particular preferred standpoint of abnormality-based location is recognizing episodes that may be inactivated by a standard IPS manager or signature.

2.5. Vulnerability-based principles versus adventure-based signatures

One of the primary issues with a mark-based (for instance, misuse-based) approach is the inability to distinguish zero-day assaults. Although some zero-day assaults are misusing another defenselessness, numerous objective vulnerabilities are now known. Given this, it is necessary for an IPS to have its guidelines because of real vulnerabilities instead of marks considering the known assaults. For example, consider a lock that might have a plan shortcoming that makes it powerless against picking. It would be better for an IPS to be comfortable with the bolt's powerlessness so that it will have the capacity to recognize any sort of assault upon it. Notwithstanding, if the IPS was instead arranged to identify just known bolt picking strategies (assaults), at that point, any new techniques for picking the bolt would go undetected.

2.6. SYN floods

An SYN surge is an assault on an objective framework, particularly an assault at a key outline trait of the TCP/IP organizing convention. In an SYN surge, the assailant sends many SYN bundles to an objective framework. An SYN parcel is usually a message sent from another PC that needs to build up a system associated with the objective. After getting the SYN, the objective framework answers with an SYN/ACK, and the discussion starts. Notably, the objective PC will designate assets (primarily memory) completely expecting the new association. In an SYN surge, the aggressor sends numerous SYN pardes and disregards all the SYN/ACKs. This is to surge the objective framework until it is unequipped for imparting on any open channels. An SYN surge is an exceptional type of disavowal of administration assault.

2.7. Denial of service

A foreswearing of administration (DoS) assault is an assault on an objective framework where the target of the assault is to debilitate the objective framework. A DoS assault renders the target framework unusable for good. Because an assailant would do a DoS assault, it could incorporate vengeance, desire, philosophy, or financial aspects. Conferring a DoS assault is similar to hindering the passageways to a business in order not to belittle its clients. There are two essential sorts of DoS assaults:

- Flooding: The most well-known type of DoS assault is when the aggressor sends such a high volume of messages to an objective framework that it either breakdowns or is otherwise inaccessible for good purposes.
- Breakdown: The other standard type of DoS assault is one where an exceptionally created message is sent to the objective framework; the message makes the objective framework breakdowns or crashes.

Another DoS assault is known as the Conveyed Disavowal of Administration (DDoS) assault. In a DDoS assault, the aggressor makes a wide range of frameworks surge an objective framework simultaneously. Such an assault can be brutal to deal with if there are hundreds or, on the other hand, many distinctive sources. Botnets are regularly used to confer DDoS assaults.

2.8. Centralized insurance of IoT environments

NBIPS altogether diminishes the time and costs related to anchoring a multivendor condition. The regular endeavour has five working frameworks sent and has no less than nine mission basic applications running simultaneously. Each working framework and application seller has its novel security vulnerabilities and weaknesses, leaving a regular association with many vulnerabilities at any given time. How would you oversee such vast numbers of vulnerabilities with such a significant number of sellers? You may disregard the circumstance, or you spend a considerable number of dollars consistently to stay aware of patches. Alternatively, you could use Network-based Interruption Counteractive action (NBIPS) to midway secure this horde of working frameworks, hosts, and applications.

2.9. Securing working frameworks (Work area and server)

Working frameworks like Windows, Linux, and Unix are generally powerless against assaults. A system that has been enlarged with NBIPS will channel assaults before they can contaminate further, penetrating your figuring frameworks. Assaults like Code Red (single assault vector), Nimda (various assault vectors), and Sapphire (single UDP bundle) are ceased chilly in the system.

2.10. Application security

Mission basic applications like monetary frameworks and collaborative effort frameworks are powerless against assaults. Viewpoint and Trade, Notes, Texting, Prophet database, MS SQL and IBM DB2 are secured by an all-around planned NBIPS. Cases of adventures that are doused through an NBIPS equipped system include: Determination Administration Buffer Flood, Contorted Emulate Header DoS, e-Manager Buffer Flood, Sign-On Web Administrations Security (Port 80 Applications) Web servers (for example, IIS and Apache), and Web-based applications (for example, Websphere,.Net, Prophet 9i, and BEA) are particularly helpless against assaults. These applications ordinarily incorporate complex components for parsing and dealing with self-assertive client input, leading to numerous programming mistakes. Cases of endeavours that are stifled through an NBIPS outfitted system are First Page Server Augmentations, Visual Studio Buffer Overflows, Sun Cobalt RaQ4 Server, Remote Compromise via CGI Errors, IIS Web Server, Chunked Exchange Encoding Heap Flood Apache Web Server, OpenSSL Slapper Worm, Allaire ColdFusion, and Test Content Vulnerabilities.

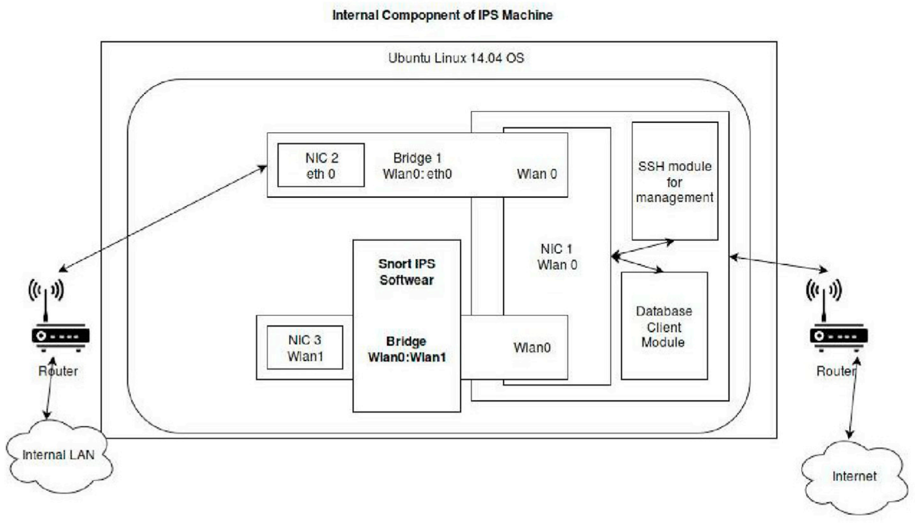


Fig. 2. Internal architecture of NBIPS system.

2.11. *Infrastructure protection*

Center network system administration foundations like the DNS and even Cisco switches can be pushed to the edge of total collapse by a keen assailant. Foundation assaults, for example, the deformed SNMP DoS activated by PROTOS (Cisco) and the Quandary TSIG buffer flood utilized by the Lion worm (DNS) are thumped around NBIPS-prepared LANs.

2.12. *Expenses of instability turn into the benefits of security*

A system enhanced with the Interruption Avoidance framework turns into a solidified shield for everything associated with it from inside and outside. System crashes, server crashes, and stolen data all affect the primary corporate concern.

Network-based interruption anticipation NBIPS might be the most intense innovation in the world for securing against traded-off work areas and servers, exploitative workers, and modern undercover work. Finally, the payback of network-based interruption counteractive action is relatively quick, and over a long haul, it might distinguish between survival and disappointment.

3. Summary

Considering these criteria, it is essential to build an IPS. The issues of data security are discussed herein, considering the security needs of an association for ensuring the safety of their basic data from assaults. Huge number of experienced examiners are required to continuously observe the framework. Development of new security methodologies includes high exertion, which is examined herein. In the multi-layer approach in IPS screens, a single host is given. This approach comprises three layers: Record Analyzer, Framework Asset Analyzer, and Association Analyzer. The benefit of this strategy is that it gives both mark-based and anomaly-based recognition and avoidance. However, it requires large memory to secure framework information and system activity. In the proposed IPS, the IDPS is separated and partitioned into two types: in-source and out-source. The critical business of Managed Security Services Providers (MSSP) is to provide security for an association against assaults. Mostly, MSSP incorporates inspecting innovation to anchor the data far better than previously.

Snort and source fire are considered the best interruption counteractive action frameworks for a multinational organization. Snort provides an office to adjust its source code with the assistance of source fire. The hindrance of snort is that it utilizes a signature-based system for recognizing the interruption. If any peculiarity conduct happens, it is not workable for snort to recognize that assault. A mobile agent depends on the proposed IDPS. The protected portable operator is dependable for observing the framework, handling the logs, distinguishing the assaults, and securing the host via constant computerized reaction. The real downside of this method is that if the mobile agent is the objective for the assailants, then it winds up monotonously for any IDPS to protect the framework being hacked. Subsequently, a different security framework is essential for securing the portable specialist. David and Paola proposed a method that demonstrates the association of use with the working framework and looks at how an IDS could be broken without being recognized by utilizing the sequence matching technique. However, this strategy is unaware of the exertion and information required for delivering such an assault. It is additionally uninformed of the expectation of the working of IDS by the assailants.

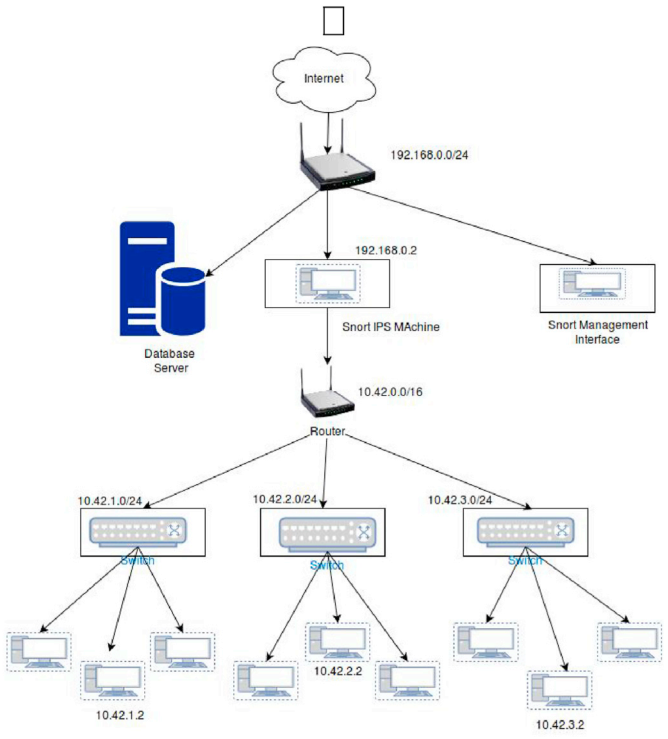


Fig. 3. External architecture of NBIPS system.

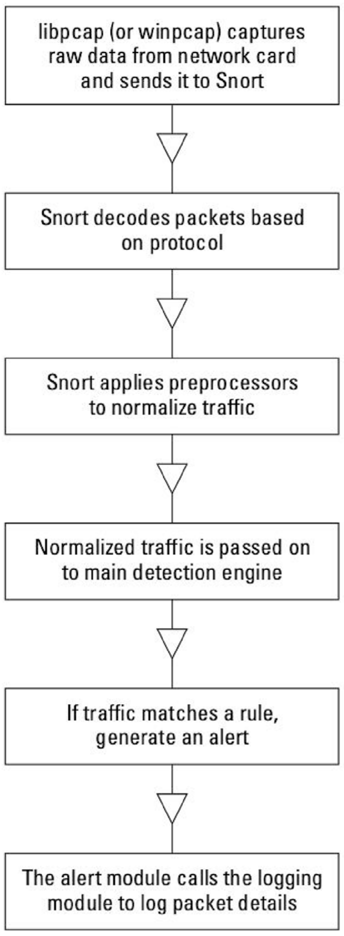

libpcap (or winpcap) captures raw data from network card and sends it to Snort
Snort decodes packets based on protocol
Snort applies preprocessors to normalize traffic
Normalized traffic is passed on to main detection engine
If traffic matches a rule, generate an alert
The alert module calls the logging module to log packet details

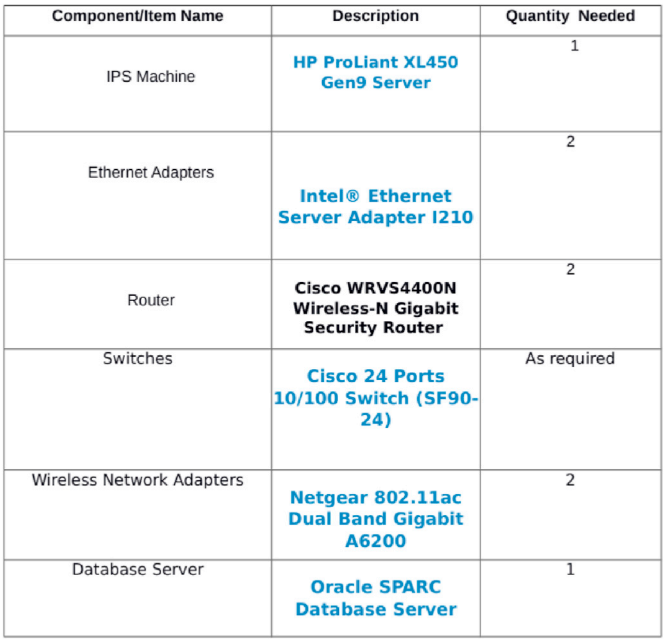
Fig. 4. Snort Internal Packet processing diagram.

4. Proposed solution

An IPS is a system security/risk aversion innovation that inspects network activity streams to identify and counteract misuses. Mostly, powerlessness misuses come as pernicious contributions to a target application or administration that assailants use to hinder and gain control of an application or machine. Following a fruitful adventure, the assailant can handicap the target application (bringing a DoS) or possibly access all rights and authorizations accessible to the traded-off application. Fig. 2 shows the interior design of the NBIPS framework.

4.1. Prevention

The IPS is usually located behind a firewall, and it gives a reciprocal layer of investigation that adversely chooses unsafe substances. Unlike its forerunner, the IDS— which is a latent framework that outputs activity and reports back on the dangers— is set to inline (for immediate communication way amongst the source and goal), currently examining and taking mechanized activities on all movement streams that enter the system. In particular, these activities include: sending a caution to the head (as would be found in an IDS), dropping the pernicious packets, blocking activity from the source address, and resetting the connection. The IPS must work effectively to abstain from corrupting network execution for inline security part. It should likewise work quickly since an attack can happen closely and consistently. The IPS should likewise recognize and react precisely to remove dangers and false positives (authentic bundles are misread as dangers). Fig. 3 shows the external architecture of the NBIPS system.



Component/Item Name	Description	Quantity Needed
IPS Machine	HP ProLiant XL450 Gen9 Server	1
Ethernet Adapters	Intel® Ethernet Server Adapter I210	2
Router	Cisco WRVS4400N Wireless-N Gigabit Security Router	2
Switches	Cisco 24 Ports 10/100 Switch (SF90-24)	As required
Wireless Network Adapters	Netgear 802.11ac Dual Band Gigabit A6200	2
Database Server	Oracle SPARC Database Server	1

Fig. 5. Hardware Requirements of implemented IPS.

4.2. Components of IPS

Inline Mode: The IPS is set to inline behind a firewall or switch with the goal that all system movement passes through it. This arrangement underpins the two IPS (blocking) and IDS (alarming) modes.

Network tap: A tap is an equipment gadget that allows information to flow through the system. A sidestep tap is commonly utilized for inline IPS arrangements for IPS gadgets that do not have a quick-open feature, or for associations where it may be desirable to keep their inline IP separate from the system for maintenance or reconfiguration. A recovery tap is often utilized for inactive IDS setups when the traverse ports on observed switch gadgets are expended (see Fig. 4).

Switch span port: This is a port on a system switch where a duplicate of all activity that courses through the switch can be observed, thereby providing an aloof IDS setup.

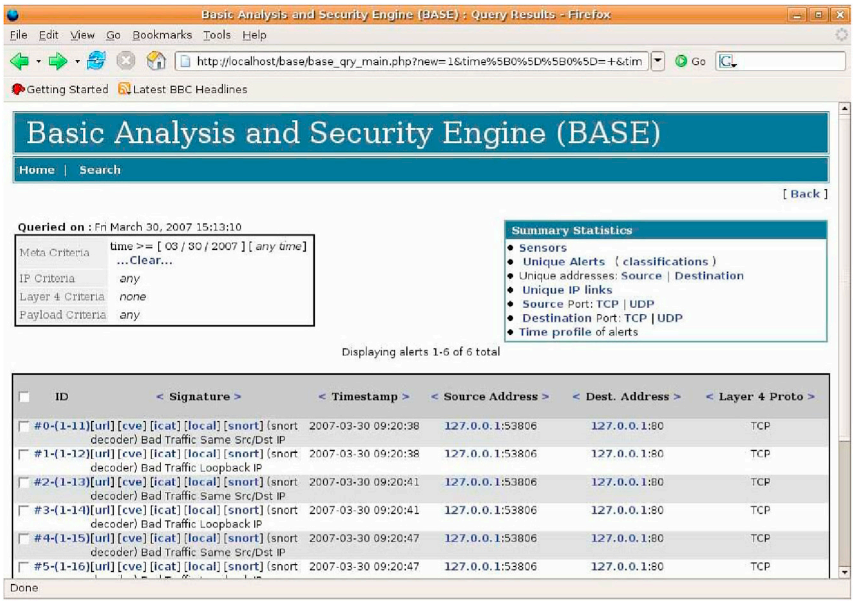
Core or data center network: More associations are broadening the assurance of their edge IPS by introducing IPS sensors (ordinarily set in the latent IDS mode) in the center or information focus. This gives an extra layer of barrier and identifies assaults hand-conveyed into the workplace on portable processing gadgets.

Extranets: Bigger associations with extranet associations with accomplice or provider systems may put an inline IPS gadget before related switches to shield against potential approaching assaults and guarantee that nearby malware does not spread to accomplice systems.

Remote access points: Contractual workers and visitors typically interface with the system through remote access focuses. As these gadgets are regularly uncontrolled by IT, IPS sensors behind remote passages to screen for potential unwanted movement.

Virtualization platforms: Although virtualization has great advantages in terms of reducing the cost, it likewise presents new dangers and vulnerabilities. A physical IPS placed before a virtualization or a virtual IPS introduced on every virtualization can help protect against concealed assaults beginning from inside or focusing on virtual machines.

Primary network segments: These may be systems containing basic frameworks (for example, servers containing money related or restorative information), where interruptions can be exceptionally genuine.


Basic Analysis and Security Engine (BASE) : Query Results - Firefox

http://localhost/base/base_qry_main.php?new=1&time%5B0%5D%5B0%5D=+&tim

Basic Analysis and Security Engine (BASE)

Home | Search

[Back]

Queried on : Fri March 30, 2007 15:13:10

Meta Criteria	time >= [03 / 30 / 2007] [*any time*] ...Clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

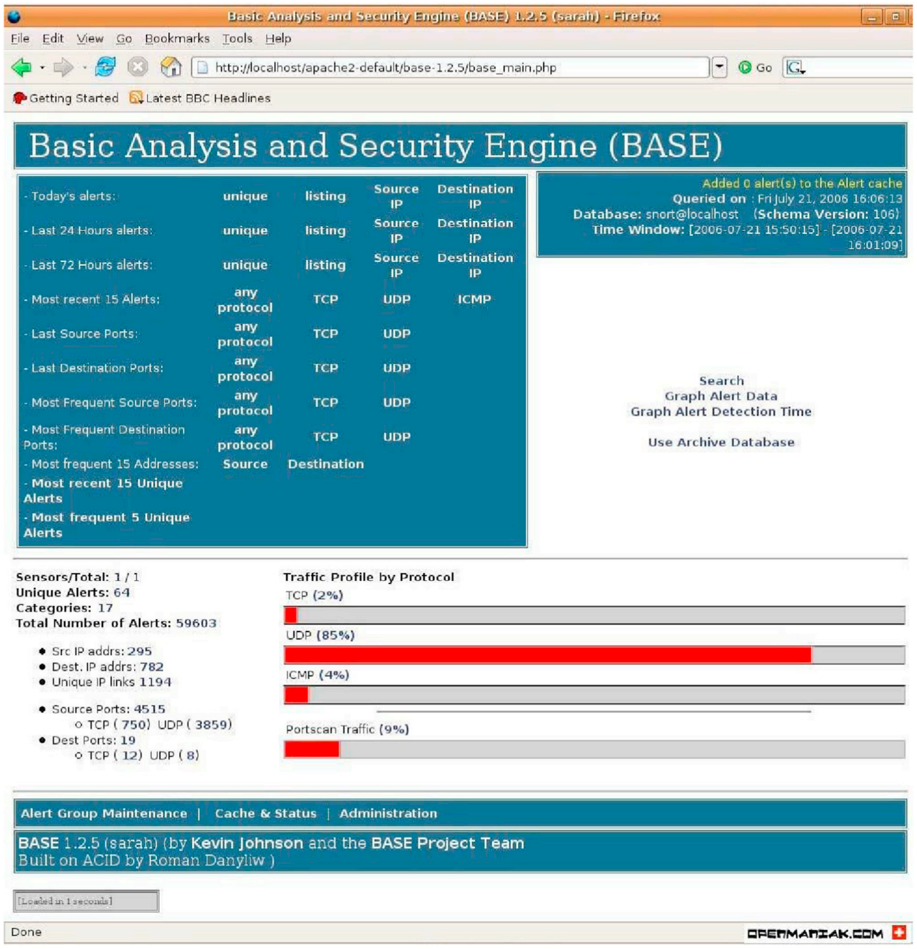
Summary Statistics
- Sensors
- Unique Alerts (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-6 of 6 total

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(1-11)	[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Same Src/Dst IP	2007-03-30 09:20:38	127.0.0.1:53806	127.0.0.1:80	TCP
#1-(1-12)	[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Loopback IP	2007-03-30 09:20:38	127.0.0.1:53806	127.0.0.1:80	TCP
#2-(1-13)	[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Same Src/Dst IP	2007-03-30 09:20:41	127.0.0.1:53806	127.0.0.1:80	TCP
#3-(1-14)	[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Loopback IP	2007-03-30 09:20:41	127.0.0.1:53806	127.0.0.1:80	TCP
#4-(1-15)	[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Same Src/Dst IP	2007-03-30 09:20:47	127.0.0.1:53806	127.0.0.1:80	TCP
#5-(1-16)	[url] [cve] [icat] [local] [snort] (snort	2007-03-30 09:20:47	127.0.0.1:53806	127.0.0.1:80	TCP

Done

Fig. 6. Basic analysis and security Engine 1.


Basic Analysis and Security Engine (BASE) 1.2.5 (sarah) - Firefox

http://localhost/apache2-default/base-1.2.5/base_main.php

Basic Analysis and Security Engine (BASE)

- Today's alerts:	unique	listing	Source IP	Destination IP
- Last 24 Hours alerts:	unique	listing	Source IP	Destination IP
- Last 72 Hours alerts:	unique	listing	Source IP	Destination IP
- Most recent 15 Alerts:	any protocol	TCP	UDP	ICMP
- Last Source Ports:	any protocol	TCP	UDP	
- Last Destination Ports:	any protocol	TCP	UDP	
- Most Frequent Source Ports:	any protocol	TCP	UDP	
- Most Frequent Destination Ports:	any protocol	TCP	UDP	
- Most frequent 15 Addresses:	Source	Destination		
- **Most recent 15 Unique Alerts**				
- **Most frequent 5 Unique Alerts**				

Added 0 alert(s) to the Alert cache
Queried on : Fri July 21, 2006 16:06:13
Database: snort@localhost (**Schema Version:** 106)
Time Window: [2006-07-21 15:50:15] - [2006-07-21 16:01:09]

Search
Graph Alert Data
Graph Alert Detection Time

Use Archive Database

Sensors/Total: 1 / 1
Unique Alerts: 64
Categories: 17
Total Number of Alerts: 59603

- Src IP addrs: 295
- Dest. IP addrs: 782
- Unique IP links 1194

- Source Ports: 4515
  - TCP (750) UDP (3859)
- Dest Ports: 19
  - TCP (12) UDP (8)

Traffic Profile by Protocol
TCP (2%)
UDP (85%)
ICMP (4%)
Portscan Traffic (9%)

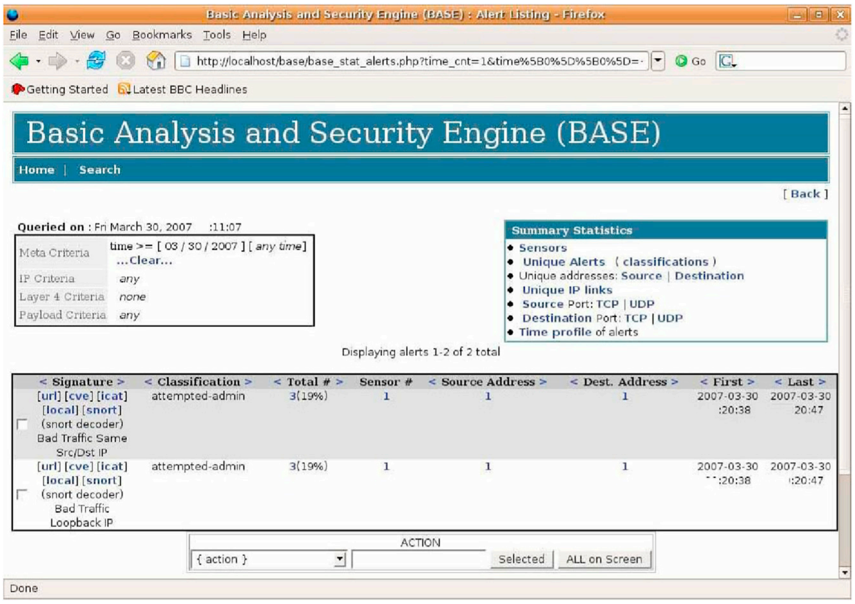
Alert Group Maintenance | Cache & Status | Administration

BASE 1.2.5 (sarah) (by **Kevin Johnson** and the **BASE Project Team**
Built on ACID by Roman Danyliw)

[Loaded in 1 seconds]

Done

Fig. 7. Basic analysis and security Engine 2.


Basic Analysis and Security Engine (BASE) : Alert Listing - Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/base/base_stat_alerts.php?time_cnt=1&time%5B0%5D%5B0%5D=
Go
Getting Started
Latest BBC Headlines

Basic Analysis and Security Engine (BASE)

Home | Search

[Back]

Queried on : Fri March 30, 2007 :11:07

Meta Criteria	time >= [03 / 30 / 2007] [*any time*] ...Clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Summary Statistics
- Sensors
- Unique Alerts (classifications)
- Unique addresses: Source | Destination
- Unique IP links
- Source Port: TCP | UDP
- Destination Port: TCP | UDP
- Time profile of alerts

Displaying alerts 1-2 of 2 total

< Signature >	< Classification >	< Total # >	Sensor #	< Source Address >	< Dest. Address >	< First >	< Last >
[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Same Src/Dst IP	attempted-admin	3(19%)	1	1	1	2007-03-30 :20:38	2007-03-30 20:47
[url] [cve] [icat] [local] [snort] (snort decoder) Bad Traffic Loopback IP	attempted-admin	3(19%)	1	1	1	2007-03-30 ˙˙:20:38	2007-03-30 :20:47

ACTION
{ action } Selected ALL on Screen

Done

Fig. 8. Basic analysis and security Engine 3.

5. System requirement

5.1. *Software required for implementation*

Software Barnyard2 version 2.1–13. Basic Analysis and Security Engine (BASE) version 1.4.5. Snort version 2.9.8.2 Pulled pork, and Operating System Linux Ubuntu 14.04 Server or Desktop Version LTS were used.

5.2. *Hardware requirements*

Fig. 5 shows the hardware required for implementing IPS.

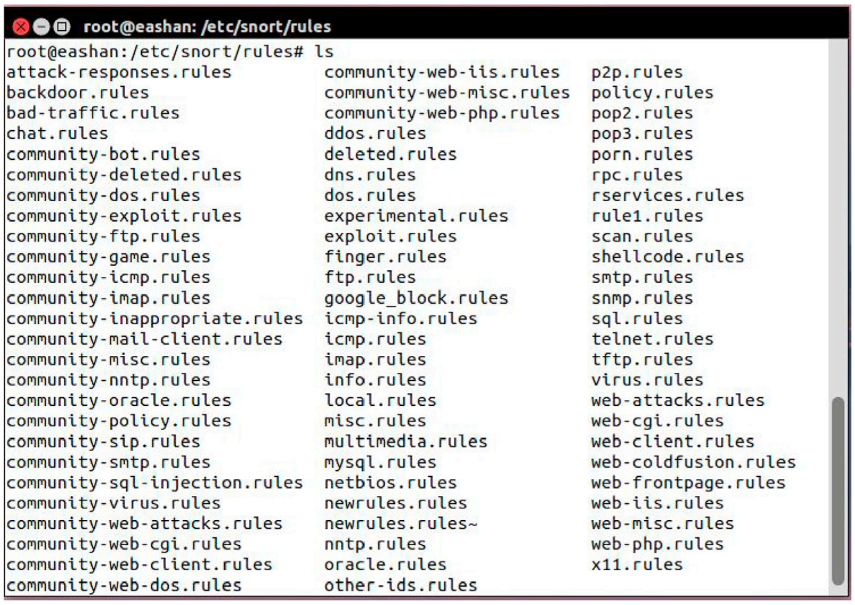
5.3. *Configuring barnyard*

Barnyard2 is an open-source translator for Snort unified2 twofold yield records.

It is essential to enable Snort to write to the disk effectively and remove the hassle of parsing parallel information into different organizations to different programs or organizational activities.

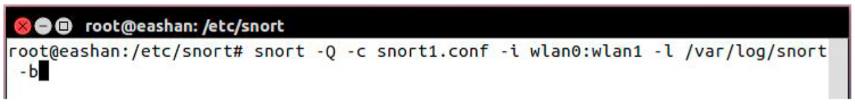
Barnyard2 has three methods of activity:

- group (or one-shot),
- persistent,
- persistent w/bookmark.



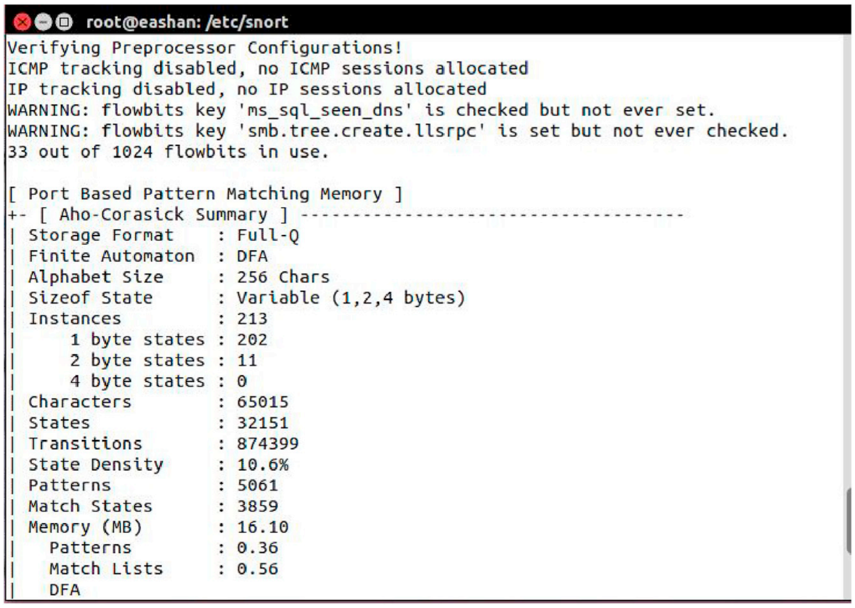
```
root@eashan:/etc/snort/rules# ls
attack-responses.rules          community-web-iis.rules     p2p.rules
backdoor.rules                  community-web-misc.rules    policy.rules
bad-traffic.rules               community-web-php.rules     pop2.rules
chat.rules                      ddos.rules                  pop3.rules
community-bot.rules             deleted.rules               porn.rules
community-deleted.rules         dns.rules                   rpc.rules
community-dos.rules             dos.rules                   rservices.rules
community-exploit.rules         experimental.rules          rule1.rules
community-ftp.rules             exploit.rules               scan.rules
community-game.rules            finger.rules                shellcode.rules
community-icmp.rules            ftp.rules                   smtp.rules
community-imap.rules            google_block.rules          snmp.rules
community-inappropriate.rules   icmp-info.rules             sql.rules
community-mail-client.rules     icmp.rules                  telnet.rules
community-misc.rules            imap.rules                  tftp.rules
community-nntp.rules            info.rules                  virus.rules
community-oracle.rules          local.rules                 web-attacks.rules
community-policy.rules          misc.rules                  web-cgi.rules
community-sip.rules             multimedia.rules            web-client.rules
community-smtp.rules            mysql.rules                 web-coldfusion.rules
community-sql-injection.rules   netbios.rules               web-frontpage.rules
community-virus.rules           newrules.rules              web-iis.rules
community-web-attacks.rules     newrules.rules~             web-misc.rules
community-web-cgi.rules         nntp.rules                  web-php.rules
community-web-client.rules      oracle.rules                x11.rules
community-web-dos.rules         other-ids.rules
```

Fig. 9. Snort rule.



```
root@eashan: /etc/snort
root@eashan:/etc/snort# snort -Q -c snort1.conf -i wlan0:wlan1 -l /var/log/snort
 -b
```

Fig. 10. Snort configuration.



```
root@eashan: /etc/snort
Verifying Preprocessor Configurations!
ICMP tracking disabled, no ICMP sessions allocated
IP tracking disabled, no IP sessions allocated
WARNING: flowbits key 'ms_sql_seen_dns' is checked but not ever set.
WARNING: flowbits key 'smb.tree.create.llsrpc' is set but not ever checked.
33 out of 1024 flowbits in use.

[ Port Based Pattern Matching Memory ]
+- [ Aho-Corasick Summary ] ------------------------------------
| Storage Format    : Full-Q
| Finite Automaton  : DFA
| Alphabet Size     : 256 Chars
| Sizeof State      : Variable (1,2,4 bytes)
| Instances         : 213
|     1 byte states : 202
|     2 byte states : 11
|     4 byte states : 0
| Characters        : 65015
| States            : 32151
| Transitions       : 874399
| State Density     : 10.6%
| Patterns          : 5061
| Match States      : 3859
| Memory (MB)       : 16.10
|   Patterns        : 0.36
|   Match Lists     : 0.56
|   DFA
```

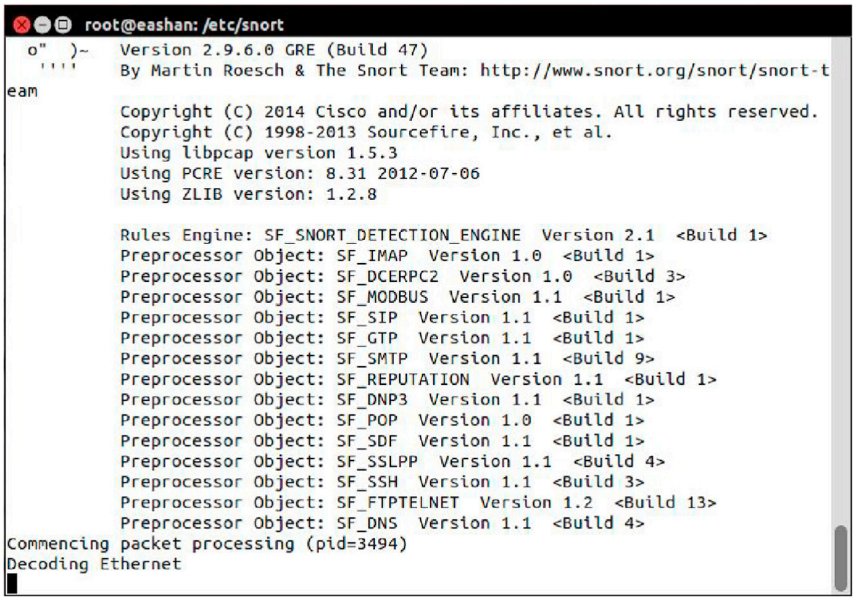
Fig. 11. Snort configuration.

Barnyard is a fundamental instrument for parsing snort, which brings together similar binary documents, handling, and on-sending to an assortment of yield plugins. However, it has not been updated in more than four years, and it will not be kept up by the first designers.

With the new form of the combined configuration (i.e., unified2), we must connect this lacuna. Barnyard is a yield framework for Snort. Snort makes a unique binary document format called brought unified. Barnyard peruses this document and then resends the information to a database at the backend. Unlike the database yield module, Barnyard knows about the inability to send the alarm to the database, and it quits sending cautions. It is also mindful when the database can acknowledge associations again and will begin resending the alarms.

5.4. BASE

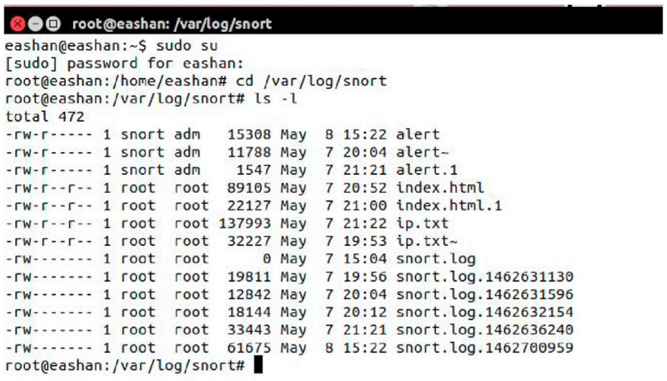
BASE is an essential investigation and security tool (refer to Fig. 6). It depends on the code from the Analysis Console for Intrusion Databases (ACID) project. This application gives a web front-end to inquiry and



```
root@eashan: /etc/snort
  o"  )~   Version 2.9.6.0 GRE (Build 47)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.1  <Build 1>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
Commencing packet processing (pid=3494)
Decoding Ethernet
```

Fig. 12. Summary of analysis.



```
eashan@eashan:~$ sudo su
[sudo] password for eashan:
root@eashan:/home/eashan# cd /var/log/snort
root@eashan:/var/log/snort# ls -l
total 472
-rw-r----- 1 snort adm   15308 May  8 15:22 alert
-rw-r----- 1 snort adm   11788 May  7 20:04 alert~
-rw-r----- 1 snort adm    1547 May  7 21:21 alert.1
-rw-r--r-- 1 root  root  89105 May  7 20:52 index.html
-rw-r--r-- 1 root  root  22127 May  7 21:00 index.html.1
-rw-r--r-- 1 root  root 137993 May  7 21:22 ip.txt
-rw-r--r-- 1 root  root  32227 May  7 19:53 ip.txt~
-rw------- 1 root  root      0 May  7 15:04 snort.log
-rw------- 1 root  root  19811 May  7 19:56 snort.log.1462631130
-rw------- 1 root  root  12842 May  7 20:04 snort.log.1462631596
-rw------- 1 root  root  18144 May  7 20:12 snort.log.1462632154
-rw------- 1 root  root  33443 May  7 21:21 snort.log.1462636240
-rw------- 1 root  root  61675 May  8 15:22 snort.log.1462700959
root@eashan:/var/log/snort# █
```

Fig. 13. Logs record.

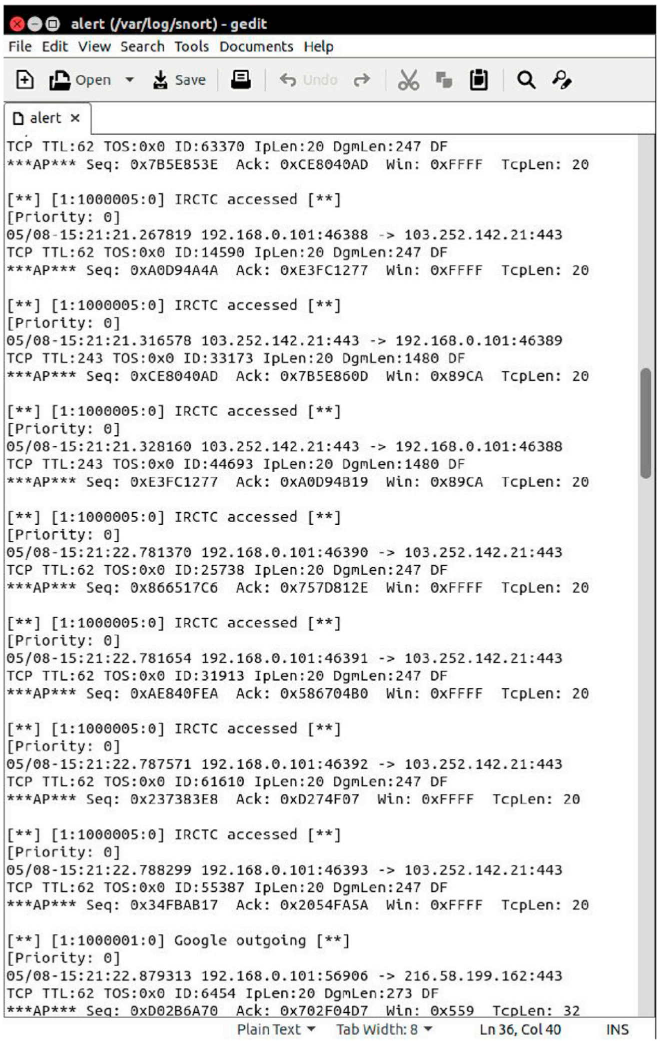
examines the alarms originating from a Snort IDS framework. Individually, SNORT continues running in independent mode as a parcel sniffer and lumberjack[logger]. Upon combination with different applications and some configurations, a Snort framework becomes considerably more helpful as a NIDS. The bolstering software segments we introduce are as follows: Barnyard2 is a dedicated spooler for Snort's unified2 two-fold yield arrangement. Parcel handling is exceptionally dependent on the asset and hence diminishes the load on the Snort process: we have Snort saves suspicious parcels to an index in a local parallel format without handling the bundles. Barnyard2 then non concurrently handles those bundles and saves them in a MySQL database. PulledPork is the Perl content that consequently downloads the most recent Snort rulesets. Because the risk scene advances continually, new rulesets are needed by Snort to distinguish the most recent sorts of suspicious activity (rulesets are like antivirus marks). The BASE gives a web front-end to question and examine the alarms originating from a Snort framework (see Figs. 7 and 8). First, we have to introduce every one of the essentials from the Ubuntu stores: sudo apt-get install -y build-essential libpcap-dev libpcre3-dev libdumbnet-dev bison flex zlib1g-dev Breakdown of the bundles we are introducing:

- Build-fundamental: gives the construct apparatuses (GCC and so forth) to compile programming.
- Bison and flex: they have the required parser (DAQ is introduced later beneath).
- Libpcap-dev: library for capturing the network traffic required by Snort.
- Libpcre3-dev: library of capacities to help standard articulations required by Snort.
- Libdumbnet-dev: the libdnet library gives a disentangled and convenient interface to a few low-level systems administration schedules. Numerous aides for introducing Snort introduce this library from the source, though it is not important.
- Zlib1g-dev: a compression library needed by Snort.
- Snort utilizes the information securing library (DAQ) to digest calls to bundle capture libraries. DAQ is downloaded and introduced from the SNORT site.

6. Implementation

6.1. Running SNORT in inline mode

1. List of rules configured to trigger is shown in Fig. 9.
2. The SNORT command to begin inline processing of packets via a network bridge made between wlan0 and wlan1 network interfaces (refer to Fig. 10).
3. The command in Step 2, when executed, gives output similar to that shown in Fig. 11.



alert (/var/log/snort) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
alert
Plain Text
Tab Width: 8
Ln 36, Col 40
INS

```
TCP TTL:62 TOS:0x0 ID:63370 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x7B5E853E  Ack: 0xCE8040AD  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.267819 192.168.0.101:46388 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:14590 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0xA0D94A4A  Ack: 0xE3FC1277  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.316578 103.252.142.21:443 -> 192.168.0.101:46389
TCP TTL:243 TOS:0x0 ID:33173 IpLen:20 DgmLen:1480 DF
***AP*** Seq: 0xCE8040AD  Ack: 0x7B5E860D  Win: 0x89CA  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.328160 103.252.142.21:443 -> 192.168.0.101:46388
TCP TTL:243 TOS:0x0 ID:44693 IpLen:20 DgmLen:1480 DF
***AP*** Seq: 0xE3FC1277  Ack: 0xA0D94B19  Win: 0x89CA  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.781370 192.168.0.101:46390 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:25738 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x866517C6  Ack: 0x757D812E  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.781654 192.168.0.101:46391 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:31913 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0xAE840FEA  Ack: 0x586704B0  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.787571 192.168.0.101:46392 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:61610 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x237383E8  Ack: 0xD274F07  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.788299 192.168.0.101:46393 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:55387 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x34FBAB17  Ack: 0x2054FA5A  Win: 0xFFFF  TcpLen: 20

[**] [1:1000001:0] Google outgoing [**]
[Priority: 0]
05/08-15:21:22.879313 192.168.0.101:56906 -> 216.58.199.162:443
TCP TTL:62 TOS:0x0 ID:6454 IpLen:20 DgmLen:273 DF
***AP*** Seq: 0xD02B6A70  Ack: 0x702F04D7  Win: 0x559  TcpLen: 32
```

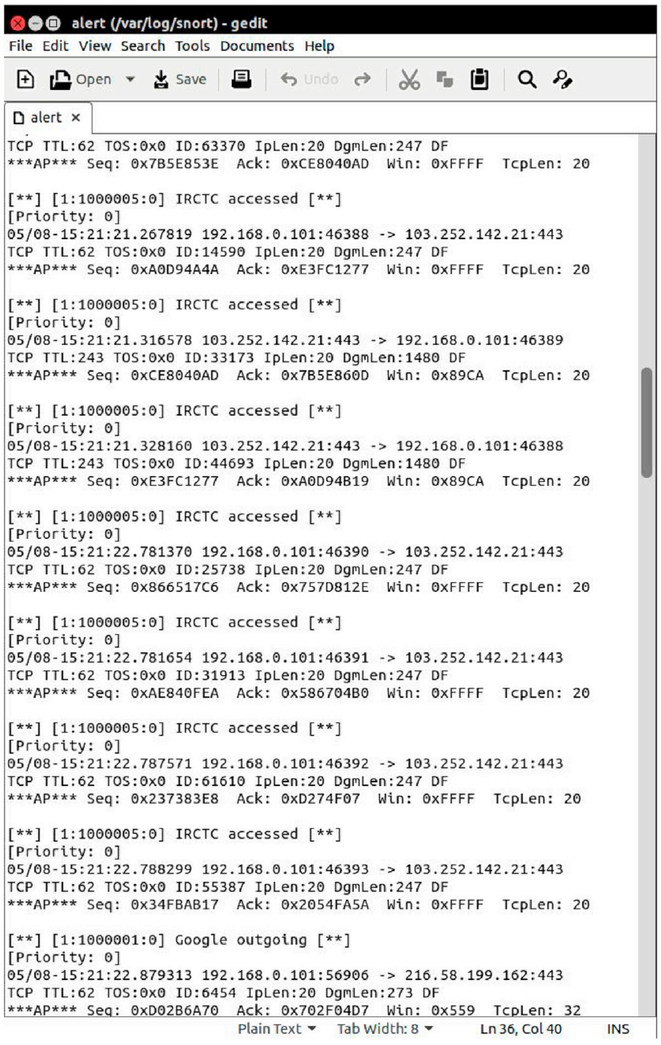
Fig. 14. Alert generate.

4. The inline packet processing has ended. The summary of the analysis of the session is provided by SNORT (shown in Fig. 12).
5. SNORT logs its alerts and packets in the/var/log/snort directory, as stated in Step 2 (see Fig. 13).
6. Alerts fired by SNORT (refer to Fig. 14).
7. Packets logged by SNORT are shown in Fig. 15.
8. The additional rules are defined by us to trigger the passive alert rules (see Fig. 16).

6.2. Pullpork for rule list

Features and capabilities of using Pullpork are as follows:

- Computerized downloading, parsing, state change and lead adjustment for the greater part of your grunt rule sets.
- Checksum confirmation for all genuine manage downloads
- Programmed age of the refreshed sid-msg. map record
- Capacity to incorporate the local rules in the sid-msg. map record
- Capacity to pull rules tarballs from custom URLs
- Overall shared object bolster
- Overall IP notoriety rundown bolster
- Ability to download numerous divergent rule sets immediately
- Keeps up the precise change log
- Ability for HUP forms after standards download and processing
- Helps in the tuning of rule sets


alert (/var/log/snort) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
alert
Plain Text
Tab Width: 8
Ln 36, Col 40
INS

```
TCP TTL:62 TOS:0x0 ID:63370 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x7B5E853E  Ack: 0xCE8040AD  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.267819 192.168.0.101:46388 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:14590 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0xA0D94A4A  Ack: 0xE3FC1277  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.316578 103.252.142.21:443 -> 192.168.0.101:46389
TCP TTL:243 TOS:0x0 ID:33173 IpLen:20 DgmLen:1480 DF
***AP*** Seq: 0xCE8040AD  Ack: 0x7B5E860D  Win: 0x89CA  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:21.328160 103.252.142.21:443 -> 192.168.0.101:46388
TCP TTL:243 TOS:0x0 ID:44693 IpLen:20 DgmLen:1480 DF
***AP*** Seq: 0xE3FC1277  Ack: 0xA0D94B19  Win: 0x89CA  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.781370 192.168.0.101:46390 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:25738 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x866517C6  Ack: 0x757D812E  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.781654 192.168.0.101:46391 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:31913 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0xAE840FEA  Ack: 0x586704B0  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.787571 192.168.0.101:46392 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:61610 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x237383E8  Ack: 0xD274F07  Win: 0xFFFF  TcpLen: 20

[**] [1:1000005:0] IRCTC accessed [**]
[Priority: 0]
05/08-15:21:22.788299 192.168.0.101:46393 -> 103.252.142.21:443
TCP TTL:62 TOS:0x0 ID:55387 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x34FBAB17  Ack: 0x2054FA5A  Win: 0xFFFF  TcpLen: 20

[**] [1:1000001:0] Google outgoing [**]
[Priority: 0]
05/08-15:21:22.879313 192.168.0.101:56906 -> 216.58.199.162:443
TCP TTL:62 TOS:0x0 ID:6454 IpLen:20 DgmLen:273 DF
***AP*** Seq: 0xD02B6A70  Ack: 0x702F04D7  Win: 0x559  TcpLen: 32
```

Fig. 15. Packets logging.

- Verbose yield with the goal such that the user precisely understands the situation.
- Negligible Perl module conditions
- A sweet smokey enhance all through the pork

In addition, each document contains rules particular to a specific class. The DNS rules document contains all guidelines identified for assaults on DNS servers, the telnet. Rules record contains all tenets identified with assaults on the telnet port, etc.

6.2.1. *Communication cost*

In this subsection, we make assumptions for computing the transmission cost of the suggested system, which is mentioned in Ref. [48]. Table 1 lists the assumed cost of components in the suggested protocol. For an elliptic curve $E_p(a, b)$, all parameters (p, a and b) are assumed as 160-bit each. Therefore, an ECC point X, $Y = (x_P, y_P) \in E_p(a, b)$ takes (80 + 80) = 160 bits. For the suggested protocol, the communication parameters are X_i, Y_i, and ID_i, X_j, Y_j, and ID_j. Thus the communication cost is (X_i, Y_i, ID_i) + (X_j, Y_j, ID_j) and (800 + 800) = 384 bits, which is less than that for the existing systems, as mentioned in Table 2.

6.2.2. *Storage cost*

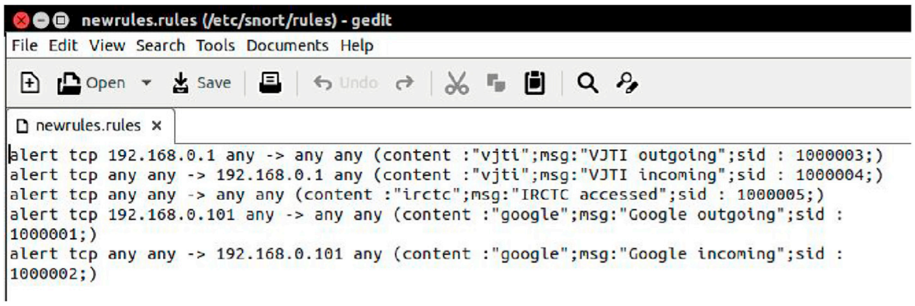
In this subsection, we make assumptions for computing the storage cost of the suggested system, as mentioned in Ref. [48]. Table 3 presents the assumed cost of components in the proposed protocol. In the proposed system, a smart meter stores two challenges (i.e., P_x, P_y) and consumes 20 + 20 = 40 bytes in its storage mentioned. This is less than that for the existing systems mentioned in Table 3.

7. Conclusion and future work

Network systems-based interruption avoidance frameworks predict an immediate future where turmoil, nervousness, cost and sweat are supplanted with sureness, efficiency and gainfulness. We trust it is officeholder on all associations, private and open, to convey NBIPS for the accompanying reasons: It will enhance corporate efficiency and gainfulness. It will secure delicate data frames from being stolen. It will shield key frameworks from rapidly occurring worldwide digital assaults safeguarding ways of life and lifestyles. It will restrict copyright encroachment obligations. NBIPS can stop any attack based on malicious traffic sent over a network, provided it has a known attack signature.

Our future work will delve into IPSs, which are in great demand world wide due to the increasing complexity of network-based attacks and different number of network parameters used for staging new and inventive attacks. Each new solution implemented by an organization introduces its own set of network and system vulnerabilities, which are put to the test by hackers globally. The challenges for the conventional IPSs will be updated with all kinds of software and network vulnerabilities well in advance, since by the time the vulnerability is released to the public, the damage may already have been done to the network of an organization.

The prevention system must have an analytical module to assess the robustness of its own network by deploying attacks on its network with the help of artificial intelligence-based methods. Problems related to the increasing speed of the Internet also arise because the IPSs may not be able to perform with scalability with network speeds of 1Gbps and more. A single server can handle only a limited number of vulnerabilities while monitoring a network flow of 1Gbps and providing latency-free network access. IPSs need to reinvent their software to include virtualization techniques and high-performance computing to achieve scalability in their operation. Because the increase in vulnerabilities increases the


newrules.rules (/etc/snort/rules) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
newrules.rules

```
alert tcp 192.168.0.1 any -> any any (content :"vjti";msg:"VJTI outgoing";sid : 1000003;)
alert tcp any any -> 192.168.0.1 any (content :"vjti";msg:"VJTI incoming";sid : 1000004;)
alert tcp any any -> any any (content :"irctc";msg:"IRCTC accessed";sid : 1000005;)
alert tcp 192.168.0.101 any -> any any (content :"google";msg:"Google outgoing";sid :
1000001;)
alert tcp any any -> 192.168.0.101 any (content :"google";msg:"Google incoming";sid :
1000002;)
```

Fig. 16. Rules used to trigger passive alert.

Table 1
Components cost Table.

Components	Size in bits
ID_i, ID_j	32
(X_i, X_j)	80
(Y_i, Y_j)	80
(P_x, P_y)	160
$2(X_i + Y_i + ID_i)$	$(160 + 160{+}64) = 384$
$(P_x + P_y)$	$(160 + 160) = 320$

Table 2
Communication cost comparison.

Systems	Number of messages	Number of bits
Chim. et al. [49]	**3**	**4448**
Fouda. et al. [50]	**3**	**3744**
Suggested	**2**	**384**

Table 3
Storage cost comparison.

Systems	cost
Chim. et al. [49]	**3232**
Fouda. et al. [50]	**320**
Suggested	**320**

number of test parameters for each network packet, high computing power and forwarding capability are critical. Hence, we conclude that there is much scope for further development and research of IPSs in the future.

Ethical approval and consent to participate

Not applicable.

Consent for publication

All the authors of this paper have shown their participation voluntarily.

Availability of supporting data

The data will be provided based on data requests by the evaluation team.

Funding

This research received no specific grant from any funding agency in public, commercial or not-for-profit sectors.

Authors' contributions

The authors have proposed an intrusion prevention system that inspects network activity streams to identify and counteract misuses. The IPS regularly sits specifically behind the firewall, and it gives a reciprocal layer of investigation that adversely chooses unsafe substances. Together it is described as the NBIPS framework.

Declaration of competing interest

The authors of this research article declare no conflict of interest in preparing this research article.

References

[1] P.J. Taylor, T. Dargahi, A. Dehghantanha, R.M. Parizi, K.-K.R. Choo, A systematic literature review of blockchain cyber security, Digital Communications and Networks 6 (2) (2020) 147–156.
[2] A. Singh, R.M. Parizi, Q. Zhang, K.-K.R. Choo, A. Dehghantanha, Blockchain smart contracts formalization: approaches and challenges to address vulnerabilities, Comput. Secur. 88 (2020), 101654.
[3] W. Kabaciński, M. Abdulsahib, Wide-sense nonblocking converting-space-converting switching node architecture under xsvarswitch control algorithm, IEEE/ACM Trans. Netw..
[4] M. Roesch, Snort: lightweight intrusion detection for networks, Lisa 99 (1999) 229–238.
[5] D. Wagner, P. Soto, Mimicry attacks on host-based intrusion detection systems, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM, 2002, pp. 255–264.
[6] P. Nerurkar, Review of data storage by fusion drive in mac, Int. J. Adv. Res. Comput. Sci. 4 (3).
[7] A. Saeed, A. Ahmadinia, A. Javed, H. Larijani, Intelligent intrusion detection in low-power iots, ACM Trans. Internet Technol. 16 (4) (2016) 1–25.
[8] Y. Fu, Z. Yan, J. Cao, O. Koné, X. Cao, An Automata Based Intrusion Detection Method for Internet of Things, Mobile Information Systems, 2017.
[9] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for iot security based on learning techniques, IEEE Commun. Surv. Tutorials 21 (3) (2019) 2671–2701.
[10] M. Abdulsahib, W. Kabaciński, Wide-sense nonblocking and blocking converting-space-converting switching node architecture under xsvarslot algorithm, Opt. Switch. Netw. 37 (2020), 100557.
[11] B.A.A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: past, present, and future of programmable networks, IEEE Commun. Surv. Tutorials 16 (3) (2014) 1617–1634.
[12] Y.-D. Lin, Y.-K. Lai, Q.T. Bui, Y.-C. Lai, Refsm: reverse engineering from protocol packet traces to test generation by extended finite state machines, J. Netw. Comput. Appl. (2020), 102819.
[13] Y. Lin, Editorial: second quarter 2020 ieee communications surveys and tutorials, IEEE Commun. Surv. Tutorials 22 (2) (2020) 790–795.
[14] H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K. R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, IEEE Transact. Emerging Topics Comput..
[15] E. Benkhelifa, T. Welsh, W. Hamouda, A critical review of practices and challenges in intrusion detection systems for iot: toward universal and resilient systems, IEEE Commun. Surv. Tutorials 20 (4) (2018) 3496–3509.
[16] S. Chen, A. Fu, J. Shen, S. Yu, H. Wang, H. Sun, Rnn-dp: a new differential privacy scheme base on recurrent neural network for dynamic trajectory privacy protection, J. Netw. Comput. Appl. 168 (2020), 102736.
[17] L. Cui, Y. Qu, L. Gao, G. Xie, S. Yu, Detecting false data attacks using machine learning techniques in smart grid: a survey, J. Netw. Comput. Appl. (2020), 102808.
[18] F. Al-Turjman, Intelligence and security in big 5g-oriented iont: an overview, Future Generat. Comput. Syst. 102 (2020) 357–368.
[19] M.A. Rahman, N. Zaman, A.T. Asyhari, F. Al-Turjman, M.Z.A. Bhuiyan, M. Zolkipli, Data-driven dynamic clustering framework for mitigating the adverse economic impact of covid-19 lockdown practices, Sustain. Cities Soc. 62 (2020), 102372.
[20] P. Nerurkar, A. Pavate, Study of angular js: a client side javascript framework for single page applications, Int. J. Contempor. Res. Comp. Sci. Technol. 1 (4) (2015) 92–96.
[21] C. Kreibich, Network Intrusion Detection: Evasion, Traffic Normalization, and End-To-End Protocol Semantics.
[22] J. Zhang, Y. Zhang, K. Lu, J. Wang, K. Wu, X. Jia, B. Liu, Detecting and identifying optical signal attacks on autonomous driving systems, IEEE Internet Things J..
[23] Z. Huang, W. Wu, F. Shan, Y. Bian, K. Lu, Z. Li, J. Wang, J. Wang, Couas: enable cooperation for unmanned aerial systems, ACM Trans. Sens. Netw. 16 (3) (2020) 1–19.
[24] M. Dabbour, I. Alsmadi, E. Alsukhni, Efficient assessment and evaluation for websites vulnerabilities using snort, Int. J. Secur. Appl. 7 (1) (2013) 7–16.
[25] S. Han, H. Lee, Performance analysis of coverage probability according to transmission range of devices, J. Korea Inst. Inform. Commun. Eng. 20 (10) (2016) 1881–1886.
[26] P. Nerurkar, S. Bhirud, Modeling influence on a social network using interaction characteristics, Int. J. Comput. Mathemat. Sci. 6 (8) (2017) 152–160.
[27] P. Nerurkar, A. Shirke, M. Chandane, S. Bhirud, A novel heuristic for evolutionary clustering, Procedia Comput. Sci. 125 (2018) 780–789.
[28] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey, Proc. IEEE 103 (1) (2014) 14–76.
[29] K. Benzekki, A. El Fergougui, A. Elbelrhiti Elalaoui, Software-Defined Networking (sdn): A Survey, Security and Communication Networks 9, 2016, pp. 5803–5833 (18).
[30] Z. Zhou, C. Zhongwen, Z. Tiecheng, G. Xiaohui, The study on network intrusion detection system of snort, in: Networking and Digital Society (ICNDS), 2010 2nd International Conference on, 2, IEEE, 2010, pp. 194–196.
[31] H.G. Kayacik, A.N. Zincir-Heywood, A case study of three open source security management tools, in: Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on, IEEE, 2003, pp. 101–104.
[32] V. Paxson, J. Rothfuss, B. Tierney, Bro Quick Start Guide, Retrieved April 22 (2004) 2010.

[33] B.B. Zarpelão, R.S. Miani, C.T. Kawakani, S.C. de Alvarenga, A survey of intrusion detection in internet of things, J. Netw. Comput. Appl. 84 (2017) 25–37.

[34] S. Raza, L. Wallgren, T. Voigt, Svelte, Real-time intrusion detection in the internet of things, Ad Hoc Netw. 11 (8) (2013) 2661–2674.

[35] M. Karakus, A. Durresi, Quality of service (qos) in software defined networking (sdn): a survey, J. Netw. Comput. Appl. 80 (2017) 200–218.

[36] I.T. Haque, N. Abu-Ghazaleh, Wireless software defined networking: a survey and taxonomy, IEEE Commun. Surv. Tutorials 18 (4) (2016) 2713–2737.

[37] N.A. Jagadeesan, B. Krishnamachari, Software-defined networking paradigms in wireless networks: a survey, ACM Comput. Surv. 47 (2) (2014) 1–11.

[38] W. Braun, M. Menth, Software-defined networking using open flow: protocols, applications and architectural design choices, Future Internet 6 (2) (2014) 302–336.

[39] D. Evans, The Internet of Things: How the Next Evolution of the Internet Is Changing Everything, White Paper, Cisco Internet Business Solutions Group, Apr. 2011, 2011.

[40] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, Nox: towards an operating system for networks, Comput. Commun. Rev. 38 (3) (2008) 105–110.

[41] P. Nerurkar, A. Shirke, M. Chandane, S. Bhirud, Empirical analysis of data clustering algorithms, Procedia Comput. Sci. 125 (2018) 770–779.

[42] S. Morzhov, I. Alekseev, M. Nikitinskiy, Firewall application for floodlight sdn controller, in: 2016 International Siberian Conference on Control and Communications (SIBCON), IEEE, 2016, pp. 1–5.

[43] J. Medved, R. Varga, A. Tkacik, K. Gray, Open day light: towards a model-driven sdn controller architecture, in: Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, IEEE, 2014, pp. 1–6.

[44] M.F. Elrawy, A.I. Awad, H.F. Hamed, Intrusion detection systems for iot-based smart environments: a survey, J. Cloud Comput. 7 (1) (2018) 21.

[45] P. Nerurkar, M. Chandane, S. Bhirud, A comparative analysis of community detection algorithms on social networks, in: Computational Intelligence: Theories, Applications and Future Directions, I, Springer, Singapore, 2019, pp. 287–298.

[46] P. Nerurkar, M. Chandane, S. Bhirud, Community detection using node attributes: a non-negative matrix factorization approach, in: Computational Intelligence: Theories, Applications and Future Directions, I, Springer, Singapore, 2019, pp. 275–285.

[47] P. Nerurkar, A. Pavate, M. Shah, S. Jacob, Analysis of probabilistic models for influence ranking in social networks, in: Computing, Communication and Signal Processing, Springer, Singapore, 2019, pp. 215–223.

[48] L. Zhang, S. Tang, H. Luo, Elliptic curve cryptography-based authentication with identity protection for smart grids, PLoS One 11 (3) (2016), e0151253.

[49] T.W. Chim, S.-M. Yiu, L.C. Hui, V.O. Li, Pass: privacy-preserving authentication scheme for smart grid network, in: 2011 IEEE International Conference on Smart Grid Communications (Smart Grid Comm), IEEE, 2011, pp. 196–201.

[50] M.M. Fouda, Z.M. Fadlullah, N. Kato, R. Lu, X.S. Shen, A lightweight message authentication scheme for smart grid communications, IEEE Trans. Smart Grid 2 (4) (2011) 675–685.