DESIGNING RULES TO IMPLEMENT RECONNAISSANCE AND UNAUTHORIZED ACCESS ATTACKS FOR INTRUSION DETECTION SYSTEM

Subhi A. Mohammed
College of Information Engineering, Al-Nahrain University, Baghdad, Iraq
subhiaswad@coie-nahrain.edu.iq

Received:14/7/2019, Accepted:21/9/2019

***Abstract*- Network attacks are classified according to their objective into three types: Denial of Services (DOS), reconnaissance and unauthorized access. A base signature Intrusion Detection System (IDS) which gives an alarm when the monitor network traffic meets a previously specified set of criteria of attack traffic. This paper will focus on design, compose, and process IDS rules, and then to decide whether that packet is intrusive or not, by examining the signatures of the attacks in both incoming packets headers and payload to networks. Packet sniffer is performs capturing, decoding and reassembling of the network packet traffic, then passes it to the programmed rules. Linux backtrack tools was used to implement an IDS scenario for two types of attacks (Reconnaissance and Unauthorized access). The results show that IDS rules are able to detect large numbers of various attacks.**

keywords: **Intrusion detection system (IDS), IDS rules, Reconnaissance attacks, Denial of services (DOS) attacks, Packet sniffer.**

I. Introduction

Network intruders can overcome the authentication mechanisms designed to protect systems. Intruders are becoming skilled at finding system imperfections, and also use intervention patterns that are difficult to recognize and trace. They are useful not only in detecting successful violations of security, but also in monitoring attempts to breaking security [1]. The networks are becoming ever more vulnerable to a wider range of security threats. The primary network requirement is to have multiple internet access points, both public and private networks; thus, securing these networks has become extreme importance. The goals of network security are as follows:

- Prevention: refers to preventing computer or information violations from occurring.
- Detection: refers to identifying events when they occur. Incident detection includes identifying the assets being attacked, how it happened, and who did it.
- Response: means to develop strategies and techniques to deal with an attack.

The primary function of IDS is to detect intrusive and attacks directed against computer networks. IDS must be programmed with the various condition commands called IDS rules, so that when they are uniquely combined together they can identify an attack. IDS rules check collective criteria that characterize an attack; these criteria are called signatures of attacks. The IDS rules allows to uniquely identify a specific attack based on a signature test. These signatures can exist in both packet header and packet payload of networks traffics [2].

II. IDS Related Works

There are many IDS implementation in literature. **In 2010**, proposed a structure of cooperative IDS in cloud computing to reduce the impact of Distributed Denial- Of-Service (DDoS). Each IDS within the system has a cooperative agent that calculates and determines whether or not to accept the sent alerts from other IDSs. The suggested cooperative IDS system only slightly increases the computational effort and prevents system failure by a single point of attack [3]. **In 2012** , hybrid ANN for anomaly detection and Back Propagation Neural Network (BPNN), Self- Organizing Maps (SOM), and Simulated Annealing Neural Network (SA). The proposed system compared the different ANN techniques in terms of training time, number of the epochs required, detection rate, and learning approach [4]. **In 2015**, presents the mechanism to improve the efficiency of the IDS using streaming data mining technique. They apply four selected stream data classification algorithms and compare their results [5].

In 2016, presents the performance of neural network for various values of number of clusters. The optimization of output is done using Particle Swarm Optimization (PSO) by appropriate selecting the input parameters. An algorithm based on the PSO and neural network for analyzing program behaviour in IDS is evaluated. Preliminary experiments show effectively detect intrusive attacks and achieves a low false positive rate [6]. **In 2018**, present a distributed machine learning based IDS for cloud environments. The IDS inserted in the cloud side by side with the edge network components of the cloud provider. The IDS constitutes of 5 principal modules. The network traffic module capture the incoming network traffic to the cloud, then preprocessed and passed to a first anomaly detection step using a naive bayes model. Next, the suspected traffic are synchronized to central server. Then, an ensemble learning classifier based on random forest is used to classify the network traffic data and detect the types of each attack. The proposed IDS is implemented on the Google cloud platform. The system achieved an average accuracy of 97%, an average false positive rate of 0.21% and an average running time of 6.23s [7]. **In 2019**, apply artificial immune principles to IDS in wireless sensor networks, such as negative selection algorithms and dendritic cell algorithms. A negative selection algorithm analyzes the distribution of self- set in the real-valued space then divides the real- valued space, and several subspaces are obtained. Selves are filled into different subspaces. The randomly generated candidate detector only needs to be tolerated with selves in the subspace where the detector is located, not all the selves. Theoretical analysis and experimental results show better time efficiency and quality of detectors, saves sensor node resources and reduces the energy consumption, and is an effective algorithm for wireless sensor network intrusion detection [8].

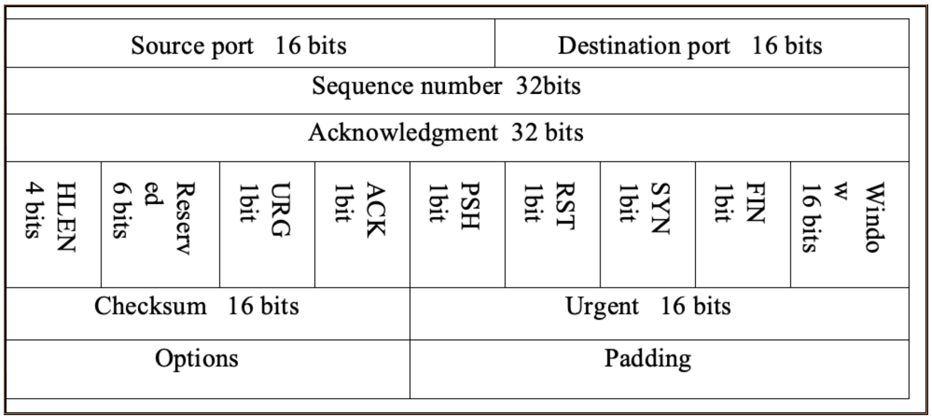
III. Computer Network Protocols

TA network can be a peer-to-peer network connecting a small number of users, a LAN, a MAN or WAN. The most common example of networks is the internet. The *Open Systems Interconnection (OSI)* model is divides network communications into seven layers: At each layer, protocols perform services unique to that layer [8].

When a host transmits data across a network to another device, the data goes through transmission media: It is wrapped with protocol information at each layer of the OSI model. To communicate and exchange information, each layer uses *Protocol Data Units (PDUs)*. These hold the control information attached to the data at each layer of the model. The protocol defines the message format and the rules for exchanging the messages. The protocol features are; Initialization, framing, synchronization, flow control, error control [9].

The primary function of IP is to facilities routing and implement network layer addressing. The global IP address space is a 32-bit number and divided into five classes. Each IP address has two parts a *network ID* and a *host ID*. The IP header refers to the control information that is placed before the actual data is sent [10].

The Transmission Control Protocol (TCP) is a connection-oriented protocol. TCP header supports error detection and correction as well as packet sequencing. The TCP header consists of (12) fields as illustrated in Fig. 1



<table>
<tr><td colspan="5">Source port 16 bits</td><td colspan="4">Destination port 16 bits</td></tr>
<tr><td colspan="9">Sequence number 32bits</td></tr>
<tr><td colspan="9">Acknowledgment 32 bits</td></tr>
<tr><td>HLEN
4 bits</td><td>Reserv
ed
6 bits</td><td>URG
1bit</td><td>ACK
1bit</td><td>PSH
1bit</td><td>RST
1bit</td><td>SYN
1bit</td><td>FIN
1bit</td><td>Windo
w
16 bits</td></tr>
<tr><td colspan="4">Checksum 16 bits</td><td colspan="5">Urgent 16 bits</td></tr>
<tr><td colspan="4">Options</td><td colspan="5">Padding</td></tr>
</table>

Figure 1: TCP header

IV. Attacks Detection in Intrusion Detection System

It is the ability of a system to detect threats against the network. The criteria used to detect if any sort of attacks, have occurred is called a trigger; triggers are of two types: (Anomaly detection and Signature based intrusion detection).

1) *Anomaly Detection*

   IDS tries to determine if a deviation, of any sort, from an established normal behavior profile has occurred that can be marked as an intrusion. Typically, each profile is made up of a number of statistical measures on system activities, for example, the frequency of system commands and the CPU usage during a user login session. Deviations from a given profile can be calculated as the weighted sum of deviations from the constituent statistical measures. The anomaly detection systems can sense unknown intrusions since they don't need prior knowledge of specific intrusions [11].The disadvantages of anomaly detection are: IDS needs an appropriate period of learning to determine normal network behavior, user profiles should be up to date since user procedures change and the alarms are difficult to recognize.

2) *Signature Based Violation Recognition*

   Signature- based violation recognition generates an alarm if the network transportation being surveyed equals previously determined criteria that indicate the traffic is being attacked. A hacker may modify the attack slightly in an attempt to overcome regular aggression sign. If the signatures are well planned and strong, it will be difficult for a hacker to hide an attack. A signature- based ID methodologies:

- Pattern Matching: The IDS looks for a fixed sequence of information within each packet.
- Stateful Pattern Matching: Many attackers try to confuse pattern identical by issuing the attacks over several packets or by disarrange packets sequence.
- Protocol Analysis: The IDS makes sure that the data is transmitted using a specific protocol and abides by the procedures of that protocol. This validity of the traffic and that it is not an attack considered to overcome security systems. Such detection capabilities requires that the IDS must have knowledge of the common protocols.

V. Types of Intrusion Detection System

IDS can be classified by its location into : Host Based Intrusion Detection System (HIDS) and Network Based Intrusion Detection System (NIDS) [12]. In HIDS, the IDS application (called an *agent*) runs in a network. It inspects a lot of parts in the system, containing privileged access tries, fault messages, and local occurrence records.

The advantages of host-based intrusion detection system are to detect fragment reassembly, Time To Live (TTL) attacks and attacks hidden by encryption from network-based IDS. The system requires one agent per host to protect.

In NIDS, the intrusion recognition comprises inserting a dedicated IDS on a network part that is responsible of monitoring traffic through this segment without being detected. IDS can be inserted in main parts during the entire network.

There are two primary methods of data collection in a network:

- Port reflecting: when duplicates of the received and outgoing packets are sent from one port of a network switch to another where they are analyzed.
- Network taps: they copy incoming and leaving packets and resend them out on the network.

Another type of collecting data in networks is called promiscuous mode, normally computer in network reads and responds only to traffic sent directly to its MAC address in promiscuous mode, the system reads all traffic and sends it to the IDS processing [13]. The most effective intrusion prevention strategy is to implement both host-based and network-based IDS.

VI. IDS Structure and Organization View

The IDS consists of two parts (sensor and operator platform) as shown in Fig. 2.

1) *Sensor*

A sensor is a component in the IDS that collects data from the source and passes it to the set of programmed rules to identify if any signature of attack is matched. The sensor is based on using (snort) software tool works under Linux platform. The components of a sensor are:

A. Packet Capture/ Decoder Engine

Packets are acquired from the network and passed through the decoding engine, which are then decoded for higher-level protocols such as TCP and UDP ports.

B. Preprocessors

Preprocessors are used to modify or arrange data packets before the IDS rules operation. Also are used to perform fragmentation/defragmentation packet for a large data chunk. The IDS must perform reassembly of the packets before applying the rules to find the signature. To detect the signature correctly all packet segments must be combined.

C. IDS Rules and Signatures

The IDS rules can be defined as a set of condition command that determines if that the captured packets is intrusive or not. The rules check the contents in the packet header and payload. The signature of packet header of TCP protocol is represented by source/destination port and TCP flags. The signature of IP packets is represented by source/destination address and other options of IP header.

The network analyzer (Wiresharek software tool) is used to analyze the network traffics. IDS rules can be applied on different parts of packet; the IP header, the transport layer header, the application layer header, and Packet payload.

D. Recording and Warning System

Conditional on what the instructions detect in a packet, it may be used to record the action or make an aware.

2) *Operator Platform*

It is management software used to logs and display alarms generated by the sensor. it consists of two parts:

- Database: used to store logs and alerts generated from logging and alerts system.
- Display engine or Web application: used to display database log information about attacks such as date of attack, types of attacks and messages to describe them. Fig. 3 shown the relation between sensor and operator platform, and Fig. 4 shown the IDS process flow chart.

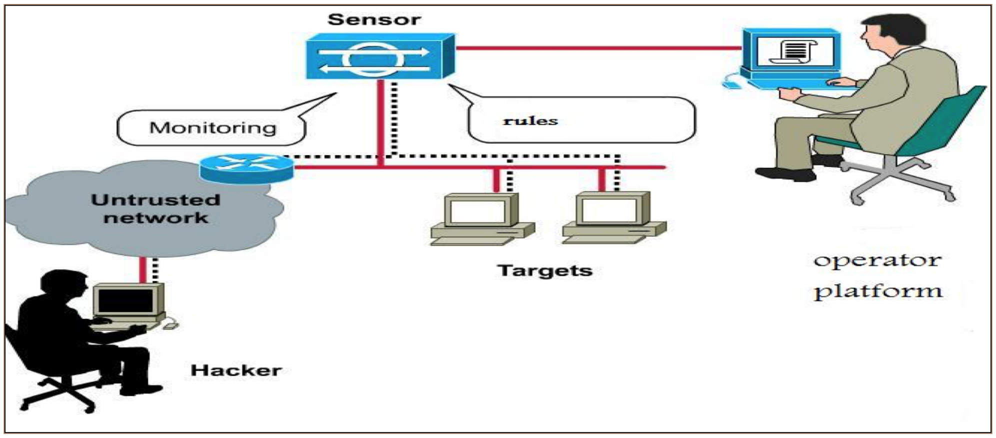


Figure 2: Intrusion detection system structure

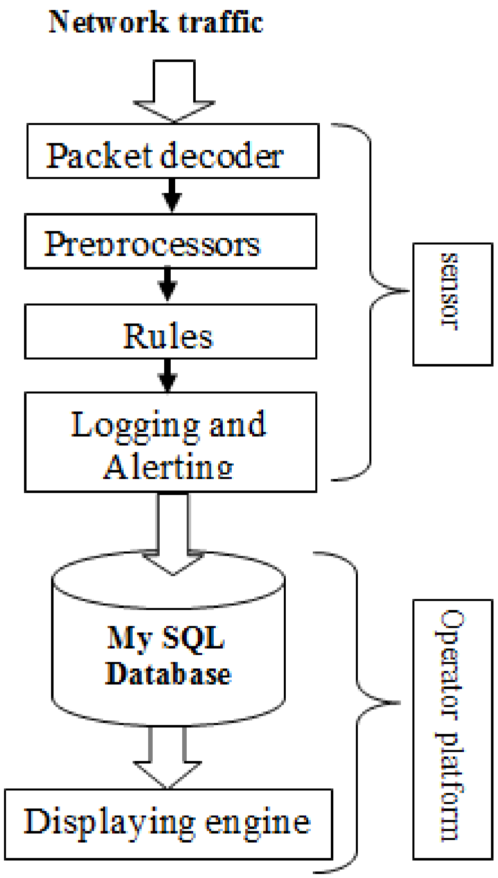


Figure 3: Relation between sensor and operator platform

VII. IDS Design

The following section describe IDS elements, their relationships, platform and configuration. It also explains how to design sensor rules, database system, web browser and display engine.

1) *Network Identification and System File Configuration*

   The IDS design includes network design, suggestion of IP addressing class, definition of system files configuration and their path. The path of IDS rules must be identified so that the sensor can pass the captured packets to this path in order to compare the packets with rules and signatures. The following next step is to describe how to build a relationship between the sensor and the database and how to join the database with web browser to display alarms.

2) *IDS Database Design*

   The database design with respect to IDS as shown in Fig. 5 includes the following activities:

   - Create MySQL database under Linux platform.
   - Define security privileges for database access.

- Link and create a relationship between the sensor and the database.
- Define a local or remote database address and file configuration.

3) *Web Display Engine*

The web application used to display an alarm when attacks are detected. The Web design use the following elements and aspects:

- Web application called (BASE) application using PHP script to display database alarms records.
- Display source/destination IP addresses of attacks of alarm.
- Using Apache Web server to create local host.
- Using ADODB data base abstraction library for PHP and python language.

4) *Design IDS Rules* The IDS rules are designed according to the pattern of attacks signatures. The rules are frequently updated so that IDS can detect new types of attacks. The rules were divided into two sections; Rule header and Rule option.

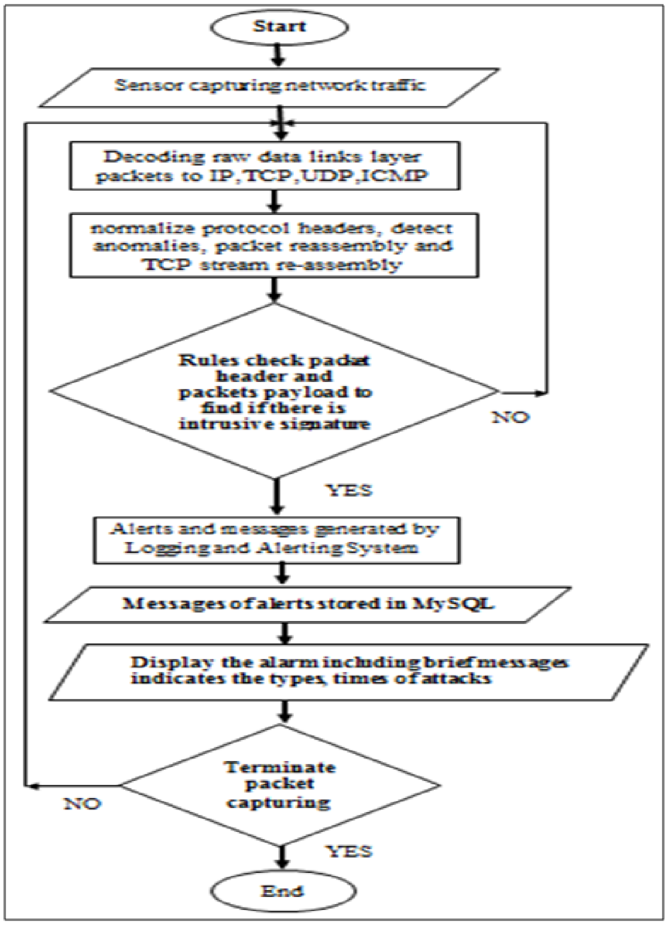

Start
Sensor capturing network traffic
Decoding raw data links layer packets to IP,TCP,UDP,ICMP
normalize protocol headers, detect anomalies, packet reassembly and TCP stream re-assembly
Rules check packet header and packets payload to find if there is intrusive signature
NO
YES
Alerts and messages generated by Logging and Alerting System
Messages of alerts stored in MySQL
Display the alarm including brief messages indicates the types, times of attacks
Terminate packet capturing
NO
YES
End

Figure 4: IDS process flow chart

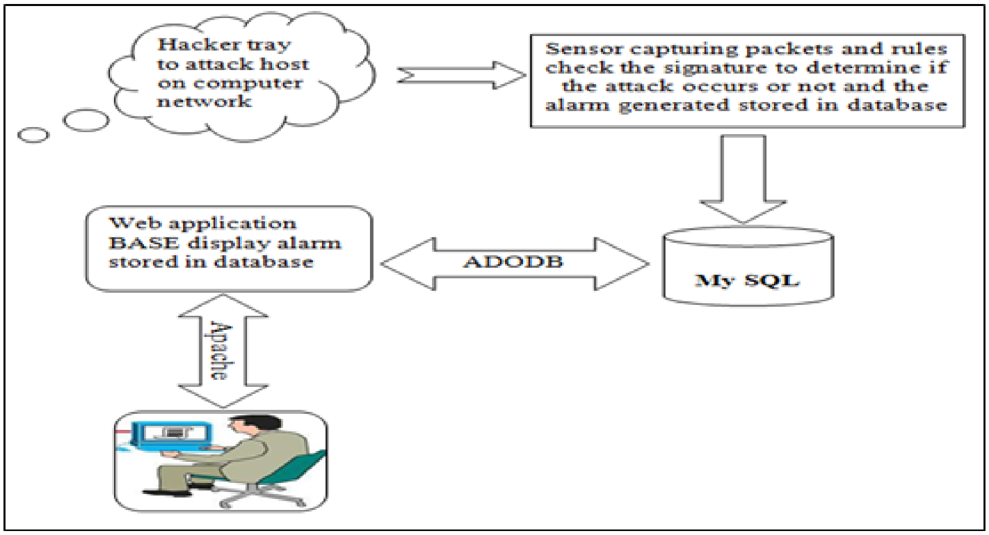

Hacker tray to attack host on computer network
Sensor capturing packets and rules check the signature to determine if the attack occurs or not and the alarm generated stored in database
Web application BASE display alarm stored in database
ADODB
My SQL
Apache

Figure 5: IDS system implementation

A. *Rule Header*

The rule header is divided into four parts:

- Rule Action: Defines what action to execute when the sensor finds a packet that match specific rule criteria. The available actions are: **alert, log, pass, dynamic, and activate.**
- Protocol: The protocol used during the generation of the packet captured.
- Source and Destination: It refers to the IP address and the port number of the source/destination of the attack. It can be represented as: **any any, EXTERNAL- NET, HOME- NET**, and the port number.

B. *Rule Option*

It is the part of rules that check the signature of packets because the signature can be present in packet header and payload. The rule option is divided into two types: the first deals with signature of packets header, and the second deals with packet payload. Another three types of option are: (flow control, MSG, and threshold).

a) *Command or Keyword Dealing with Packet Payload*

- **content:** used to search for specific content in the packet.
- **nocase:**used to deactivate case sensitivity in **(content)** keyword.
- textbfoffset:used to search in **(content)** at a specific offset from the start of the packet.
- **depth:** is used with the **(content)** keyword to specify an upper limit for the pattern matching.
- **dsize:** used to test packet payload size.

b) *Command or keyword Dealing with Packet Header*

- **Flags:** used to tests TCP flags in the packet. There are (6) flags in TCP header; (F, S, R, P, A, and U).

- **Keywords to check IP header:** set of keywords used to test specific field value of the IP header; (**ttl, tos, id, ipoption , fragbits , and same ip).**
- **keywords to check ICMP header:** set of keywords used to test specific field value of the ICMP header; $(itype, icode, icmp_id, icmp, seq, andfic)$**.**

c) *Flow Control Option Keywords*

Define the packet's direction in reference to client-server communication. The rule detects TCP packets sent from the client to the server in a TCP stream transmitting toward the server with a confirmed attack string overflow.

d) *MSG Keywords:* this rule options is used to add a text string to the log file to describe type of alerts.

e) *Threshold Keyword:* This feature is used to reduce the number of logged alerts for similar attacks.

5) *Network Analyzer* Software tool network analyzer (wiresharek) has the capability to display captured packets in readable formats and show the packet header or packets payload. The information extracted from wiresharek can be used in rule designing [14].

6) *Design and Formulation Attacks* Backtrack software tool is used to design and implement different attacks scenes. Backtrack provides scripts and tools for hacking, penetration network and for network security testing. It allow user to include customizable scripts additional tools and configurable kernel in personalized distribution focuses on penetration testers [15].

7) *Design Reconnaissance Attacks* Reconnaissance is a preliminary activity where the hacker tries to get information regarding the target network, it is a form of preparation that precedes launching the definite violence. It involves inspection the network from the private or external without being authorized to do so. Scanning level, is an early attempt to collect network related information and system characteristics.

Reconnaissance involves checking the network to gather information about the system resources. Hackers may use dumpster diving to obtain information through the discarded material of people or organizations. Active reconnaissance on the other hand "probes" the network to gather information about the operating systems in use, available services, open ports, routers, and hosts. Reconnaissance attacks may not cause any specific damage, but it is similar to a thief observing a region, surveillance for times of idleness, and sometimes trying holes and entries for access. Reconnaissance attacks are a severe risk to an association since they may give possible aggressors the material required to accomplish access or attacks [16].

C. Network Scanning Types

Scanning tools can either be used by safety supervisors to find defect in the network; or by hackers to find those same imperfection. It is used to collect information about system resources rapidly and professionally, offering it to the administrator or hacker in a format relevant to their goals. Some tools include infiltration features, which decrease the probabilities of detecting scans or probes by the target network's security defense systems.

The following are different types of network scanning which can be performed:

a) **TCP Xmas Tree Scan:** the hacker sends packets with flags (FIN, PUSH, URG) are active to the target system. For the closed ports an RST flags should be issued by the target system.
b) **TCP Null Scan:** the hacker sends packet header with no flag in TCP header is active. Target system should be issued RST for all closed ports.
c) **TCP FIN Scan:** the hacker sends packet with FIN flags is active.
d) **TCP SYN Scan:** known as half open scanning because there is no full TCP connection made. The hacker only sends a packet with a SYN flag to the target port. If a SYN/ACK is received from the target port, it is mean that the port is open. If he receives an RST/ACK, it usually means the port is closed.
e) **UDP Scan:** the hacker sends a UDP packet to the target port. If it responds with an "ICMP port unreachable" message, the port is closed. Conversely, if the hacker doesn't receive a message, the port is open.

D. Scanning and Identifying Network Resource

The main objectives of network scanning are:

- Detecting live system in a network.
- Discovering which ports are active/ running.
- Discover what operating system is running on the target system.
- Discovering the services running on the target system.
- Discovering both IP & MAC address of the target system.

Network mapper **(Nmap)** software tool is used to identify network host, platform, and to count undefended ports on likely target. It is used to identify applications running and to determine the general security situation of a network [17]. Two types of network resource scanning:

- Port Scanning: A series of messages sent to hack into a computer to learn about the computer network services, each service is associated with a "well-known" port number. Port scanning is the process of identifying open and available TCP/IP ports on a system.
- Network Scanning: A technique that recognizes dynamic hosts, either to violence them or as a security mission.

8) *Unauthorized Access Attacks* Unauthorized access attacks are meant give an aggressor access to target systems without authorization. Access attacks generally take advantage of weaknesses in a target system by using known exploits (such as a hacking tool or script) against the target system.

E. Access Attacks Types

Access attacks can be classified as providing system admission via the following tools:

a) *Unauthorized Data Management:* This discusses to the unauthorized reading, writing, copying, moving, or deleting of information that normally isn't accessible to an intruder.
b) textitSystem Access: This refers to the intruder gaining system access without previous knowledge of the system or possessing an account in it. Access to the system is generally expanded by making use of known application faults

that would give fractional or full access to a system. System access may also be gotten through poor structure or via back doors installed by an intruder during an earlier system settlement.

c) *Privilege Growth:* This states to the capability of intruders to escalate their privileges from limited access to limited or full access to the system. Privilege growth is frequently used to give an intruder the chance to install a back door giving them future access, install other tools to aid future attempts to hack deeper into the network, and also delete all traces of intrusion on the attacked system by acts such as the deletion of record archives or occurrence logs.

F. Design Password cracker Access attack

A password cracker is any software tool that can break passwords or otherwise disable password protection. The aim of password cracker is to obtain root/admin and the password of target system, and after gaining the root access the attacker escalates to admin privileges.

1) **Password Cracker Methodology** Secret code are stowed and transferred in a scrambled form called a **hash**. When a user step inside the password to login to a system, a hash is created and matched to the stored **hash**. If the both **hashes** equal, the user is given access [18].The cracker methodology consist of creating password wordlist file, encryption, applying rules, and comparing to the target password. A password cracker tool uses different methods to find out what the user has set as a password. Password crackers operate on the theory that eventually, given enough time, combinations, and permutations the password can be cracked. **(Hydra)** is a software tool that is used to implement password cracking attack.

G. Password Cracker Tool

Hydra utilizes "brute force attacks" to test passwords on one or multiple isolated system running a range of different applications. It was planned as a utility to show cracking weak passwords.

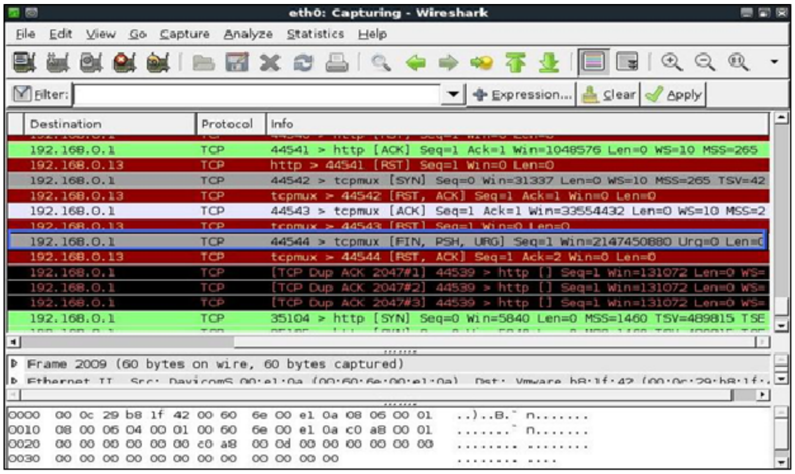
VIII. IDS Implementation Under Experimental Attack

1) *Implementation of Reconnaissance Attacks* **Zenmap** software tool is used for reconnaissance scan, and **Xmass** tree is one of these types of sca-ns. The attacker sends packets where the flags (FIN, URG, and PSH) are active. **Zenmap** is provides the ability to define the IP address of the target. Fig. 6display the result of **Zenmap** which contain the following information:

- Target system use Linux Ubuntu as an operating system.
- Port (80) which is associated with http service is open.
- MAC address of target system is **00:60:6E:00:E1:0A.**
- Service running in target system is Apache and using PHP
- textitDetection of Reconnaissance Attack Reconnaissance attacks were detected using the following IDS rule: alert tcp any any -> **$ HOME-NET any (msg:"Nmap attack is detected"; flow:stateless; flags:FPU)** The rule is checking TCP header, and if the flags (FIN, PSH, and URG) are active, this mean that TCP **Xmass** scan has

occurred, then an alert is generated. **(Wiresharek)** software tool is used to analyze network traffic generated by **Zenmap**, as shown in Fig. 7 The procedure steps for attack detection is shows Fig. 11 which contains the following activity:

- **First step:** Capturing and decoding network traffics.
- **Second step:** The IDS rules check the packet header searching for any signature that may belongs to the attacks.
- **Third step:** When the IDS rule is matched, an alarm is generated and stored in a database. MySQL is used to record system alarms in (log and alerts) database system. The Web application is used to display the contents of the database log records (attacks, hacker, network and packet) information as shown in Fig. 9.

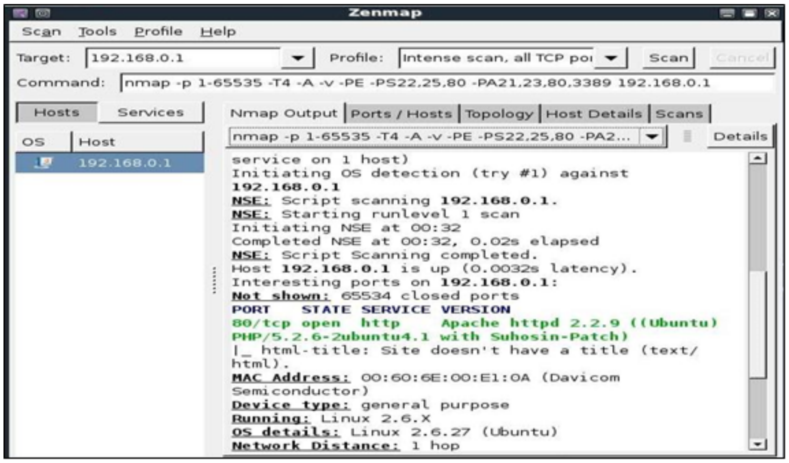


```
Destination       Protocol  Info
192.168.0.1       TCP       44541 > http [ACK] Seq=1 Ack=1 Win=1048576 Len=0 WS=10 MSS=265
192.168.0.13      TCP       http > 44541 [RST] Seq=1 Win=0 Len=0
192.168.0.1       TCP       44542 > tcpmux [SYN] Seq=0 Win=31337 Len=0 WS=10 MSS=265 TSV=42
192.168.0.13      TCP       tcpmux > 44542 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
192.168.0.1       TCP       44543 > tcpmux [ACK] Seq=1 Ack=1 Win=33554432 Len=0 WS=10 MSS=2
192.168.0.13      TCP       tcpmux > 44543 [RST] Seq=1 Win=0 Len=0
192.168.0.1       TCP       44544 > tcpmux [FIN, PSH, URG] Seq=1 Win=2147450880 Urg=0 Len=(
192.168.0.13      TCP       tcpmux > 44544 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0
192.168.0.1       TCP       [TCP Dup ACK 2047#1] 44539 > http [] Seq=1 Win=131072 Len=0 WS=
192.168.0.1       TCP       [TCP Dup ACK 2047#2] 44539 > http [] Seq=1 Win=131072 Len=0 WS=
192.168.0.1       TCP       [TCP Dup ACK 2047#3] 44539 > http [] Seq=1 Win=131072 Len=0 WS=
192.168.0.1       TCP       35104 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=489815 TSE

▷ Frame 2009 (60 bytes on wire, 60 bytes captured)

0000  00 0c 29 b8 1f 42 00 60  6e 00 e1 0a 08 06 00 01   ..)..B.` n.......
0010  08 00 06 04 00 01 00 60  6e 00 e1 0a c0 a8 00 01   .......` n.......
0020  00 00 00 00 00 00 c0 a8  00 0d 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

Figure 6: Zenmap analysis results



Target: 192.168.0.1

Profile: Intense scan, all TCP poı

Command: nmap -p 1-65535 -T4 -A -v -PE -PS22,25,80 -PA21,23,80,3389 192.168.0.1

192.168.0.1

```
nmap -p 1-65535 -T4 -A -v -PE -PS22,25,80 -PA2...
service on 1 host)
Initiating OS detection (try #1) against
192.168.0.1
NSE: Script scanning 192.168.0.1.
NSE: Starting runlevel 1 scan
Initiating NSE at 00:32
Completed NSE at 00:32, 0.02s elapsed
NSE: Script Scanning completed.
Host 192.168.0.1 is up (0.0032s latency).
Interesting ports on 192.168.0.1:
Not shown: 65534 closed ports
PORT    STATE SERVICE VERSION
80/tcp open  http     Apache httpd 2.2.9 ((Ubuntu)
PHP/5.2.6-2ubuntu4.1 with Suhosin-Patch)
|_ html-title: Site doesn't have a title (text/
html).
MAC Address: 00:60:6E:00:E1:0A (Davicom
Semiconductor)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.27 (Ubuntu)
Network Distance: 1 hop
```

Figure 7: Network analyzer showing packet header for zenmap

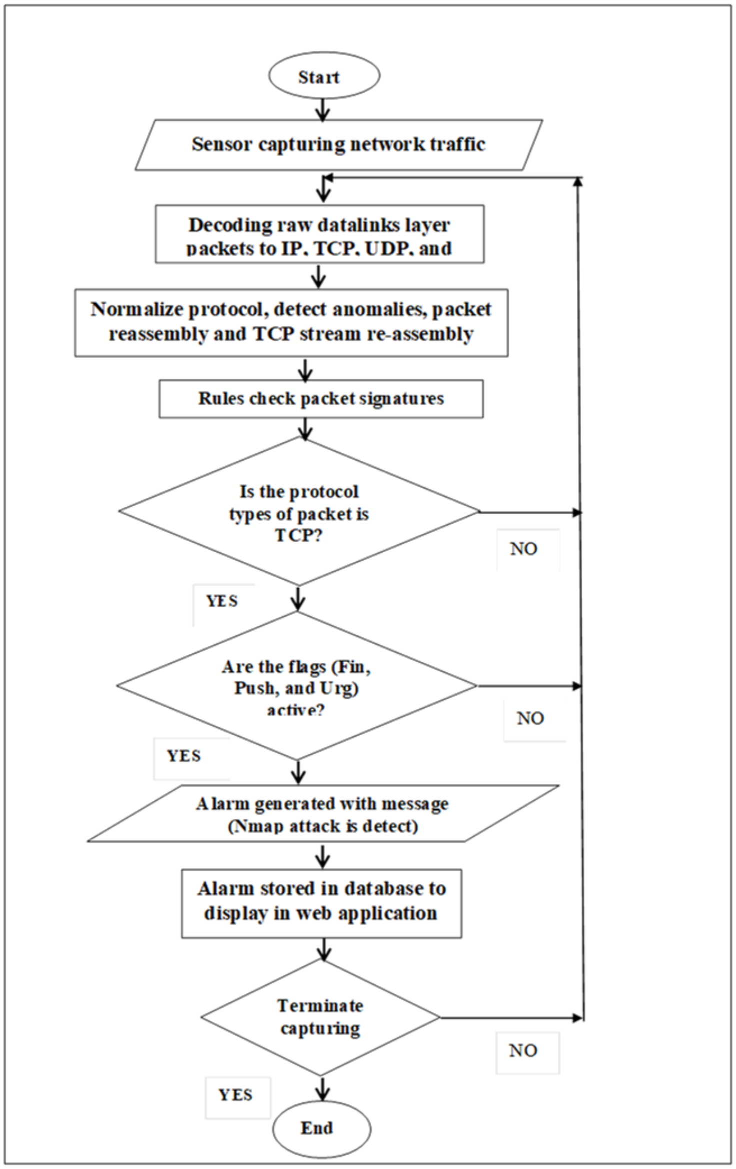
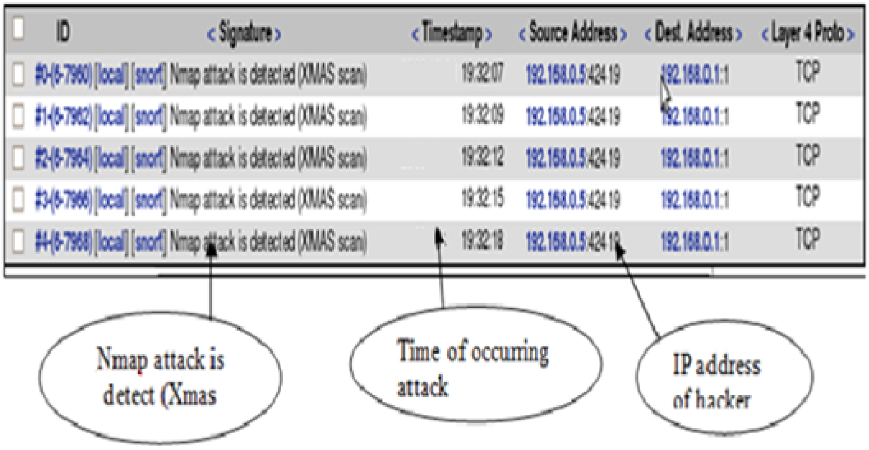

Start
Sensor capturing network traffic
Decoding raw datalinks layer packets to IP, TCP, UDP, and
Normalize protocol, detect anomalies, packet reassembly and TCP stream re-assembly
Rules check packet signatures
Is the protocol types of packet is TCP?
NO
YES
Are the flags (Fin, Push, and Urg) active?
NO
YES
Alarm generated with message (Nmap attack is detect)
Alarm stored in database to display in web application
Terminate capturing
NO
YES
End

Figure 8: Attack detection procedure using zenmap



□	ID	‹Signature›	‹Timestamp›	‹Source Address›	‹Dest. Address›	‹Layer 4 Proto›
□	#0-(6-7960)	[local] [snort] Nmap attack is detected (XMAS scan)	19:32:07	192.168.0.5:42419	192.168.0.1:1	TCP
□	#1-(6-7962)	[local] [snort] Nmap attack is detected (XMAS scan)	19:32:09	192.168.0.5:42419	192.168.0.1:1	TCP
□	#2-(6-7964)	[local] [snort] Nmap attack is detected (XMAS scan)	19:32:12	192.168.0.5:42419	192.168.0.1:1	TCP
□	#3-(6-7966)	[local] [snort] Nmap attack is detected (XMAS scan)	19:32:15	192.168.0.5:42419	192.168.0.1:1	TCP
□	#4-(6-7968)	[local] [snort] Nmap attack is detected (XMAS scan)	19:32:18	192.168.0.5:42419	192.168.0.1:1	TCP

Figure 9: IDS results for zenmap attack

2) *Implementation Unauthorized Access Attack* **(Xhydra)** software tool is used to implement unauthorized access attack. The success of these types of attacks depends on the ability of the hacker to acquire knowledge about the open ports in the target system. This type of information can be obtained from reconnaissance attacks. Xhydra build a wordlist which contains a list of suggested passwords. The hacker can achieve access attacks by using the following procedure:

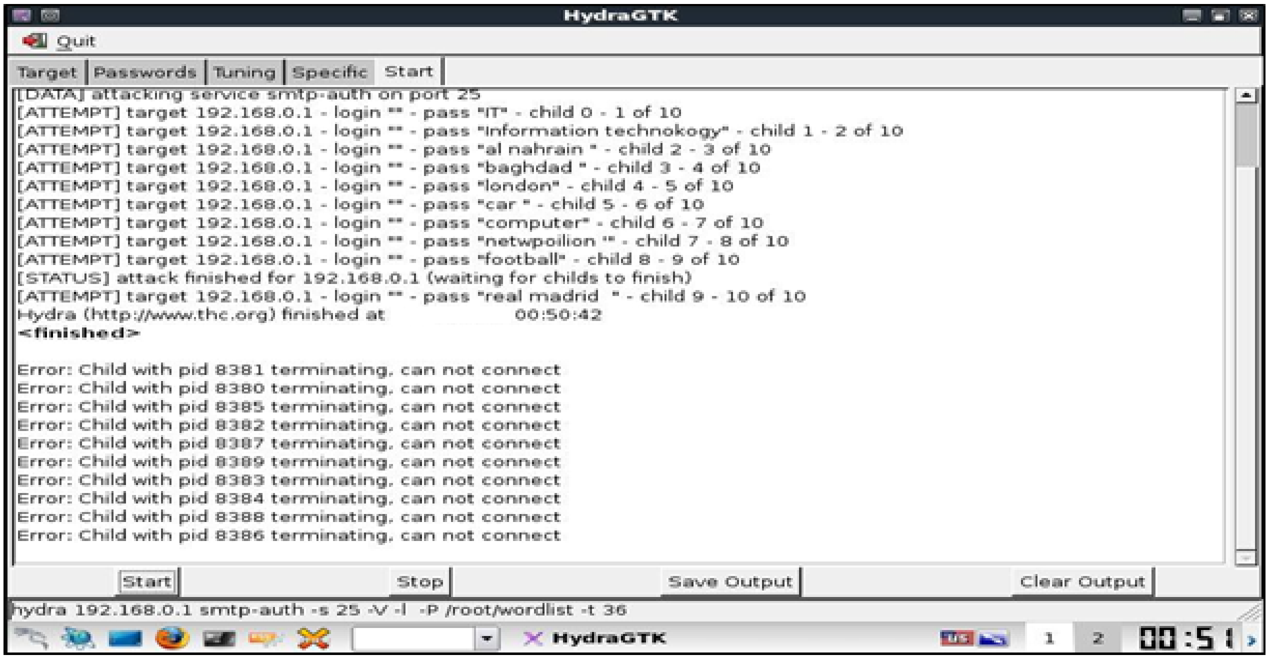
   - Building a file for a suggested wordlist password.
   - Identify an open port in the target system (which obtained from **Zenmap**).
   - Identify an IP address of the target system.
   - Identify the path of wordlist password.

   Fig. 10 shows unsuccessful hacking connection to the target system, because the selected port (25) was closed port therefore the access is failed. Fig. 11 shows port (80) is open and hacking connection access to the target system is successful. The success of unauthorized access attacks is depends on a successful connection to the open ports in the target system and successful suggested password wordlist.

3) Detection of Unauthorized Access Attack Using Packet payload Detection The IDS rules are designed to detect unauthorized access implemented by **Xhydra** tool. Both **(content)** and **(nocase)** keywords in the rule are used to examine the packet payload and to search for the following contents: **(User-Agent Mozilla/4.0(Hydra)**. If this signature is matched then an alarm will be generated. The IDS rule is: **alert tcp any any** $->80$ (msg : $"Xydraattack"; flow : to_server, established; content : "User - Agent\ Mozilla/4.0(Hydra)"; nocase;$**)** This rule is implement only on port (80) which belong to http services. The packet payload by implement Xhydra attack was showing in Fig. 12 Fig. 13 shows an alarm which indicates that an Xhydra attack has occurred, time of the attack, IP address of attacker and source address. Fig. 14 shows the procedure for detecting Xhydra attacks

4) Detection of unauthorized access using packet header detection In this section **Xhydra** attack was detected using

packet header technique. it was seen that two flags (ACK, PSH) are active. The following IDS rule is designed to detect Xhydra attack: **alert tcp any any** $->$ $\$HOME-NET$ **80 (msg:"Xhydra detected by packet header "; flow: stateless; flags:AP)**. The result of detection is shown in Fig. 15

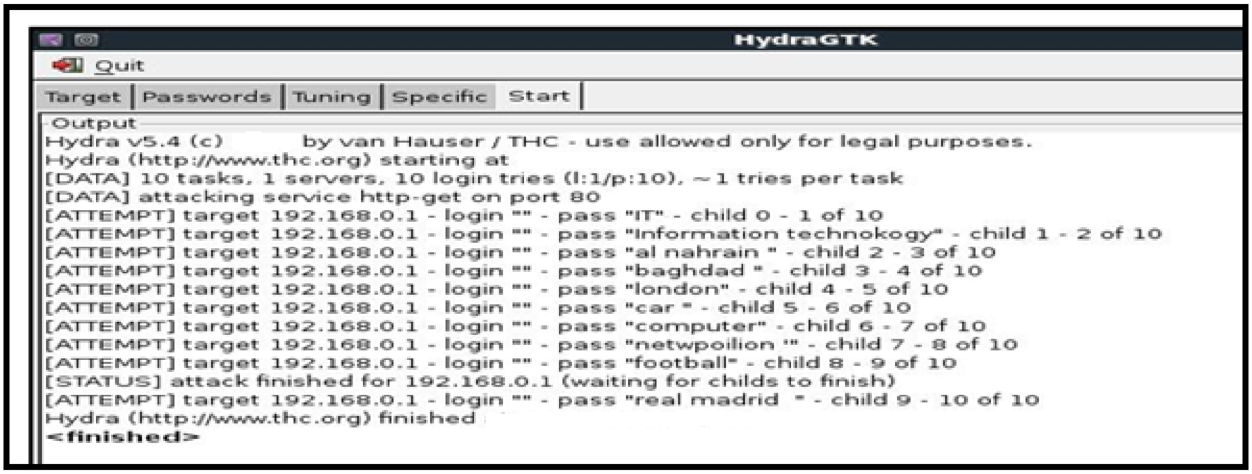


```
[DATA] attacking service smtp-auth on port 25
[ATTEMPT] target 192.168.0.1 - login "" - pass "IT" - child 0 - 1 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "Information technokogy" - child 1 - 2 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "al nahrain " - child 2 - 3 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "baghdad " - child 3 - 4 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "london" - child 4 - 5 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "car " - child 5 - 6 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "computer" - child 6 - 7 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "netwpoilion '" - child 7 - 8 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "football" - child 8 - 9 of 10
[STATUS] attack finished for 192.168.0.1 (waiting for childs to finish)
[ATTEMPT] target 192.168.0.1 - login "" - pass "real madrid  " - child 9 - 10 of 10
Hydra (http://www.thc.org) finished at               00:50:42
<finished>

Error: Child with pid 8381 terminating, can not connect
Error: Child with pid 8380 terminating, can not connect
Error: Child with pid 8385 terminating, can not connect
Error: Child with pid 8382 terminating, can not connect
Error: Child with pid 8387 terminating, can not connect
Error: Child with pid 8389 terminating, can not connect
Error: Child with pid 8383 terminating, can not connect
Error: Child with pid 8384 terminating, can not connect
Error: Child with pid 8388 terminating, can not connect
Error: Child with pid 8386 terminating, can not connect
```

```
hydra 192.168.0.1 smtp-auth -s 25 -V -l  -P /root/wordlist -t 36
```

Figure 10: Xhydra unsuccessful connection to the target system

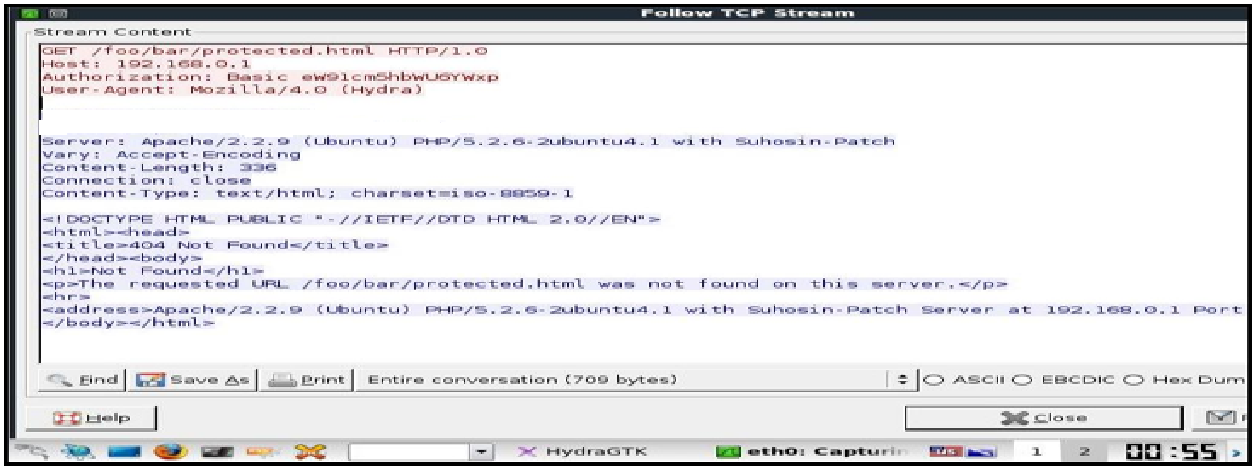


```
Hydra v5.4 (c)         by van Hauser / THC - use allowed only for legal purposes.
Hydra (http://www.thc.org) starting at
[DATA] 10 tasks, 1 servers, 10 login tries (l:1/p:10), ~1 tries per task
[DATA] attacking service http-get on port 80
[ATTEMPT] target 192.168.0.1 - login "" - pass "IT" - child 0 - 1 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "Information technokogy" - child 1 - 2 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "al nahrain " - child 2 - 3 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "baghdad " - child 3 - 4 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "london" - child 4 - 5 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "car " - child 5 - 6 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "computer" - child 6 - 7 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "netwpoilion '" - child 7 - 8 of 10
[ATTEMPT] target 192.168.0.1 - login "" - pass "football" - child 8 - 9 of 10
[STATUS] attack finished for 192.168.0.1 (waiting for childs to finish)
[ATTEMPT] target 192.168.0.1 - login "" - pass "real madrid  " - child 9 - 10 of 10
Hydra (http://www.thc.org) finished
<finished>
```

Figure 11: Attack detection procedure using zenmap

ID <signature>	<Timestamp>	<Source Address>	<Dest. Address>	<Layer 4 proto.>
#0-(6-7167) [local] [snort] Xhydra attack	18.20.38	192.1680.2 56722	192.168.0.1:80	TCP
#1-(6-7166) [local] [snort] Xhydra attack	18.20.32	192.1680.2 56721	192.168.0.1:80	TCP
#1-(6-7165) [local] [snort] Xhydra attack	18.20.28	192.1680.2 56720	192.168.0.1:80	TCP
#1-(6-7164) [local] [snort] Xhydra attack	18.20.24	192.1680.2 56719	192.168.0.1:80	TCP

Figure 12: Packet payload for xhydra attack



```
GET /foo/bar/protected.html HTTP/1.0
Host: 192.168.0.1
Authorization: Basic eW91cm5hbWU6YWxp
User-Agent: Mozilla/4.0 (Hydra)


Server: Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.1 with Suhosin-Patch
Vary: Accept-Encoding
Content-Length: 336
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /foo/bar/protected.html was not found on this server.</p>
<hr>
<address>Apache/2.2.9 (Ubuntu) PHP/5.2.6-2ubuntu4.1 with Suhosin-Patch Server at 192.168.0.1 Port
</body></html>
```

Figure 13: IDS result for xhydra attacks

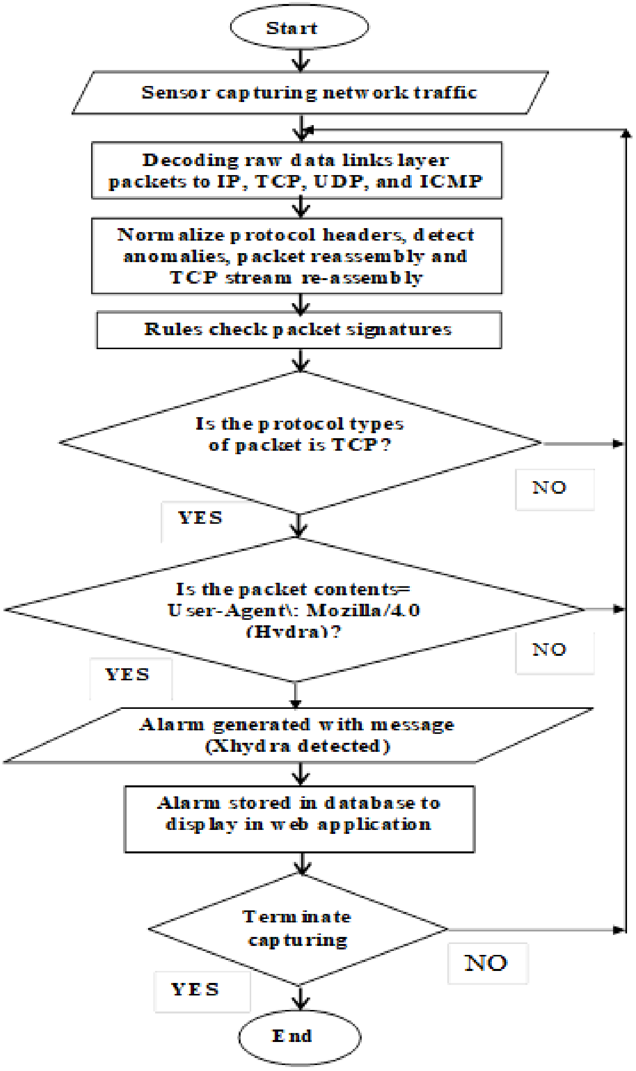
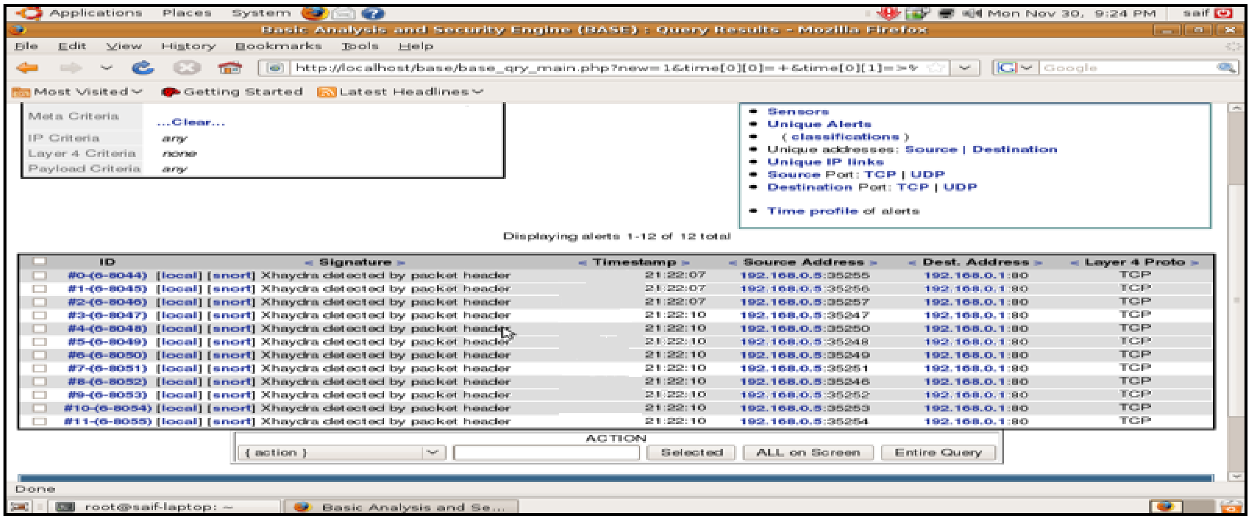

Start
Sensor capturing network traffic
Decoding raw data links layer packets to IP, TCP, UDP, and ICMP
Normalize protocol headers, detect anomalies, packet reassembly and TCP stream re-assembly
Rules check packet signatures
Is the protocol types of packet is TCP?
NO
YES
Is the packet contents= User-Agent\: Mozilla/4.0 (Hydra)?
NO
YES
Alarm generated with message (Xhydra detected)
Alarm stored in database to display in web application
Terminate capturing
NO
YES
End

Figure 14: Detection procedure for xhydra attack


Applications Places System
Mon Nov 30, 9:24 PM
saif
Basic Analysis and Security Engine (BASE) : Query Results - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/base/base_qry_main.php?new=1&time[0][0]=+&time[0][1]=>%
Google
Most Visited
Getting Started
Latest Headlines

Meta Criteria	...Clear...
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

- 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 1-12 of 12 total

□	ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
□	#0-(6-8044)	[local] [snort] Xhaydra detected by packet header	21:22:07	192.168.0.5:35255	192.168.0.1:80	TCP
□	#1-(6-8045)	[local] [snort] Xhaydra detected by packet header	21:22:07	192.168.0.5:35256	192.168.0.1:80	TCP
□	#2-(6-8046)	[local] [snort] Xhaydra detected by packet header	21:22:07	192.168.0.5:35257	192.168.0.1:80	TCP
□	#3-(6-8047)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35247	192.168.0.1:80	TCP
□	#4-(6-8048)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35250	192.168.0.1:80	TCP
□	#5-(6-8049)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35248	192.168.0.1:80	TCP
□	#6-(6-8050)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35249	192.168.0.1:80	TCP
□	#7-(6-8051)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35251	192.168.0.1:80	TCP
□	#8-(6-8052)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35246	192.168.0.1:80	TCP
□	#9-(6-8053)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35252	192.168.0.1:80	TCP
□	#10-(6-8054)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35253	192.168.0.1:80	TCP
□	#11-(6-8055)	[local] [snort] Xhaydra detected by packet header	21:22:10	192.168.0.5:35254	192.168.0.1:80	TCP

Figure 15: Xhydra detected using packet header

IX. Conclusions

In this paper two types of attacks (Reconnaissance and Unauthorized access) were implemented to evaluate IDS performance. The result show a main limitation of the IDS in being able to detect only predefine attacks signature for both packet header and payloads. The result of Reconnaissance attacks is important in applying other attacks. The system administrator must frequently update to the IDS rule according to the analysis of system alerts database to detect any new types of attacks. Unauthorized access attacks can be highly successful if the target system is using a weak password. Using (threshold) keyword can useful to reduce number of alerts generated by IDS in similar attacks. Xhaydra attacks can be detected in two ways using packet header and packet payload Reconnaissance attacks result is necessary to success another attacks as shown in applying Xhydra attack.

References

[1] S. Kumar, " Classification and Detection of Computer Intrusions" , Phd Thesis, Purdue University, 1995.
[2] W. Edwards, T. Lammle, T. Lancaster, " CCSP Complete Study Guide" , Sybex, 2005.
[3] Chi- Chun Lo, Chun- Chieh Huang, Joy Ku. , "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks" , 39th International Conference on Parallel Processing Workshops, IEEE computer society, 2010.
[4] Bhavin Shah, Bhushan H. , " Artificial Neural Network based Intrusion Detection System: A Survey " , International Journal of Computer Applications, 2012, Volume 39- No. 6.
[5] Ketan Sanjay Desale, Chandrakant Namdev Kumathekar and Arjun Pramod Chavan, " Efficient Intrusion Detection System using Stream Data Mining Classification Technique" , International Conference on Computing Communication Control and Automation, 2015 IEEE.
[6] Priyanka Pawar and Damodar Tiwari, " Intrusion Detection System based on Particle Swarm Optimized Neural Network" , International Journal of Digital Application & Contemporary Research, Volume 4, Issue 11, June 2016.
[7] M. Belouch, S. El Hadaj, M. Idhammad, " Performance evaluation of intrusion detection based on machine learning using Apache Spark" , The First International Conference On Intelligent Computing in Data Sciences, Procedia Computer Science 127 (2018) 35- 41, Published by Elsevier B. V, 2018.
[8] R. Zhang, X. Xiao, "Intrusion Detection in Wireless Sensor Networks with an Improved NSA Based on Space Division" , Journal of Sensors, Volume 2019, Article ID 5451263, Hindawi Limited, UK, 2019.
[9] T. Dean, " Network+ 2005 In Depth" , Thomson Course Technology, 2005.
[10] D. Reynders, E. Wright, 2003, "Practical TCP/ IP Ethernet networking", IDC technology
[11] S. Panwar, S. Mao, J. dong Ryoo, Y. Li, "TCP/IP Essentials, A Lab-Based Approach" , Cambridge University Press, 2004.

[12] W. Lee, S. Stolfo, " Adaptive Intrusion Detection: a Data Mining Approach" , Computer science Department, Columbia University, Kluwer Academic Publishers, 2000.
[13] Endorf , E. Schultz, J. Mellander, "Intrusion Detection & Prevention" , McGraw- Hill, 2004.
[14] A. Orebaugh, G. Ramirez, J. Burke, L. Pesce, J. Wright, G. Morris, " Wireshark & Ethereal Network Protocol Analyzer Toolkit" , Syngress Publishing, Inc, 2007.
[15] S. Harris, A. Harper, C. Eagle, J. Nessm, " Gray hat hacking" , second edition Macgrawhill, 2008.
[16] J. Burton, I. Dubrawsky, V. Osipov, C. Tate Baumrucker , M. Sweeney, " Cisco Security Professional Guide To Secure Intrusion Detection System " , 2008.
[17] A. Orebaugh, B. Pinkard , " Nmap in the enterprise your guide to network scanning" , Syngress Publishing, Inc , 2008.
[18] S. Mcclure, J. Scambray, G. Kurtz, " Hacking Exposed, Network Security Secret & Solution" , 6th edition, McGraw Hill, 2009.
[19] M. Shema, C. Davis, A. Philipp, D. Cowen, "Anti hacker tool kit " , 3rd edition, McGraw- Hill/ Osborne, 2012.