Recent Advances in Intrusion Detection Systems: An Analytical Evaluation and Comparative Study

Ashish Kumar
School of Information Technology,
MATS University, Raipur
(C.G.), India-492001

Shrikant Chandak
School of Information Technology,
MATS University, Raipur
(C.G.), India-492001

Rita Dewanjee
School of Information Technology,
MATS University, Raipur
(C.G.), India-492001

ABSTRACT

As the emergence of computer, the use and effect of internet in our daily life is increased day by day. Security has become the greatest problem within and outside the organizations. User ID, passwords and firewalls are the common steps that organizations take to secure their computers. However, these are not so effective mediums in current context of unsecure eon. Intruders and attackers are so advanced that they access the computer and manipulate it, so they cannot be traced easily. Through this contribution objective is to find out & present existing intrusion detection system (IDS) with their pros and cons that will be helpful to select the best one.

General Terms

IPS (Intrusion Prevention System), IDS (Intrusion Detection System), Denial-of-Service, Snort, Security.

Keywords

Network Security, Intrusion Detection, Intrusion Prevention, Social Engineerimg Toolkit.

1. INTRODUCTION

Day by day internet is growing and reached to everyone affects on many. People’s dependencies are also increased with this. Most of our transaction are now shifted to the internet, thus it is also become a soft area for various types of security threats and attracts various types of attackers whose objective can be data stealing or data diddling, sniffing, password cracking or any other form of misuse of unauthorized data. So IDS can be the best solution for the above problem.

“Intrusion detection system plays a very crucial role in security of organization’s information and data. It is a system that checks systems and network traffic and inspects that attacks originating from inside or outside the organization”.[1].

These are the some selecting criteria of IDS

- Objective/Aim Identification (Perform a risk/attack evaluation).
- General Overview of available IDS.
- Define the need of IDS precisely.
- List out the IDS according to need or Purpose.
- Cost Expense (Benefit) Analysis & Selection of IDS.
- Decide Working & Policy (Implementation).

2. EXPLORATORY STUDY ON EXISTING IDS

Based on the study of various research contributions [2] [3] [4] [5], it has been found that around hundred IDS are available. Through this table some popular ones are presented here:

Table 1. Currently Available IDS: At a Glance

AnaDisk	AuditTrack for Netware	Authd
AIDE (Advanced Intrusion Detection Environment)	AppShield	BlackICE Defender
Bro	bv-LifeLine	Check Point RealSecure
Cisco Secure IDS	Clog	CMDS Computer Misuse Detection System
CRCMd5 Data Validation Tool	CyberCop Monitor NT	CyberCop Monitor Solaris
CyberCop Scanner	CyberCop Sting	DesktopSentry
Detect IT	DiskSearch	Dragon Sensor
eTrust Internet Defense	Extercept	Exodus Cyber Attack Management Service
Firewall Reporting Suite	FireStorm	ForensicToolkit
Gabriel	HP Openview Node Sentry	HP-Tcpdump
ICEcap	ICEpac	iD2 Secure Transport
Ifstatus	Incident Manager	IP-Watcher
ISensor Intrusion Prevention	Kane Security Monitor	KaVaDo InterDo
Klaxon	LSOF	Lucent RealSecure

Man Hunt	NetDetector	NetIQ Corp. Security Manager
Netlog	Network Flight Recorder	NetProwler
neuSecure	OSSEC HIDS	OnlineGuardian
Patriot IDS	Prelude Hybrid IDS	Peek & Spy
QRadar	RealSecure	Reactive IDS
Red Sire, 1st Watch Managed IDS	Review	SafeBack
Samhain	Security Management Pack for MOM 2000	Security Manager
Sentry	Shadow	SilentRunner
SMARTWatch	Snort	Strata Guard
StealthWatch	Suricata	Tcpdump
Tcp_wrappers	Tklogger	Tocsin
Tripwire	T-sight	TTY-Watcher
TurnKey Network Appliance	Vanguard Enforcer	Verizon Federal Network System NetFacade
ViewDisk	Vigient Security Agents	WinSNORT

Each IDS has specific design goals and objectives, thus having some specific/unique as well as some common features. From above listed IDSs, some well-known IDSs used by the organizations and now become de-facto standard as well, described here in brief with their features:

Table 2. Comparison Table of some well-known IDS

Parameters	Bro	Snort	NFR	Suricata	Dragon Squire
Operating system	Unix	Win/Unix/Mac	Unix	Win/Unix/ Mac/BSD	Unix
Analysis GUI	Less	Many	Less	Many	Standalone
Installation/ Deployment	Typical	Easy	Easy	Intermediate	Easy
Open Source	Yes	Yes	Yes	Yes	No
IPS Capability	No	Yes	No	Yes	No
Large User Community	No	Yes	-	No (Emerging)	Less
Speed	Faster	Fast	Medium	Faster	Fast
Throughput	Maximum	Moderate	Moderate	Maximum	Maximum
Network Usage	Less	Medium	Less	Very Less	Medium
Intrusion Detection Technique	Anomaly Based	Signature based	Signature based	Signature based	Host based

These well-known IDSs with their conclusive highlights are as follows:

2.1 Bro

It is an anomaly based IDS, provides a real time network or pure traffic analysis that match the captured packets with desired rules applied by the user. It is used to analyze extensive logging functions related to application level details and to recording every connection observed on the wires. [6]

Highlights of Bro are as follows:

- Open-source.
- Specifically timely for scientific environments.
- Very useful in servers with default policies.
- Powerful on traded hardware.
- Intrusion prevention schemes are also supported.

The major benefits of Bro IDS are permitting it to measures from the desires of smaller institutions to those of the biggest research universities. Bro is implemented as a cluster set of conditions that uses three kinds of methods: Manager, Worker, and Proxy.

2.2 Snort

Snort has become the de-facto accepted for signature-based network IDS. It is based on the Libpcap library to capture packets. Snort is lightweight cross-platform network sniffing tool; it's evolved into a strong and full-featured IDPS. Snort engine allowed a single rule to be applied to any variation of a protocol. [7]

Some features of Snort are as follows:

- Can work on any operating system.
- Protocol examining capability.
- Condition examining capability.
- Packet reassembly capability.
- Lots of serving GUI's are available for analyzing the results.

Snort is configured using command line switches and optional Berkeley Packet Filter commands. Snort rules can be easily written by normal users, but powerful enough to detect a wide variety of hostile or merely suspicious network traffic. There are three basic actions that Snort can trigger when a packet matches a specified rule pattern: log, alert, or pass. [8]

2.3 NFR

Network Flight Recorder (NFR) gives the users a powerful tool to get the illegal attack in networks. With the efficiency of this tool, network administrator can know better about who is using or accessing their network and what their workers are doing. It is stand-alone configuration; a single NFR station gathers and stores information in a single instance.

Analysis in NFR is developed by script based language knows as N-code, it uses web based interface to display the result with the use of java language. By its feature of not to interfere in network activity, this helps to analyze the standard or to free data from unwanted errors. Through the NFR one can store, retrieve, or archived the information to external drives [9] as well. However, this doesn't eliminate the requirement specialist to first analyze and categorize attack situations and system vulnerabilities, and hand-code the analogous rules and model or patterns in N-code for misuse detection. Due to the manual and ad hoc nature of the development process, NFR has low extensibility and adaptability. [10]

Highlights of the NFR are as follows:

- It provides built-in means for customization and extension.
- It is determined language, i.e., flexible and portable language for traffic exploration.
- It doesn't obstruct with network activity
- Having dynamic alerting function.

It works on UNIX based systems and is available in source code form. As it was previously available open source but now under proprietary license.

NFR is a multiuse network monitoring implement, operational for intrusion detection, usage exploration, and troubleshooting by system administrators. The construed N-code language allows a user to write randomly complex scripts for examining incoming packets, in which NFR engine provide timing constrains to limit packets which helps to share system resources properly.

NFR comprises of a sum of modules, all responsible for a specific activity: packet suckers, a decision engine, back ends and a query interface. [11]

2.4 Suricata

An open source, high performance recent signature based network IDPS and Security Monitoring System for UNIX and Window based systems. It use PCAP recorder to log the traffic and provides offline analysis facility of PCAP files. [7][12]

Some Additional features are:

- Supports all operating system.
- Along with the IPS.
- Automatic detection of protocols with high performance.
- Network Security Monitoring (NSM).
- Filtering of alerts and events.
- Output format support many other tools to analyze data.

Suricata detects many anomalies in traffic it investigates. Having functionality of automatic detection of protocols on any port is very useful which helps to determine the malware and attacks.

Another progress in the Suricata engine is to employ native multi-threaded actions, something more necessary as network bandwidth increases. Suricata is planned from the start to take improvement of operating with multiple CPUs. A multi-threaded discovery engine can make intelligent results on how to fragment processing and can match signature detection between these threads all within the same detection engine. [13]

2.5 Dragon Squire

The Dragon IDS products come across the experiments of inspecting over a modern network by providing high speed sensors to sense suspicious activity, allowed data to decide the impact of network attacks and scalability to set up and able to huge numbers of sensors, lacking negatively impacting the task of large networks. [14][15]

Dragon Squire is a host-based intrusion detection and firewall monitoring system that express at system logs for proof of malicious or suspicious submission activity, and monitors key system files for proof of damaging in real time. Dragon Squire has been tuned to prevent high load levels and minimize any negative system impact to a server's performance. Besides being an excellent system security tool, it also examines firewall logs, router actions and just about everything that can communicate SNMP or SYSLOG.

Some features of Dragon Squire are as follows:

- Ease of use because of single interface.
- Host & network based versions.
- Clever to work over high-speed networks without dropping packets.
- Able to replay stored attacks for post-mortem and forensic analysis.
- Strong reporting capabilities.
- It has the ability to detect evasive IDS Techniques.
- GUI / command line alternative if preferred.
- Capability of secure remote administration.

The Dragon Squire's signature archive contains suspicious events from a comprehensive range of operating systems. These actions check for guarding file transfers, failed login attempts, physical messages and system reboots. The library contains security messages such as Secure Shell, Qmail, Send mail and Apache Web servers.

It is used to observe network packets for proof of hacker and malicious worker activity by monitoring their system logs and firewall logs. It analyzes the system logs for proof of malicious or suspicious application programs in real time. It also observes main system records for evidence of tampering.

Being a tremendous system security tool, it can examine firewall logs, router events and reasonable about everything that can start SNMP or SYSLOG.

Dragon Squire has been engineered to have a slight influence on the servers it is protecting. It is also a superb supplement to the Dragon Sensor Network IDS.

3. EXPERIMENTAL SCENARIO

For experimental purpose a popular IDS, SNORT tool has been selected. Two hosts primarily connected to a LAN has been used, one become a victim machine and another becomes an attacker machine. SNORT, is installed on victim machine having Windows 8 and Intel's X86-based PC (64-bits i5-2430M CPU@2.4GHz Processor with 2 GB RAM) configuration. Dummy attacker's machine having the same hardware configuration with BackTrack 5 R2 (a flavor of Linux) operating system [16]. All recorded data is contained in a local snort log file. The experiment was done to identify particular attacks and some packets. The traffic speed was low. With the use of Wireshark Packet Analyzer [17] the packets has been analyzed.

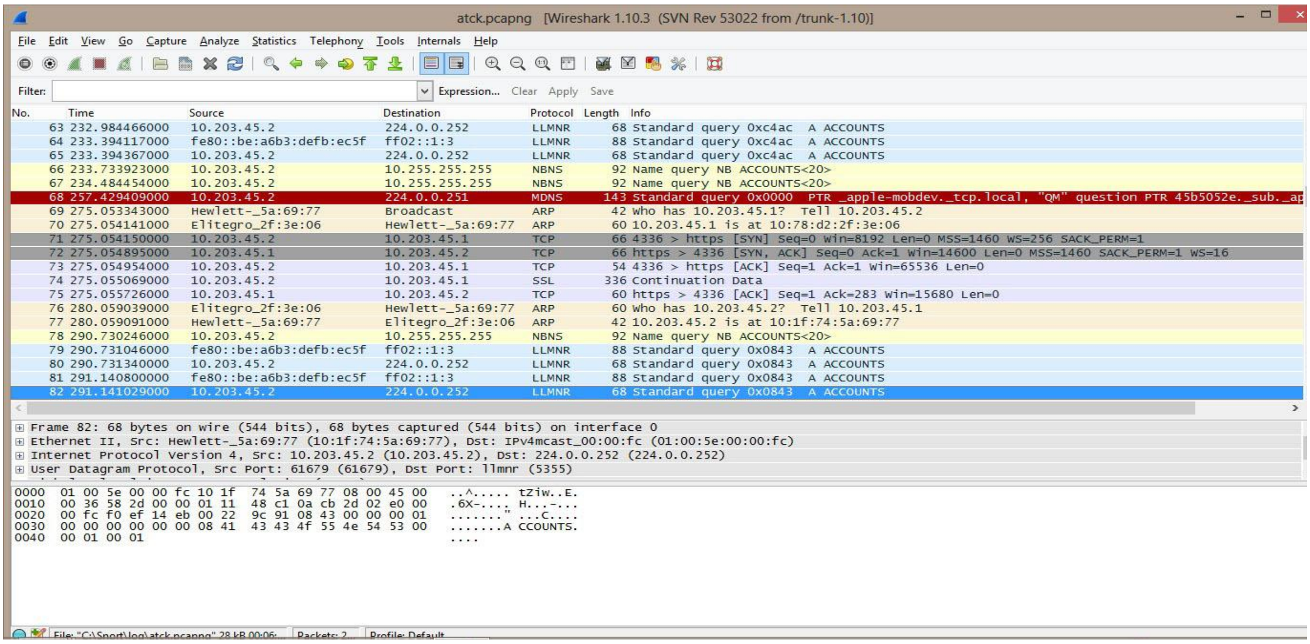
The three types of intrusions has been evaluated and examined for experimental purposes at here:

- Attempt by a program to access the network resources.
- Denial-of-service (e.g., Ping-of-death).
- Unauthorized access from a remote machine (using Social Engineer Toolkit-SET [18]).

4. EXPERIMENT RESULTS

The generated log files are used to save the captured packets. The snort log file should be converted to ".csv" format for analysis in Wireshark. The log file of snort contains the following information, No., time, Source, Destination, Protocol, length, & Info. Wireshark uses color coding to differentiate the packets received.

4.1 Unauthorized Access by a Program to the Network Resources

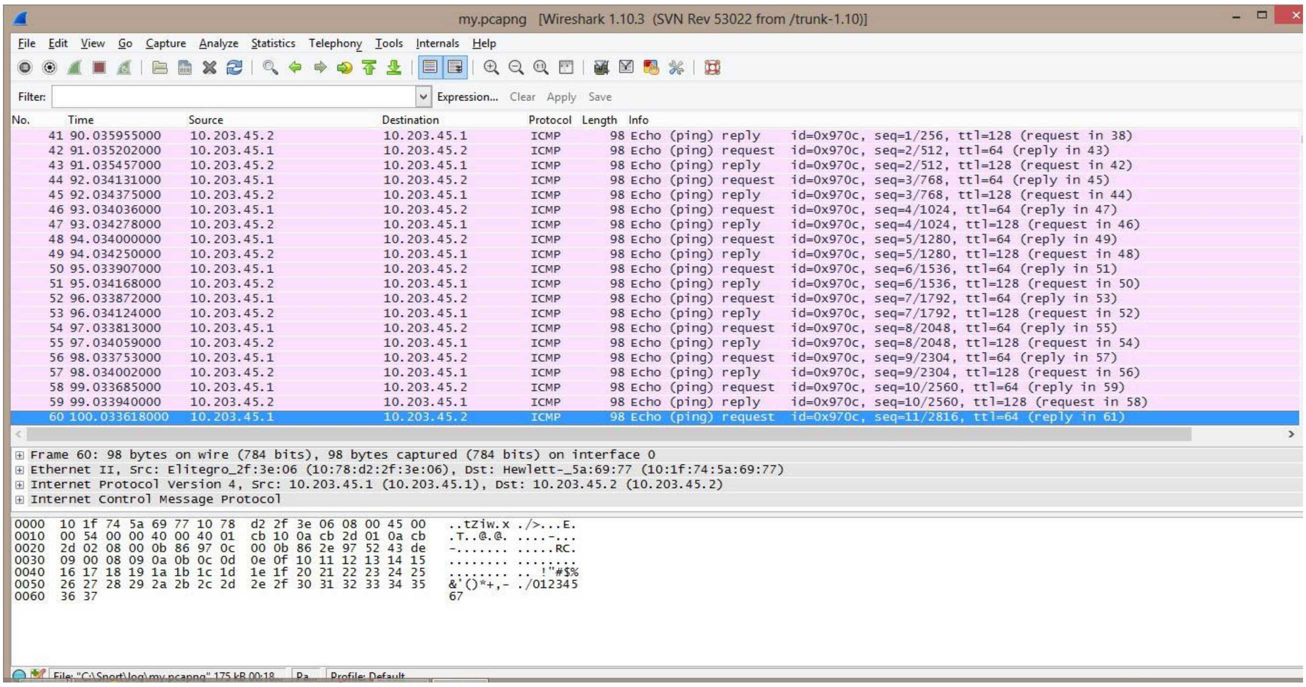

atck.pcapng [Wireshark 1.10.3 (SVN Rev 53022 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time            Source                      Destination         Protocol Length Info
  63 232.984466000   10.203.45.2                 224.0.0.252         LLMNR     68 Standard query 0xc4ac  A ACCOUNTS
  64 233.394117000   fe80::be:a6b3:defb:ec5f     ff02::1:3           LLMNR     88 Standard query 0xc4ac  A ACCOUNTS
  65 233.394367000   10.203.45.2                 224.0.0.252         LLMNR     68 Standard query 0xc4ac  A ACCOUNTS
  66 233.733923000   10.203.45.2                 10.255.255.255      NBNS      92 Name query NB ACCOUNTS<20>
  67 234.484454000   10.203.45.2                 10.255.255.255      NBNS      92 Name query NB ACCOUNTS<20>
  68 257.429409000   10.203.45.2                 224.0.0.251         MDNS     143 Standard query 0x0000  PTR _apple-mobdev._tcp.local, "QM" question PTR 45b5052e._sub._ap
  69 275.053343000   Hewlett-_5a:69:77           Broadcast           ARP       42 Who has 10.203.45.1?  Tell 10.203.45.2
  70 275.054141000   Elitegro_2f:3e:06           Hewlett-_5a:69:77   ARP       60 10.203.45.1 is at 10:78:d2:2f:3e:06
  71 275.054150000   10.203.45.2                 10.203.45.1         TCP       66 4336 > https [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
  72 275.054895000   10.203.45.1                 10.203.45.2         TCP       66 https > 4336 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
  73 275.054954000   10.203.45.2                 10.203.45.1         TCP       54 4336 > https [ACK] Seq=1 Ack=1 Win=65536 Len=0
  74 275.055069000   10.203.45.2                 10.203.45.1         SSL      336 Continuation Data
  75 275.055726000   10.203.45.1                 10.203.45.2         TCP       60 https > 4336 [ACK] Seq=1 Ack=283 Win=15680 Len=0
  76 280.059039000   Elitegro_2f:3e:06           Hewlett-_5a:69:77   ARP       60 Who has 10.203.45.2?  Tell 10.203.45.1
  77 280.059091000   Hewlett-_5a:69:77           Elitegro_2f:3e:06   ARP       42 10.203.45.2 is at 10:1f:74:5a:69:77
  78 290.730246000   10.203.45.2                 10.255.255.255      NBNS      92 Name query NB ACCOUNTS<20>
  79 290.731046000   fe80::be:a6b3:defb:ec5f     ff02::1:3           LLMNR     88 Standard query 0x0843  A ACCOUNTS
  80 290.731340000   10.203.45.2                 224.0.0.252         LLMNR     68 Standard query 0x0843  A ACCOUNTS
  81 291.140800000   fe80::be:a6b3:defb:ec5f     ff02::1:3           LLMNR     88 Standard query 0x0843  A ACCOUNTS
  82 291.141029000   10.203.45.2                 224.0.0.252         LLMNR     68 Standard query 0x0843  A ACCOUNTS

⊞ Frame 82: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
⊞ Ethernet II, Src: Hewlett-_5a:69:77 (10:1f:74:5a:69:77), Dst: IPv4mcast_00:00:fc (01:00:5e:00:00:fc)
⊞ Internet Protocol Version 4, Src: 10.203.45.2 (10.203.45.2), Dst: 224.0.0.252 (224.0.0.252)
⊞ User Datagram Protocol, Src Port: 61679 (61679), Dst Port: llmnr (5355)

0000  01 00 5e 00 00 fc 10 1f  74 5a 69 77 08 00 45 00   ..^..... tZiw..E.
0010  00 36 58 2d 00 00 01 11  48 c1 0a cb 2d 02 e0 00   .6X-.... H...-...
0020  00 fc f0 ef 14 eb 00 22  9c 91 08 43 00 00 00 01   ......." ...C....
0030  00 00 00 00 00 00 08 41  43 43 4f 55 4e 54 53 00   .......A CCOUNTS.
0040  00 01 00 01                                        ....

File: "C:\Snort\log\atck.pcapng" 28 kB 00:06:...   Packets: 2...   Profile: Default
```

Fig 1: Attempt by a program to access the network resources

Figure 1 shows the detail view of packets that are not trusted by the machine as the valid traffic in the network. Here a packet from the machine having IP address 10.203.45.2 from an unknown location in the network to the machine having IP address 224.0.0.251 is an illegal traffic.

4.2 Denial of Service (Ping-of-death)


my.pcapng [Wireshark 1.10.3 (SVN Rev 53022 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
File: "C:\Snort\log\my.pcapng" 175 kB 00:18...
Pa...
Profile: Default

```
No.  Time            Source        Destination   Protocol Length Info
  41 90.035955000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=1/256, ttl=128 (request in 38)
  42 91.035202000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=2/512, ttl=64 (reply in 43)
  43 91.035457000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=2/512, ttl=128 (request in 42)
  44 92.034131000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=3/768, ttl=64 (reply in 45)
  45 92.034375000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=3/768, ttl=128 (request in 44)
  46 93.034036000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=4/1024, ttl=64 (reply in 47)
  47 93.034278000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=4/1024, ttl=128 (request in 46)
  48 94.034000000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=5/1280, ttl=64 (reply in 49)
  49 94.034250000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=5/1280, ttl=128 (request in 48)
  50 95.033907000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=6/1536, ttl=64 (reply in 51)
  51 95.034168000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=6/1536, ttl=128 (request in 50)
  52 96.033872000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=7/1792, ttl=64 (reply in 53)
  53 96.034124000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=7/1792, ttl=128 (request in 52)
  54 97.033813000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=8/2048, ttl=64 (reply in 55)
  55 97.034059000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=8/2048, ttl=128 (request in 54)
  56 98.033753000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=9/2304, ttl=64 (reply in 57)
  57 98.034002000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=9/2304, ttl=128 (request in 56)
  58 99.033685000    10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=10/2560, ttl=64 (reply in 59)
  59 99.033940000    10.203.45.2   10.203.45.1   ICMP     98 Echo (ping) reply    id=0x970c, seq=10/2560, ttl=128 (request in 58)
  60 100.033618000   10.203.45.1   10.203.45.2   ICMP     98 Echo (ping) request  id=0x970c, seq=11/2816, ttl=64 (reply in 61)
```

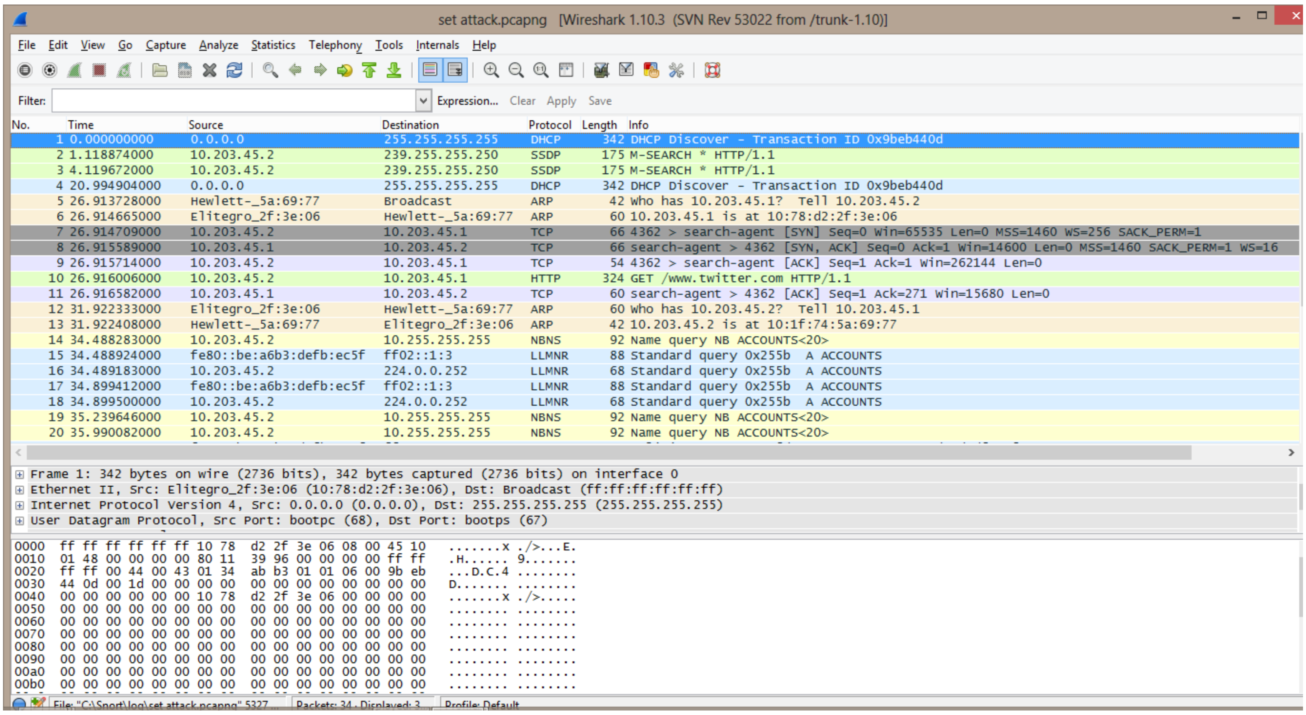
```
⊞ Frame 60: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
⊞ Ethernet II, Src: Elitegro_2f:3e:06 (10:78:d2:2f:3e:06), Dst: Hewlett-_5a:69:77 (10:1f:74:5a:69:77)
⊞ Internet Protocol Version 4, Src: 10.203.45.1 (10.203.45.1), Dst: 10.203.45.2 (10.203.45.2)
⊞ Internet Control Message Protocol
```

```
0000  10 1f 74 5a 69 77 10 78  d2 2f 3e 06 08 00 45 00   ..tZiw.x ./>...E.
0010  00 54 00 00 40 00 40 01  cb 10 0a cb 2d 01 0a cb   .T..@.@. ....-...
0020  2d 02 08 00 0b 86 97 0c  00 0b 86 2e 97 52 43 de   -....... .....RC.
0030  09 00 08 09 0a 0b 0c 0d  0e 0f 10 11 12 13 14 15   ........ ........
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ........ .. !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Fig 2: Denial-of-service (e.g., Ping-of-death)

In the figure 2, the ping of death attack is shown, that lead to Denial of Service. The machine having IP address 10.203.45.1 generated ping attack for the machine having IP address 10.203.45.2.

4.3 Social Engineering Attack (Cloning twitter.com)


set attack.pcapng [Wireshark 1.10.3 (SVN Rev 53022 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
File: "C:\Snort\log\set attack.pcapng" 5327 ...
Packets: 34 · Displayed: 3...
Profile: Default

```
No.  Time           Source                    Destination        Protocol Length Info
   1 0.000000000    0.0.0.0                   255.255.255.255    DHCP     342 DHCP Discover - Transaction ID 0x9beb440d
   2 1.118874000    10.203.45.2               239.255.255.250    SSDP     175 M-SEARCH * HTTP/1.1
   3 4.119672000    10.203.45.2               239.255.255.250    SSDP     175 M-SEARCH * HTTP/1.1
   4 20.994904000   0.0.0.0                   255.255.255.255    DHCP     342 DHCP Discover - Transaction ID 0x9beb440d
   5 26.913728000   Hewlett-_5a:69:77         Broadcast          ARP       42 Who has 10.203.45.1?  Tell 10.203.45.2
   6 26.914665000   Elitegro_2f:3e:06         Hewlett-_5a:69:77  ARP       60 10.203.45.1 is at 10:78:d2:2f:3e:06
   7 26.914709000   10.203.45.2               10.203.45.1        TCP       66 4362 > search-agent [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
   8 26.915589000   10.203.45.1               10.203.45.2        TCP       66 search-agent > 4362 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
   9 26.915714000   10.203.45.2               10.203.45.1        TCP       54 4362 > search-agent [ACK] Seq=1 Ack=1 Win=262144 Len=0
  10 26.916006000   10.203.45.2               10.203.45.1        HTTP     324 GET /www.twitter.com HTTP/1.1
  11 26.916582000   10.203.45.1               10.203.45.2        TCP       60 search-agent > 4362 [ACK] Seq=1 Ack=271 Win=15680 Len=0
  12 31.922333000   Elitegro_2f:3e:06         Hewlett-_5a:69:77  ARP       60 Who has 10.203.45.2?  Tell 10.203.45.1
  13 31.922408000   Hewlett-_5a:69:77         Elitegro_2f:3e:06  ARP       42 10.203.45.2 is at 10:1f:74:5a:69:77
  14 34.488283000   10.203.45.2               10.255.255.255     NBNS      92 Name query NB ACCOUNTS<20>
  15 34.488924000   fe80::be:a6b3:defb:ec5f   ff02::1:3          LLMNR     88 Standard query 0x255b  A ACCOUNTS
  16 34.489183000   10.203.45.2               224.0.0.252        LLMNR     68 Standard query 0x255b  A ACCOUNTS
  17 34.899412000   fe80::be:a6b3:defb:ec5f   ff02::1:3          LLMNR     88 Standard query 0x255b  A ACCOUNTS
  18 34.899500000   10.203.45.2               224.0.0.252        LLMNR     68 Standard query 0x255b  A ACCOUNTS
  19 35.239646000   10.203.45.2               10.255.255.255     NBNS      92 Name query NB ACCOUNTS<20>
  20 35.990082000   10.203.45.2               10.255.255.255     NBNS      92 Name query NB ACCOUNTS<20>
```

```
⊞ Frame 1: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
⊞ Ethernet II, Src: Elitegro_2f:3e:06 (10:78:d2:2f:3e:06), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
⊞ User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
```

```
0000  ff ff ff ff ff ff 10 78  d2 2f 3e 06 08 00 45 10   .......x ./>...E.
0010  01 48 00 00 00 00 80 11  39 96 00 00 00 00 ff ff   .H...... 9.......
0020  ff ff 00 44 00 43 01 34  ab b3 01 01 06 00 9b eb   ...D.C.4 ........
0030  44 0d 00 1d 00 00 00 00  00 00 00 00 00 00 00 00   D....... ........
0040  00 00 00 00 00 00 10 78  d2 2f 3e 06 00 00 00 00   .......x ./>.....
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Fig 3: An Example of Social Engineering Attack (Cloning a social website twitter.com)

The social engineering attack is one of the advanced techniques used in the Metasploit framework to get a remote connection to the computer. The attackers utilize this attack to compromise the system for data theft by privilege escalation.

If the attacker successfully runs privilege escalation he can get the administrator account. And the whole control of the system is transferred to the attacker who can use the system as the Trojan machine.

ReverseTCP connection was generated after trying to access twitter.com. It leads access to the machine having IP address 10.203.45.2 by using command line interface.

5. DISCUSSION ON EXPERIMENTAL RESULTS

The experiment shows how to configure SNORT and WireShark on Windows 8 machines. It is better to install Win Snort in place of Snort because it provides a GUI for better user interaction. The ReverseTCP successfully led the attacker to gain access to the system. It should have shown an alarm like a warning message on commad-line, with the details like the port number and type of protocol that is used to access the system, the destination and source IP addresses, with a particular color coding.

The packets also revealed unauthorized access by a program to the network resource when analyzed using WireShark.

The denial of service attack was captured and it showed continuous ping packets generated to the target machine.

The social engineering attack with the help of SET has been successfully executed. A social website twitter.com was cloned and run on the attacker's machine. When victim machine try to access the cloned website (i.e., twitter.com), it lets the backdoor to get into execute on the victim machine and generates a connection back to the attacker machine. The exploit used is aurora-internet explorer that allowed executing the payload on the victim machine.

6. CONCLUSION

There are many Open source IDS available for use, but Snort is the best alternative system to ensure network security. It is non GUI interface, takes time to get familiar. Through this paper, a detailed study on IDS has been performed and a popular IDS Snort has been implemented and configured on Windows-based environment. Some well known attacks has been executed and demonstrated successfully and the system was compromised. Snort was only used to capture the packets and WireShark analysis showed the illegal packets. Only detecting the attacks will not work, Snort should trigger some action that alerts the user about the attack. The GUI available for Snort should be comfortable for native users for configuration and running.

7. REFERENCES

[1] Fagg, C.R. 2001. Intrusion Detection Systems: Definition, Need and Challenges. White Paper. SANS Institute InfoSec Reading Room.

[2] Axelsson, S. March 2000. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report. Chalmers University, Sweden.

[3] Sobirey, M. November 1999. Michael Sobirey's Intrusion Detection Systems page. http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html.

[4] Timberline Technologies LLC. Alphabetical List of Intrusion Detection Products. http://www.timberlinetechnologies.com/products/intrusiondtct.html, [Last Updated 2009].

[5] Boyce, C.A.P. and Zincir-Heywood, A.N. October 2003. A Comparison of Four Intrusion Detection Systems for Secure E-Business. In Proceedings of the 6th International Conference on Electronic Commerce Research (ICECR03), ATSMA, IFIP, Dallas, USA.

[6] Varadarajan, G.K. October 2012. Web application attack analysis using bro ids. White Paper. SANS Institute InfoSec Reading Room.

[7] White, J.S. Fitzsimmonsb, T. and Matthews J. Quantitative Analysis of Intrusion Detection Systems: Snort and Suricata. In Proceedings of the SPIE, Cyber Sensing 2012

[8] Roesch, M. 1997. Snort – Lightweight Intrusion Detection for Networks. In Proceedings of the USENIX LISA'99 USA. November 1999.

[9] Inc. Network Flight Recorder. http://www.nfr.com, 1997

[10] Lee, W. Park, C. and Stolfo, S. 1999. Automated Intrusion Detection using NFR: Methods and Experiences. USENIX Intrusion Detection Workshop. California, USA.

[11] Undy, M. and Antonelli, C.J. 1998. Sifting the Network: Performing Packet Triage with NFR. CITI Technical Report 98−6. Center for Information Technology Integration, University of Michigan, USA.

[12] http://suricata-ids.org/features/all-features [last visited: 02/12/2013]

[13] Albin, E. 2011. A Comparative analysis of the Snort and Suricata Intrusion Detection Systems. Master's Thesis, Naval Postgraduate School, Monterey, California, USA.

[14] Neumann, P. G. 1990. A Comparative Anatomy of Computer System/Network Anomaly Detection Systems, CSL, SRI BN-168, Menlo Park, CA, USA.

[15] Enterasys Networks. http://www.voxtechnologies.com/enterasys_files/pdf/overview-datasheet.pdf [last visited: 04/12/2013]

[16] BackTrack Linux. Available at: http://www.backtrack-linux.org

[17] Wire Shark Packet Analyzer. Available at: http://www.wireshark.org

[18] Social-Engineer Toolkit (SET). Available at: https://www.trustedsec.com/