ISSN:0975-9646

A Comparative Study of Open Source Network Based Intrusion Detection Systems

Yakuta Tayyebi, D.S. Bhilare

ILVA Commerce & Science College, SCSIT DAVV

tayyebiyakuta@gmail.com

bhilare@hotmail.com

Indore, M.P, India

***Abstract*—With the advancement in network based technology Cloud Computing is gaining more and more popularity as a huge number of enterprise applications and data are using cloud or Network based platforms. These services use the Internet, networking protocols, different management tools and programming language. Providing security is a vital feature required from these network management tools. Distributed nature of cloud computing, makes it prone to various attacks and intrusions. A variety of techniques are available to help in detecting and/or preventing such attacks. Network Intrusion Detection Systems (NIDS) provide an effective technique to detect and prevent intrusions. The NIDS are mainly designed to protect the availability, confidentiality and integrity of network system. Market provides a number of open sources and commercial Intrusion Detection Systems to choose from according to the enterprises requirements. In this paper we analyse some well know open source IDS tools for features, general working behaviour, utility and performance so that an organization can choose the appropriate tool as per their requirements.**

***Keywords*— Intrusion detection system, HIDS, NIDS, Open Source IDS.**

I. INTRODUCTION

Cloud based systems are more prone to intrusions and attacks due to their distributed nature. A huge variety of tools are available in market to secure network configurations. The commonly used tools are the firewall and the intrusion detection/prevention system (IDS/IDPS). Firewalls are based on predefined policies which can check incoming and outgoing traffic, and act as a barrier between secure and untrusted networks. On the other hand an intrusion occurs when an attacker attempts to gain entry into the normal operations of a system, with the intent to do harm. [1]. Intrusion detection system is a type of security system for that gathers and analyses information from various areas within a network to identify security issues and act accordingly. It aims at identifying malicious activity such as denial of service attacks, port scans, Cross Site Scripting or even attempts to crack into system by monitoring network traffic. Deployment of a robust and reliable network intrusion detection system (IDS) to maintain availability, confidentiality and integrity of a network is very importance. An IDS/IDPS can be used effectively to detect suspicious activities within the network and can take measures to log and/or block these attacks.

In general there are two main detection techniques for IDS i.e. Signature-based and anomaly-based. Signature-based techniques monitor the behaviour of machine or network and compare it with the characteristics of known attacks i.e. “signatures”. If a match is found the attack is reported and further action is taken. Signature based IDS have high detection rates for well-known attacks. We can add new signatures without modifying existing ones as and when required. However, they fail to detect unknown attacks. In anomaly-based detection techniques, normal state of system or network will be defined. If behaviour of network does not match the normal behaviour criteria, the IDS will flag it as abnormal or attack. Using this method will increase the probability of detecting unknown attacks. However, it makes lot of detection errors as defining the normal state itself is very complicated.

Intrusion Detection Systems can be implemented as a hardware or software. The software IDS is more configurable, affordable and easy to update as compared the hardware based. A lot of commercial and open source software-based IDS are available in the market. Since most of the commercial IDS are very expensive and have a significant resource requirement these are not affordable to use. Open Source IDS are increasingly being used as they offer benefits and ease in implementation to offer higher levels of security and protection free of cost. There are several open source Network based IDS such as Snort, Bro**,** Suricata etc. available in market. In the paper we will compare the features Snort, Suricata and Bro IDS are offering to the users.

The paper is organized as follows: Section II, III and IV provide an overview of general working behaviour of Snort, Bro and suricata IDS respectively. Section V compares them based on different parameters and features. Finally in section VI conclusion is provided.

II. SNORT

Snort is an open source network intrusion detection and prevention tool created by Martin Roesch in 1998. It is developed and maintained by SourceFire which was later acquired by Cisco. The main advantage of using Snort is its capability to perform real-time traffic analysis and packet logging on networks. It uses set of rules to check for hostile packets in the network and then generate alerts to the network administrator. Its engine combines the benefits of signatures and anomaly-based inspection technique and has become the most widely deployed IDS [2].Snort can be configured to run in three modes as per the need.

A. Snort configuration modes are as follows

1) Sniffer mode: In this mode IDS reads the data packets from the network and displays them in a continuous stream on the console.

2) Packet Logger mode: In this mode the IDS logs the read packets to the disk. The packets are logged in default output format i.e. ASCII text. If a more compact form of log are required for later analysis, binary mode of logging should be consider. Binary mode logs the packets in tcpdump format into a single binary file in the logging directory.

3) Network Intrusion Detection System (NIDS) mode: In this mode IDS performs detection and analysis of network traffic. This is the most complex and configurable mode. This will apply the rules included in the snort.conf file to each packet to decide if the packet is malicious. An action based upon the rule in the file is taken on detection of an attack.

B. Components of Snort

Snort comprises of logically separated components with each one having a defined role. These components work in synchronization to detect attacks and to generate output in a required format. A Snort-based IDS consists of the following major components:

1) Packet Decoder: The packet decoder takes packets from network interfaces and prepares the packets to be pre-processed or to be sent to the detection engine. The interfaces may be Ethernet, SLIP, PPP etc.

2) Pre-processors or Input Plug-ins: Pre-processors are plug-ins used with Snort to arrange or modify data packets before sending them to detection engine to find out if the packet is being used by an intruder. They are also used to normalize protocol headers, detect anomalies, packet reassembly and TCP stream re-assembly.

3) Detection Engine: The detection engine is responsible for detecting if any intrusion activity exists in a packet. The Snort rules are used for this purpose. The rules are checked against all packets. If a packet matches any rule, predefined action specified in rule is taken else the packet is dropped.
4) Logging and Alerting System: This system generates log and alert messages depending upon what the detection engine finds inside a packet.
5) Output Modules: Output modules process alerts and logs and generate final result for the user to access them in ways as needed (console, extern files, databases, etc.). [3]

Snort is supported on a number of operating systems and hardware platforms. Currently Snort can be implemented on the following operating systems: Windows, MacOS, Linux, OpenBSD, FreeBSD, NetBSD and Solaris. Thus Snort toolkit runs on any modern operating system and any hardware available. It helps to fix a number of network problems and intrusion detections. Its current limitation is that it is single-threaded, so it does not take advantage of multi-core machines. Snort Fails to detect fragmented packets at high speed networks (> 5Gbps). [4]

III. BRO

Bro is an open-source Network based IDS that monitors network traffic passively for suspicious activity and attacks. Bro is a flexible IDS developed by Vern Paxson in the Network Research Group at Lawrence Berkley National Lab, and by the International Computer Science Institute in 1998. Bro is combination of both signature and anomaly-based technique. The traffic captured is converted into a series of events by analysis engine. Bro can perform multi-layer analysis and behavioural monitoring. It performs Policy-based intrusion detection. [5]

Bro scans network traffic to extract application level semantics. Then event-oriented analysers executes that compares the activity with troublesome patterns. This analysis concentrates on detection of attacks defined by signatures or in terms of events and unusual activities if any. Bro IDS analyse the traffic in three phases. First it filters the traffic, discarding less important elements. The filtered traffic is passed to event engine, where the network packets structure is interpreted and abstracted into higher-level events describing the activity. Finally Policy Script Interpreter is executes on the stream of events, looking for possible intrusions that should generate alerts.

A. Components of Bro

The Bro IDS consist of the following major components:

1) Libpcap: Bro needs the API libpcap to capture packets from the network interfaces. Libpcap captures all the traffic that comes from the network layer and filters out the non-important elements. The filtered packet stream is forwarded to the Event engine.

2) Event Engine: It takes in the packets from the libpcap and puts them together in form of events explaining the performed actions. Event Engine is written in C++.

3) Policy Script Interpreter: The events generated by the Event Engine are compared with the policy scripts by Policy Script Interpreter. The events are sorted in a FIFO order which means the first one to come will be the first one to be processed. Specific action will be taken if event is characterised as malicious activity and discards the events, if not defined in the policy scripts. Events that seems like attacks but actually aren't (false negatives), can be detected at this point. If the policy scripts are good enough false negatives will be minimal. Policy Script Interpreter is written in Bro language.

Computer Networks using a Bro IDS requires a UNIX-like operating system. The system can run on Linux, Solaris distributions or FreeBSD. Bro is a high-speed (Gbps), high-volume intrusion detection system. It is majorly use by networks requiring high customization and flexible intrusion detection system. It has been developed mainly to provide a research platform for intrusion detection and analysis.

IV. Suricata

Suricata is a signature-based network IDS developed by the Open Information Security Foundation (OISF). The parser was written by Ivan Ristic of Mod Security fame for the OISF. It is a rule-based IDS with multi-threading capabilities. [6] It can use existing rule sets to monitor network traffic and provide alerts when detects suspicious events. The most commonly used are Emerging Threats, Emerging Threats Pro and Sourcefire's VRT. Suricata has powerful Lau scripting support for detection of complex threats. Suricata can also detect many anomalies in the traffic it inspects. It is designed to fit within existing network security components. Suricata works as a multithreaded engine. Suricata offers high speed and efficiency in network traffic analysis due to its multi-threaded design. It divides up the IDS workload based on the processing needs. Its multi-threaded architecture allows it to make optimum use of the multiprocessor architectures common in today's world.

Suricata has the same data flow as Snort in acquiring packets by first defragmenting them, then reassembling streams and finally normalizing application layer data and providing outputs for alerting and logging. The system is designed with multi-threading technique, allowing multiple-packet capture queues to each worker process that distributes the workload across multiple CPU cores available. The approach was implemented with the idea of distributing the task across multiple processors to support the ever increasing network throughput.

Suricata has the ability to work at the level 7 of the OSI model, which enhances its malwares detection ability. Some shortcomings of Suricata are that it doesn't accept some rules from VRT (Snort) due to incompatibilities and also suffers from lack of documentation .The support provided to user community is also less as compared to snort.

V. Comparition

Comparison of the three IDS is done on the basis of different parameters such as speed, signatures, flexibility, deployment, interface and operating system capability.

1) Speed: Bro and Suricata IDS have the ability to run in high-speed environments. They are very effective and are capable to capture data from Gbps networks. This makes them suitable for more large scale networks whereas Snort IDS is not able to run prefect in high speed networks without dropping packets or slowing down the traffic.
2) Signatures and Rules: Signatures used for detecting intrusions are more sophisticated in Bro as compared to the signatures used in Snort. Suricata utilizes externally developed rule sets. Suricata is capable of using the specialized Emerging Threats rule set and the VRT (Snort) rule set.
3) Flexibility and Customization: Bro is flexible in its configured. It comes with pre-written policy scripts which can be used directly to detect the most well-known attacks. User can customize policy scripts containing more rules if added features are required to detect newer attacks. Policy Scripts written in Bro Language can be customised to add new functionality. Snort on the contrary has very less provision for customization as per the user's requirement and is less flexible.
4) Deployment and Documentation: As compared to Snort system, Bro and Suricata are more difficult and time consuming to deploy, understand and work with. On the other hand snort is stable, easily configurable and well documented.
5) GUI Interface: Snort has a graphical user interface which makes it more popular. Bro's lack of a user interface (GUI) can also be considered as a disadvantage since one should have good knowledge of UNIX system and be able to handle shell commands to understand this system.
6) Operating System Support: The Snort supports most of the operating systems whereas Bro is confined to UNIX like operating systems.

TABLE I. Table of COMPARISION

Parameters	Open Source Tools		
	Snort	**Bro**	**Suricata**
Developer	Sourcefire, Inc.	National Science Foundation (NSF)	Open Information Security Foundation (OISF)
Multi-thread	No	No	Yes
Operating System Compatibility	Any	Unix like system	Any
Rules Support	VRT Snort rules SO rules Emerging Threats rules	Contextual Signatures	VRT Snort rules Emerging Threats rules
Installation /deployment	Installation also available from packages.	Manual installation	Manual installation.
User community	Large	Small	Small
Documentation	Well documented	Few resources	Few resources
GUI Support	A lot	Few	Few
High Network speed Support	Medium	High	High

Snort is the most widely used intrusion detection and prevention open source tool as its pre-processors are very efficient for reassembling fragmented packets. Nevertheless, Suricata is an upcoming IDS that could prove to be a revolution in detection techniques. Support for very high speed networks, scalability, IPv6 support, use of anomaly detection technique and scoring thresholds are some of the features of Suricata to add to its advantage. Snort is the ideal solution for a moderate traffic network, whereas for high throughput network systems with 10Gbps or more, Suricata is preferred due to its support for large scalability.

Bro could be considered as a high throughput research environment due to its great flexibility. [7] Snort is packet oriented whereas Bro is connection oriented. Its powerful scripting feature is definitely a greater advantage compared to the rule sets in Snort or Suricata.

VI. Conclusion

Security has been the major concern of any organisation using cloud services. Network security in any organisation can be achieved by using Intrusion detection tools to tackle several types of attacks. All open source IDS tools have their strengths and weaknesses that are the criteria in the selection of the appropriate solution for each organization. Users can chose and customise Open Source Intrusion Detection tools while installations as per the requirement of their organization.

References

[1] Michael E. Whitman. “Principles of Information Security”, 2012

[2] SNORT website, http://www.snort.org.

[3] Miguel A. Calvo Moya,” Analysis and Evaluation of the snort and bro network intrusion detection Systems” 2008.

[4] Tian Fu, “An Analysis of Packet Fragmentation Attacks vs. Snort Intrusion Detection System”, International Journal of Computer Engineering Science (IJCES), May 2012.

[5] BRO website, http://www.bro-ids.org, Bro intrusion detection system.

[6] Open information security foundation (OISF) website, http:// www.openinfosecfoundation.org.

[7] Surya Bhagavan Ambati, Deepti Vidyarthi, “A brief study and comparison of, open source intrusion detection system tools”, International Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-1, Issue-10, Dec 2013.

[8] George Khalil, “Open-Source IDS High-Performance Shootout”, SANS, June 2012.

[9] Yakuta Tayyebi, Dr.D.S. Bhilare “Cloud security through Intrusion Detection System (IDS): Review of Existing Solutions”, International Journal of Emerging Trends & Technology in Computer Science, ISSN 2278-6856, Volume- 4, Issue -6, Nov-Dec 2015.