International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 14 (2014), pp. 1429-1436

http://www. irphouse.com

Dynamic Rule Based Traffic Analysis in NIDS

Kuldeep Tomar[1], S. S. Tyagi[2], Rachna Gupta[3]

[1] *Department of Computer Science & Engineering, MRIU, Faridabad, Haryana*
[2] *Department of Computer Science & Engineering, MRIU, Faridabad, Haryana*
[3] *Department of Computer Science & Engineering, NGFCET, Palwal, Haryana*

Abstract

A set of rules are used to detect or prevent the hostile data traffic in a network. In dynamic computing environment, security challenges are increasing day by day. It is so important to detect the abnormal behavior in a packet. There are lots of techniques and tools are used to determine the suspicious behavior. In this paper we have used snort as an IDS engine to detect the signature of packet using rules. Any type of Virus, worm or say infected packet have some sort of signature.

These signature may be present either header part of a packet or in the option part in the term of a message. Network Intrusion Detection system uses a collection of signature to generate alert, scan alert and then decide to pass the packet or reject the packet. In this project we also have use number of tools like sguil, tcpdump, bro, argus etc. to show the collective report which is generated after applying rule in the snort base engine.

Keywords – NIDS, Snort, Rule, Session, Alert, TCP, UDP, ICMP

INTRODUCTION

Security is the major issue considered from the day one of Internet world. Internet reduce the burden of transmitting data from one place to another but add more security threats for saving the integrity of data from outside world worms like hackers, virus, spam, intruder etc. The IDS are installed in one or more nodes of the net- work, check each packet that is routed over this node, and also check the node for abnormal behaviour; like connection from disallowed networks, and unauthorised login. Each packet is classified whether it is malicious or not and the files, processes, and connections are observed for abnormal behaviour [3].

Network Intrusion Detection system (NIDS) [7] system which consist of several sensors/tools can produce better log generation and alert analysis using signature based traffic analysis. There are many IDS engine are available which provide the

facility to write rule. Like Bro, Suricata and Snort etc. In this project we have used snort as an IDS engine to generate rule.

Intrusion Detection Systems traditionally deal with getting the alert. In this paper we have uses NIDS which involves getting additional context along with the alert:

- Alert Data
- Session data Generation
- Statistical data Generation
- Full Content Data Collection
- Event Data Generation

IDS ENGINE

Snort

Snort is one of the widespread network Intrusion detection system developed by Source fire. Snort is open source intrusion detection system means that original source code is available to anyone at free of cost in this Configuration file (snort. conf) ties everything together. It will check packets passing through an interface beside "signature" or "rule" records.

Bro

Bro was developed by "Vern Paxson" with the team of researchers at the International Computer Science Institute (ICSI) in Berkeley and National Centre for super-computing Applications in Urbana Champaign [4]. The basic function of Bro is to monitor the network traffic in depth for signature for abnormal activity.

Features of Bro:

- Bro analyze high-performance networks and is used operationally at a variety of large sites.
- Bro derives with analyzers for numerous protocols, enabling high-level semantic analysis at the application layer [5].

Suricata

Suricata is a new open source Next Generation Intrusion detection and prevention system. It is developed by Open Information Security Foundation (OISF) which is a non-profit foundation supported by the US Department of Homeland Security (DHS) and a number of private companies[4]. As compare to other IDS Suricata operates by fetching one packet at a time from the network. The fetched packet is preprocessed and then fed to the core engine that runs the detection algorithms to check if the packet is normal or malicious. Based on the judgment, the packet is either one accepted or rejected [6].

Characteristics of Suricata:

- High scalability & More Efficient
- File identification and file extraction

RULES

Snort and OSSEC have a large number of rule sets. These rules set needs to be tuned to reduce the number of false positive. NIDS sensor working with Snort rules to alert on a network. Writing rules becomes most important and arguably most difficult part of the network security monitoring.

Snort Rule

Snort rules are powerful, flexible and relatively easy to write. Snort rules are divided into two sections, rule header and rule body. Rule Header contains the information about the action of rule takes. It consists of the section of the rule before starting parentheses and has many parts.

Action	Protocol	Source IP address	Source Port	Direction	Destination IP address	Destination Port

Fig. 1 Basic Structure of rule header

The optional part contains information about the packet which is used to generate an alert message. It also contains the information for matching a rule against data packets.

Action Parameter:

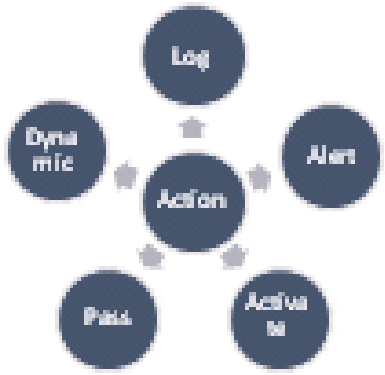


Fig. 2 Action Types

Pass Rule:

In this rule the action which is performed by the snort is to ignore the packet and pass the packet without performing any action. The pass rule plays an important share in such a case.

Rule to pass the traffic from source IP:

```
Pass tcp 192. 168. 2. 0 any -> any any (msg :" ignore redundant packet";)
```

Log Rule:
log rules in network policies which are being optimized is not an ideal situation. Log rules require creating log records which must be written on disc system or sent to log server. And those services are time and performance expensive.

When administrators create redundant rules which will accept or reject some network traffic, it is clear that those rules are not necessary for network policy. But when administrators create redundant log rules, it is possible that they didn't matches any packet that goes through the network[2].

Rule to accept network traffic:	Rule to reject network traffic:
log tcp any 80 -> any any (content:" server"; msg: "HTTP part";)	log tcp ! any 80 -> any any (content:" server"; msg: "HTTP part";)

Alert Rule:
In the case of Alert rule it generate an selected alert method and after notifying any packet that have matched to our rule find the whole log information about that packet.

Rule to generate an alert on network traffic:

```
alert tcp 192. 168. 2. 0 80 -> any any (msg: "TTL=100"; ttl: 100;)
```

Activate Rule/Dynamic Rule:
These both rule are interrelated with each other. Here first rule activate the rule. Activate rule is just like an alert rule but also add a dynamic rule for a specific event or situation. The condition which is mentioned in the dynamic rule is performed in a special case of activate rule

Rule to Activate run time action:

```
activate tcp Home_Net any ->192. 168. 1. 1/24 444 (flags:PA; activates:1; msg:"buffer overflow";)
dynamic tcp Home_Net any -> 192. 168. 1. 1/24 444(activates:1;count:100;)
```

There are 3 more type of action which is also used and these action are called intrusion prevention action because these action are applied to prevent our data packet from external world.

Drop Rule:
In this rule we can block the data packet and also save the whole log information about the packet.

Reject Rule:
In this rule we firstly have to block the packet save the detailed log information and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.

Sdrop:
In this action we just perform a block action not contain the log record.

EXPERIMENTAL DATA PACKET ANALYSIS

In this experimental study we have used Snort engine to capture the differ type of data packet which is traversed in the network environment

Fig. 3 Data Packet Types

Full Content Data:
It means to represent data traffic on the wire via radio frequency. Libpcap packet capture library is the standard library for reading full packet. It support to save captured packet in a file and then analyse it using the network tester or network monitor tools.

There are 4 tools are available which is used to store the content of entire packet.

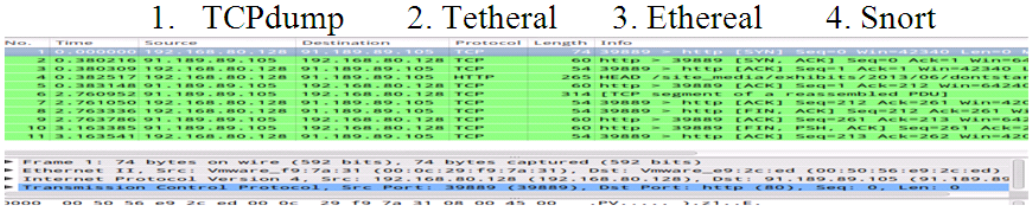


Figure 4. Full Content Data

Session Data:
Snort IDS can easily identify suspicious and malicious alert by inspecting network traffic or flow between two parties. And this representation of data is known as session data.

Rule:Log tcp 192. 168. 100. 1 any -> 192. 168. 200. 1 110 (session: printable ;)

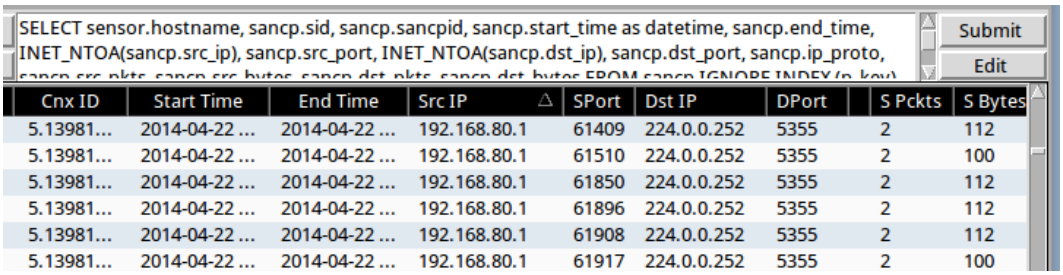
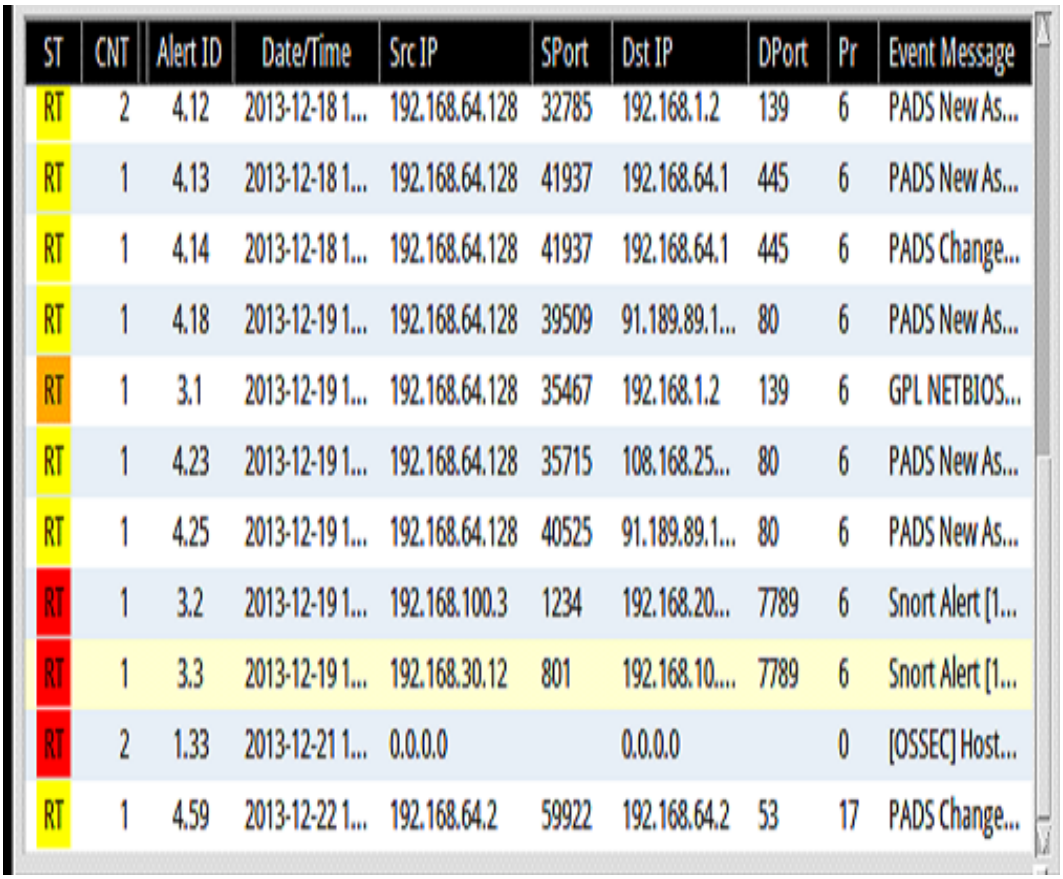


Figure 5. Session Data

Statistical Data:
Statistical Data play an important role for getting network activity. We can find thousands of packets manually at a time with the help of this pattern. Based on network traffic statistics, a decision is made regarding whether or not there is a need to rearrange the filtering rule and /or Rule field orders [1].

Alert Data:
To analyse the alert from differ IDS engine firstly we have to classify the data format on which we have applied alert. Alert can generate on any layer. The main focus to find the alert data in security mechanism for capture the faulted packet.

Alert tcp any-> any 7789(msg:"snort alert";)



ST	CNT	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	2	4.12	2013-12-18 1...	192.168.64.128	32785	192.168.1.2	139	6	PADS New As...
RT	1	4.13	2013-12-18 1...	192.168.64.128	41937	192.168.64.1	445	6	PADS New As...
RT	1	4.14	2013-12-18 1...	192.168.64.128	41937	192.168.64.1	445	6	PADS Change...
RT	1	4.18	2013-12-19 1...	192.168.64.128	39509	91.189.89.1...	80	6	PADS New As...
RT	1	3.1	2013-12-19 1...	192.168.64.128	35467	192.168.1.2	139	6	GPL NETBIOS...
RT	1	4.23	2013-12-19 1...	192.168.64.128	35715	108.168.25...	80	6	PADS New As...
RT	1	4.25	2013-12-19 1...	192.168.64.128	40525	91.189.89.1...	80	6	PADS New As...
RT	1	3.2	2013-12-19 1...	192.168.100.3	1234	192.168.20...	7789	6	Snort Alert [1...
RT	1	3.3	2013-12-19 1...	192.168.30.12	801	192.168.10....	7789	6	Snort Alert [1...
RT	2	1.33	2013-12-21 1...	0.0.0.0		0.0.0.0		0	[OSSEC] Host...
RT	1	4.59	2013-12-22 1...	192.168.64.2	59922	192.168.64.2	53	17	PADS Change...

Fig. 6 Alert Data Generate

Event Data Generation:

Snort is an event generation detection engine. Event generation in IDS allow us to collect and evaluate data in real time. By detecting the alert it will automatically responding to critical events in real-time. Event handler help us to detect and prevent the possible attacks.

Event handling data perform the number of tasks:

- Detect unwanted logon activity
- Monitor the environment from a particular console

CONCLUSION & FUTURE WORK

The idea behind the security is different by different people. We have many counter mechanism and tools are available for network security. But the main focus is that how can we implement and use the feature of security mechanism for capturing a faulted packet in a very short period. Real time Anti-Virus, Digital signature, GAIDS,

Packet Tracer etc. each have a unique feature in the field of network security. In this paper we have used a snort for rule generation which is capable to provide the differ type of activities on the data packet. (Example: Trace the packet, pass the packet, deny a packet etc.).

Apart from the work done towards the suggested design, my future work is to develop a window based NIDS application because window is more generally used by the people and then they face many type of counter problem.

REFERENCE

[1] Zouheir Trabelsi, Liren Zhang, Safaa Zeidan, Kilani Ghoudi, " *Dynamic traffic awareness statistical model for firewall performance enhancement*".

[2] Tihomir Katic, Predrag Pale Faculty of Electrical Engineering and Computing University of Zagreb Unska "*Optimization of Firewall Rules*".

[3] Igino Corona, Giorgio Giacinto, Fabio Roli, "*Adversarial attacks against intrusion detection system: Taxonomy, solutions and open issues*".

[4] Mauno Pihelgas Master Thesis on "*A Comparative Analysis of Open Source Intrusion Detection system* "2012.

[5] https://www. bro. org/

[6] Thesis "*Intrusion Detection and Prevention System: CGI Attacks* ".

[7] N. Stakhanova, S. Basu, J. Wong, "*Taxonomy of Intrusion response systems"*, International Journal of Information security 1 (2007) 169-184.