Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Developing rules or Signatures to Detect Novel Attacks using open Source IDS: Snort

Sumiti Bansal*
GGITC, Ambala
India

Mandeep Singh
SJP Polytechnic, Yamunanagar
India

Saurabh Mittal
GGITC, Ambala
India

Abstract --- Network intrusion detection along with firewall provides an important layer of security for computer system or network. Ids's main aim is to acknowledge suspicious or unauthorized activity and alert a system administrator about this activity. Snort is a freeware and open source NIDS tool which is basically a rule-driven system. Our aim is to identify a way in which snort could be developed further by generating user-defined rules to identify new novel attacks. The main aim of this research paper is to provide a brief overview of basic concepts required in developing user defined rules/signatures in Snort IDS.

Keywords — network, snort, signatures, rules, traffic, intrusion.

I. INTRODUCTION

Snort: Snort[1] is a libpcab based packet sniffer and logger tool that can be extended to a network based intrusion detection system, uses a set of signatures called rules to define what actually constitutes an attack. Snort signatures are updated on the snort website, usually several times a day. Snort is flexible in terms of its utilization as can be seen in figure 1. Previously logged packet can be used as input to snort exactly in the same way as live network data. The output thus generated can be logged to file or database in the form of alert or a network traffic log of all received traffic can be created for later processing in the case of live traffic capture.

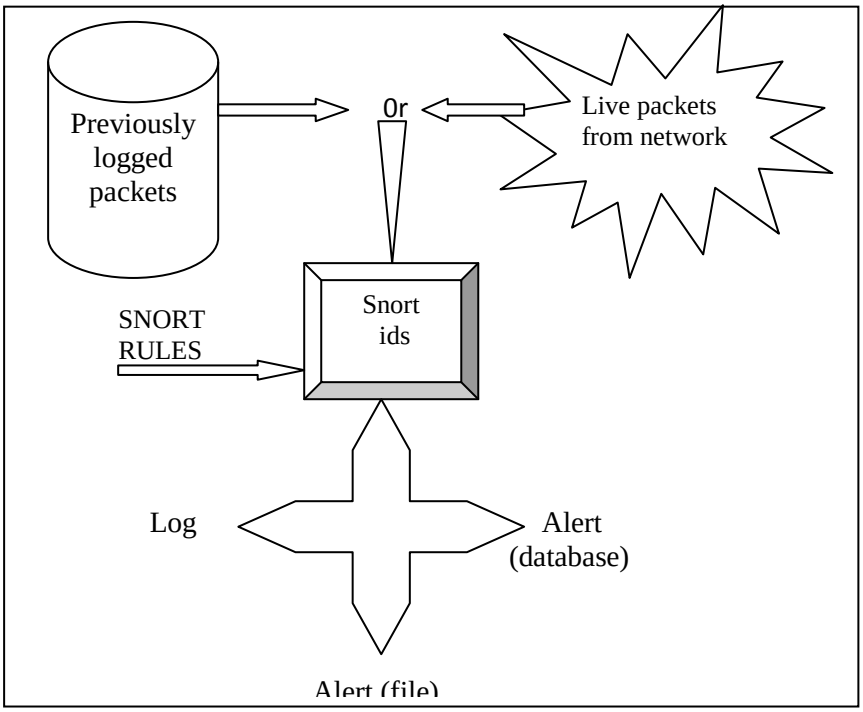

Fig. 1 Working of Snort

Snort rules: These are snort's signature sets [2], helps in identifying security violations. One of the best features of snort is its rule engine and language. Snort rule engine provides an extensive language enabling you to write your own new rules and extending it to meet the needs of your own network. Group of snort rules are referred to as a .rule file, each of which can be selectively included into snort configuration file "snort.conf". a rule file is a plain text file in which each line holds a separate rule. Example of a simple snort rule is shown below

Alert tcp any any →any any (msg:"snort alert" ;)

Fig. 2 Sample rule

II. WRITING SNORT RULES

Snort rule [3] are not only simple to write but are capable enough to detect a wide range of suspicious activities in the network. snort rule can be broken down into two basic parts:

- The rule header
- The rule option

A general form of a snort rule is shown in Fig 3.

<rule action > <protocol> <source ip> <source port> <direction> <dest. ip> <dest. port> <rule options>

Fig. 3 Format

Rule header:

It specifies the following:

- Rule action
- Protocol
- Source and destination address
- Source and destination port

Rule action: Tells snort what to do whenever a packet matches the rule criteria. There are five default actions available in snort or we can also define our own rule types.

Default rule action:

- Alert: An alert is generated using selected alert method and packet is logged
- Log: log the packet
- Pass : ignore the packet
- Activate: alert is generated while turning on another dynamic rule.
- Dynamic: remains silent until an active rule activates it.

New Rule action can be defined as an example given below. A new rule type to log to syslog and tcpdump is created..

```
ruletype dump
  {
    Type alert
   Output alert_syslog:| LOG_AUTH LOG_ALERT
   Output log_tcpdump: suspicious.log
  }
```

Protocol: There are various protocols that snort analyze for suspected behavior for example TCP, UDP, ICMP and IP.

IP addresses: when a packet comes in, its source and destination address are compared with the addresses defined in the particular rule of a rule set

Specifications for defining a IP address

- These addresses are created by a straight numeric ip address and CIDR block. for example, A combination of address/CIDR 192.168.1.0/24 would define the block address from 192.168.1.1 to 192.168.1.255
- Any IP address can be addressed using keyword "any"
- When negation operator is applied to an IP address, it asserts snort to match any IP addresses leaving the one indicated by the listed IP address. A "!" is used to indicate a negation operator
- List of IP addresses can also be specified.

Port numbers: These numbers are defined in many ways including any port, static port definitions, ranges and by negation. For example

- A unique port number can be described using static ports such as 23 for telnet
- Range operator is used to define port ranges i.e. A ":"
- Port negation is described by using a negation operator i.e.

Direction operator: Operator -> indicates the direction of the traffic on which rule is applied.

Rule option: this part specifies many attributes to check against. The semicolon (;) character is used to separate all snort rule options from each other. There are four major categories of rule options.

- General
- Payload
- Non payload
- Post detection

General rule options:

- Msg: describes the message to be print with the dump or alert.
- Reference: A reference to external attack identification system can be allowed.
- Gid: identifies part of snort that generates the event when a rule fires. The keyword is optional and if a rule has no entry for this field, it will default to 1.
- Sid: uniquely identify snort rules. This information helps output plug-in to identify rules easily.
- Rev: identify revisions of snort rules.
- Classtype: A rule is categorized as detecting an attack belonging to more general type of attack class.
- Priority : a severity level is assigned to rules
- Metadata: Additional information about the rule is embedded by rule writer.

Payload detection rule options

Some of the commonly used payload detection options are

- Content: packet payload is searched for specific content.
- Rawbytes: Raw packet data is looked by ignoring any decoding that was done by the preprocessor
- Depth: specify the part with in which snort should search the specified pattern.
- Offset: specify the start point from where snort should start searching a pattern.
- Distance: specify the length in the packet, snort should ignore before start searching a specified pattern.
- Uricontent: normalized request URI field is searched.
- Isdataat: justifies that the payload has data at a specified location.
- Pcre: perl compatible regular expressions is used in rule writing.
- Nocase: Case sensitivity is deactivated in a “content” rule
- Content-list: Multiple content strings are specified in the place of a single content option.

Non-Payload detection rule options

- Fragoffset: IP fragment offset field is compared against a decimal value.
- Ttl: checks the IP time to live.
- Tos: IP tos field is checked for a specific value.
- Id: checks the IP id field for a specific value.
- Ipopts: checks the presence of any IP option.
- Fragbits: checks whether reserved and fragmentation bits are marked in the IP header
- Dsize: payload size of the packet is tested.
- Flags: checks for specific TCP flag bits.
- Flow: Rule is applied to directions of the traffic flow
- Seq: TCP sequence number is checked for presence
- Ack: Specific TCP acknowledgment number is checked for presence.
- Window: checks the presence of specific TCP window size
- Itype: checks the value of ICMP type.
- Icode: checks the value of ICMP code field.
- Icmp_seq: checks the value of ICMP sequence field
- Icmp_id: checks for a ICMP ID value
- Ip_proto: checks against the IP protocol header.
- Sameip: checks whether destination IP matches the source IP.

Post detection rule options:

- Logto: Packets that matches the rule are logged to a specified output log file
- Session: User data is extracted from TCP sessions
- Resp: an active response is enabled killing the offending session.
- React: an active response is enabled that includes sending a web page or other content to the client and then closing the connection.

III. RULE DEVELOPMENT

Although there are official rule set available from snort website, still new rules are needed to counter new emerging attacks. Snort way of displaying data makes it a perfect tool for writing new rules. The basics for development consist of deploying the exploit of interest, such as a port scan, running the exploit on a test network with snort reporting all traffic between the attacker and target host and examining the data for a new signature and commencing the signature so developed into a rule.

Example1: Fig 4 shows the snort's view of "IAMP buffer overflow" [4]. The traced packet information is as follow:

- Date and time the packet was logged, i.e. in Fig 4 its value is 03/20 and in Fig.5 it shows 04/20.
- Source and destination IP address. these are numeric values i.e. in fig4 it is192.168.1.1
- Source and destination port numbers. it can take any value as shown in fig4 where it is 1034 and 143
- Protocol used in the packet. It take some specific value which is TCP in both packets shown below
- The TTL (time to live) field value in the IP header. The value of this field in both figures is 64.
- The TOS(type of service field value in the IP header. its value in both figure is 0x0.
- Sequence number of the packet, which is 0x5295B44E in fig.4 and 0x8538FD3A in fig.5
- Acknowledgment number of the packet has value 0x1B4F8970 in fig.4 and a value 0x0 in fig.5
- Window scaling option of the packet has value 0x7D78 in both figures.
- The remaining part is the data that follows the packet

03/20-22:27:58.403313 192.168.1.1:1034 -> 192.168.1.104:143 TCP TTL:64 TOS:0x0 DF
***PA* Seq: 0x5295B44E Ack: 0x1B4F8970 Win: 0x3908 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 EB 3B .....;5E 89 76 08 31 ED 31 C9 31 C0 88 6E 07 89 6E 0C ^.v.1.1.1..n..n……. B0 0B 89 F3 8D 6E 08 89 E9 8D 6E 0C 89 EA CD 80 .....n....n.....
31 DB 89 D8 40 CD 80 90 90 90 90 90 90 90 90 90 1...@..... 90 90 90 90 90 90 90 90 90 90 **E9 C0 FF FF FF** ..... 2F 62 69 6E 2F 73 68 90 90 90 90 90 90 90 90 90 /**bin/sh**.....

Fig. 4 Snort packet display

Using the above information a new signature can be developed easily such as the one shown below. Here the content of the packet is passed as value of content option. Based upon the match an alert is generated.

alert tcp any any -> 192.168.1.1 143 (content:"\|E9C0FFFFFF\|/bin/sh"; msg:"AlERT IMAP Buffer Overflow!";)

IMAP signature

Example2: Fig.5 shows the snort view of a "port-scan"[5]. The packet was generated by typing "nmap <IP address>", Setting the ACK field to 0 and sends a packet with the TCP ACK field set to examine if a network host is active.

04/20-13:49:29.765678 08:00:27:51:3F:56 -> 00:1E:40:40:92:E7 type:0x800 len:0x4A
192.168.1.8:45801 -> 192.168.1.1:4899 TCP TTL:64 TOS:0x0 ID:26429 IpLen:20 DgmLen:60 DF
******S* Seq: 0x8538FD3A **Ack: 0x0** Win: 0x3908 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 236039 0 NOP WS: 7

Fig. 5 snort view

Based upon the above information, a new rule is defined as shown below.

Alert tcp any any → 192.168.1.1 any (msg:"port scan detected"; flags: A; ack: 0 ;)

PORT SCAN signature

In general to develop rules [6][7], developer need to know the components of snort rule header, general rule options and basic snort rule options(i.e. protocol header field etc).
Table 1 shows some of the Rules we have developed in Snort based upon certain condition.

Table 1 Snort rules

Condition	Solution
How to send an Alert when there is a ping with ttl 254?	Alert tcp any any → any any (msg:"ping with ttl 254"; ttl:254;)
How to generate an Alert when a DF bit is set in an icmp packet?	Alert icmp any any →any any(msg:"alert don't fragment bit set";fragbits: D;)
How to generate an Alert when an ftp packet from any IP is sent to 192.168.1.1?	Alert IP any 21→192.168.1.1 any(msg: "packet is detected";)
How to generate an Alert when a syn-fin is detected in a packet?	Alert IP any any → any any(msg:"syn-fin detected";flags:SF;)
How to Log all traffic that belong to a particular application?	Alert tcp any any→any any(msg:"log created"; session: all;)
How to generate an Alert when there is an access to unauthorized sites?	Alert tcp any any<>192.168.1.0/24any(msg:"unauthorized access";content-list:"social"; react:block;) **CONTENT-LIST** # social sites www.twitter.com www.facebook.com #....
How to generate an Alert when a packet from telnet server contains the word "WHATSAPP"	Alert tcp 192.168.1.0/24 23 → any any(content: "whatsapp"; msg: "Detected whatsapp";)

IV. CONCLUSION

The paper describes the general form of rules that can be developed in Snort along with various options like how to make rules based upon content of packet, port number, IP address etc. Based upon the intrusion detected we can either send an alert message, log to an alert, pass a packet etc. The paper presented has provided solution to various conditions that can occur in the form of intrusion by developing user defined signatures. We can use and develop signatures to detect new kind of attacks as well if we know the pattern or content or source from which that attack takes place. Based upon the information we know we can also block that port or ip address from which the attack takes place and stop intrusion to attack our network. We have provided some signatures based upon the attack information. As now a day more and more new attacks are detected so future work can be to detect these new intrusions and develop signatures according to their patterns.

REFERENCES

[1] Martin Roesch ,"snort — Light weight intrusion detection for networks" Proceedings of LISA '99: 13th Systems Administration Conference. Seattle, Washington, USA, November 7–12, 1999
[2] How to write snort rules and keep your sanity, Available: http://paginas.fe.up.pt/~mgi98020/pgr/writing_snort_rules.htm
[3] Martin Roesch (2009), "Snort User Manual 2.8.5", available: http://www.snort.org/assets/125/snort_manual-2_8_5_1.pdf.
[4] Uwe Aickelin, Jamie Twycross and Thomas Hesketh-Roberts (xxxx) 'Rule Generalisation in Intrusion Detection Systems using SNORT', International Journal of Electronic Security and Digital Forensics (IJESDF), Vol. x, No. x, pp.xxx–xxx.
[5] How to use snort on backtrack4:basic example with a test attack,Available: http://insidetrust.blogspot.in/2010/12/how-to-use-snort-on-backtrack-4-basic.html
[6] Working with snort rules, chapter 3.copyrighted material. Pearson Education, Inc. All right reserved. Jinsheng Xu, Jinghua Zhang, Triveni Gadipalli, Xiaohong Yuan and Huiming Yu.learning "snort rules by capturing intrusions in

live network traffic replay", Proceedings of the 15th Colloquium for Information System Security Education Fairborn, Ohio June 13-15,2011

[7] Snort Cookbook/Rules and Signatures, available: http://commons.oreilly.com/wiki/index.php/Snort_Cookbook/Rules_and_Signatures

[8] The nma, Available: http://www.nmap.org

[9] R.rehman, "intrusion detection with snort" , upper saddle river:prentice hall,2003.

