Guidelines/Recommendations on Best Practices in Fine Tuning IDS Alarms

C.A.Obi and M.Papadaki

Centre for Information Security and Network Research,
University of Plymouth, Plymouth, United Kingdom
e-mail: info@cscan.org

Abstract

This paper presents guidelines/recommendations on best practices in fine tuning IDS alarms based on experiment conducted using the network based intrusion detection system Snort and MIT 1999 DARPA dataset. Snort generated about seventy seven percent false alerts. Experiment used fine tuning techniques namely: thresholding, rule customisation, rule disablement and combination of mentioned techniques, in order to achieve reduction in false alerts with minimal chances of missing true attacks. Evaluation of the tuning techniques led to the following guidelines put forward by this study: customised rule should be designed with context keyword which remains constant, threshold time periods should be set based on approximate time interval between successive alert instances, the limit threshold type is better suited to detect probe attack involving clear and stealth versions, technique combination improves attack detection rate with highly reduced false alarm instances.

Keywords

Intrusion Detection System (IDS), Snort, Fine Tuning.

1 Introduction

Reliance on the internet and other forms of network has led to increase in intrusions. Symantec in its Internet Security Threat Report Trends for January-June 2007 observed an alarming growth of Trojan attacks over the worldwide web (Symantec website, 2008). The Intrusion Detection System (IDS) was developed to complement the firewall in its fight against intrusions, and to enforce a defence in depth security approach. False alarms are usually the bane of the IDS (Cox and Gerg, 2004). They are alerts triggered by the IDS as a result of benign activities. It can be reduced using fine tuning. If the IDS is not properly tuned, could increase the risk of missing true attacks. This paper presents guidelines/recommendation on best practices in carrying out the technique of IDS fine tuning.

Section 2 presents existing research on IDS false alarm reduction while Section 3 is on the research procedures. In section 4, results from experiment are evaluated and analysed. Section 5 presents guidelines put forward by research on best practices in fine tuning IDS alarms while section 6 covers further work and conclusion.

2 Related Work

Law (2007) applied the use of data mining to IDS false alarms reduction. A false alarm engine built from false alarms produced from training the engine with attack free data was created. These false alarms generated were modelled as points in space within a time window, referred to as normal points. An alarm filtering engine referenced the false alarm modelling engine, using the K-nearest-neighbour (KNN) classifier to make its decision of whether data traffic was normal or abnormal traffic. KNN classifier measured the distance between the normal points and the new point (representing data traffic under observation). If the distance was below a certain set threshold, data traffic was flagged as false and filtered and if otherwise it was a true alert.

Abimbola et al, (2006) proposed a technique for false positive reduction in an HTTP data network using procedure analysis. Procedure analysis technique involved creating a data model from an HTTP 'GET' request. This HTTP data model divides the Uniform resource locator (URL) into its path component (path) and optional query string component (q). Harmful strings consistent to the HTTP request isolated from the HTTP data model, is used to design intrusive signature patterns.

The research conducted by law (2007) and Abimbola et al, (2006) used Data mining KNN classifier and procedure analysis techniques respectively to investigate false alarms reduction in contrast to fine tuning method used in this research to draw up appropriate guidelines/recommendations. Abimbola et al, (2006) were of opinion that Snort's increase in false positives was as a result of its detection rule options based on context keyword detection. As a result of the assertion made by Abimbola et al, this research made sure custom rules were designed based on keywords that were peculiar to the attack under observation. Keywords synonymous with attacks were selected based on careful observations of the attack patterns in the MIT 1999 DARPA dataset.

3 Research Procedure

Phase 1of this reasearch involved running Snort with all its rules enabled against the inside and outside tcpdump data of each day contained in the MIT 1999 DARPA dataset. This was modelled to represent an initial off the shelf IDS installation prone to generate numerous alert logs. This study assumed that the snort.conf settings: var HOME_NET and var EXTERNAL_NET have been correctly set since these variables are known to generate numerous false alarms if not properly defined (Greenwood, 2007). The generated logs were analysed to determine Snort's signatures which have raised false and true alert instances. To identify true and false alerts from each day of the experiment, the following approaches were utilised:

A. Alerts were correlated with the attack identification list released by MIT/DARPA/AFRL research team. An attack is considered true if it's time stamp, source and destination IP address and probably port numbers matched the attacks listed for that particular day on the list; otherwise it is a false alert

B. Information provided by Snort website on the rule responsible for the alert was compared with a database containing details (attack database, provided by MIT/DARPA/AFRL research team) on almost all the attacks in the 1999 DARPA dataset for a match. If there is a match, the alert is considered a true alert and if otherwise, a false alert.

C. Wireshark was used to analyse alerts generated for noticeable exploits peculiar to attack under investigation. If alert traffic pattern/payload indicates attack exploit, it was considered a true alert and if otherwise, a false alert.

Second phase (Phase 2) of this research depended on phase 1 results. Second phase involved comparison of different scenarios using various fine tuning techniques put forward by this study. Comparisons were based on two criteria namely: Tuning technique's ability to reduce false alerts and chances of increasing the risk of missing true attacks. Tuning techniques which offer great reduction in false alerts and minimizes the risk of missing attacks were adopted. From the outcome of this phase, guidelines/recommendations on fine tuning IDS alarms were proposed.

4 Experimentation Results and Analysis

4.1 Phase 1 Results and Analysis

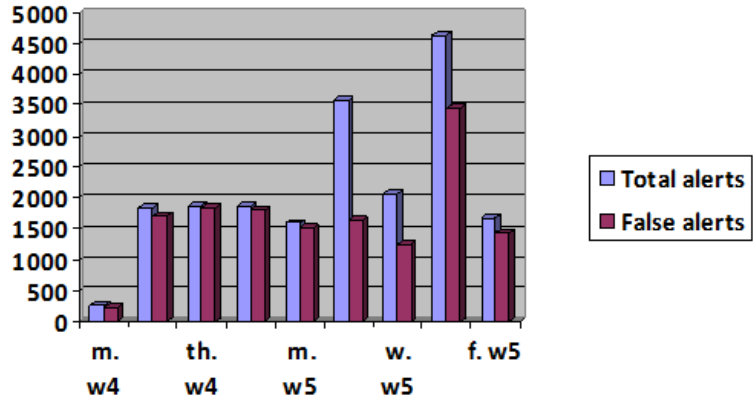

Figure 1. Total number of alerts and false alerts generated for each day in weeks 4 and 5 test data (inside tcpdump data).

Snort generated a total of nineteen thousand four hundred and thirty seven alerts. Fourteen thousand eight hundred and eighty four of the total alerts generated (over two weeks) were false alerts (figure 1).

4.2 Phase 2 Results and Analysis

From the outcome of phase 1 above, this section assesses the results achieved by implementing the various fine tuning techniques and strategy. Only signatures whose outcomes have contributed to this research work have been analysed herein.

4.2.1 ICMP PING signature

ICMP PING signature generated true alert instances for the ipsweep and POD attacks respectively. A custom tailored rule below was designed to detect the POD attack. Rule accurately alerted on all instances of the POD attack with no false alert observed.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"POD"; dsize:>64000;
reference:url,www.ll.mit.edu/mission/communications/ist/corpora/ideval/docs/attac
kDB.html; classtype:attempted-dos; sid:2000005; rev:1;)
```

Thresholding was applied to the ICMP PING signature with regards to the ipsweep attack (made up of clear and stealth versions). Scenarios 1, 2 and 3 were used to investigate the ideal threshold type for this probe attack using three hundred and forty three alerts (two hundred and sixty nine alert instances indicated the ipsweep attack) generated by signature from the outside tcpdump data of Friday week 4 .

Scenario 1: A threshold was designed instructing the ICMP PING signature to log one alert upon detection of six ICMP echo requests in sixty seconds (below).

```
threshold gen_id 1, sig_id 384, \
type both, track by_src, \
count 6, seconds 60
```

Thirty seven true alerts and six false alerts were generated by this rule. Threshold rule entirely missed out the stealth versions of the attack.

Scenario 2: Threshold count (scenario 1) was reduced to one, and its effect on signature detection for the stealth version of the attack observed. Rule generated six true alert instances indicating the stealth version of the ipsweep attack and seven false alert instances. It was observed that this scenario greatly reduced the value of true alert instances and missed all two hundred and fifty four instances of the clear version of this attack.

Scenario 3: A limit threshold was designed to alert on the first ICMP echo request in sixty seconds (below).

```
threshold gen_id 1, sig_id 384, \
type limit, track by_src, \
count 1, seconds 60
```

Threshold rule detected all stealth versions of the ipsweep attack, thirty seven true instances of the clear version of this attack and fifteen alerts were considered false alerts. This scenario showed that the limit threshold was the most ideal for probe attacks containing the stealth and clear versions together; none of the attack versions were missed by this threshold type.

4.2.2 INFO TELNET login incorrect signature

This research evaluated a total of thirty five alerts generated by this signature from the inside tcpdump data of Wednesday week 4. It was resolved to threshold the signature to generate an alert after three login failures. The alert instances for the two different but similar attacks (guesstelnet and the guest attacks) detected by signature were observed to have occurred under different time windows. A threshold set (below) instructing Snort to log an alert if four incorrect log in attempts in thirty seconds was detected entirely missed the guesstelnet attack, generating only a single alert indicating the guest attack. Increasing the time period to forty seconds, threshold rule detected six true alert instances indicating the guesstelnet and guest attacks respectively. This rule was adopted for this signature. Experiment observed that threshold time period influenced rule detection rate.

```
threshold gen_id 1, sig_id 718, \
   type both, track by_dst, \
      count 4, seconds 30
```

4.2.3 ATTACK-RESPONSES directory listing signature

The ATTACK-RESPONSES directory listing signature detected the most number of true attacks amidst false alerts from 'vol', 'dir', 'tree' commands issued during telnet sections. To fine tune signature, the research built custom rules for the respective attacks detected. Custom rule (below) for the yaga attack was designed to raise alert on initial attempt to hack the registry in order to add attacker to the Domain admins group. This custom rule generated an alert on the initial attack attempt (inside tcpdump of Tuesday and Thursday week 5). Ran against the inside tcpdump of Friday week 5 generated two alerts indicating the same attack.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 23(msg:"YAGA";
flow:to_server,established;content:"REGEDIT4";nocase;content:"domain
admins";nocase;reference:url,http://www.ll.mit.edu/mission/communications/ist/cor
pora/ideval/docs/attackDB.html;
reference:url,http://support.microsoft.com/kb/310516;classtype:attempted-
admin;sid:2000002;rev:1;)
```

To improve the alert quality, content modifier 'within' was introduced (below).

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 23(msg:"YAGA";
flow:to_server,established;content:"REGEDIT4";nocase;content:"domain
admins";nocase;within:170;reference:url,http://www.ll.mit.edu/mission/communicati
ons/ist/corpora/ideval/docs/attackDB.html;
reference:url,http://support.microsoft.com/kb/310516;classtype:attempted-
admin;sid:2000002;rev:1;)
```

The use of content modifier greatly improved the alert quality. Custom tailored rules were designed for the other attacks (casesen, sechole and netcat) detected by the ATTACK-RESPONSE directory signature. Custom rules successfully detected the

various attacks it was meant for without any observable false alert instances respectively. The research turned off the ATTACK-RESPONSE directory signature.

4.2.4 SHELLCODE x86 NOOP signature

Signature generated numerous false alerts from NETBIOS name query, HTTP 'GET' request for JPEG image and from base64 content encoding of legitimate emails; signature also alerted on true alert instances indicating the ppmacro, netcat and netbus attacks through the course of this research. Exploit codes of these attacks were sent as email attachments. Research used the netcat attack detected by signature to illustrate effect of designing keyword detection custom rules based on variable attack parameter.ie. parameter not peculiar to the attack. The custom tailored rule (below) was designed to alert on the context keyword y2ktest.exe (executing this email attachment launched the netcat exploit).

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"NETCAT";
flow:established;content:"y2ktest.exe";classtype:shellcode-detect; sid:2000007;
rev:1;)
```

Rule was meant to alert on the netcat initial attack attempt whenever it detects the 'y2ktest.exe' file. Rule ran against inside tcpdump data of Wednesday week 4 raised true alerts indicating the netcat attack but was observed to generate false alerts when ran against the inside tcpdump data of Friday and Monday week 4.

In order to fine tune the SHELLCODE x86 NOOP signature, focus was on the ppmacro and netbus attacks respectively since custom signature to detect the netcat exploit was designed on tuning the ATTACK-RESPONSE directory signature. Research considered three scenarios: the first involved setting a limit threshold type to raise an alert upon detection of an event in a second. This rule generated five hundred and fifteen false alerts and a true alert each indicating the ppmacro and netbus attack respectively (inside tcpdump data of Thursday week 4), the outcome was still very noisy. The second scenario involved the design of suppression rule to pass false alerts generated by this signature, but research observed that rule completely missed all true alert instances indicating the netbus and ppmacro attacks because machine IP addresses which were used to launch attacks shared same IP addresses amongst the IP addresses the rule was meant to pass. The third scenario was adopted, it involved the design of a custom (below) tailored rule to alert only on the detection of NOP strings 'AAAAAAA' found in email attachments.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SHELLCODE x86 NOOP";
flow:to_server,established;content:"AAAAAAAAAAAAAAAAAAAAAAAA";classtype:shellcod
e-detect; sid:2000010; rev:1;)
```

Rule comfortably detected all three alert instances each for the netbus and ppmacro attacks respectively. Custom rule ran against Tuesday week 4 outside tcpdump data was observed to raise four false alerts from a benign email attachment. Tuesday week 4 results showed rule was prone to false alerts; to put a check, research designed a limit threshold type to raise an alert upon detection of an event in a second in combination with custom rule. A combination of custom rule and threshold rule ran against the inside tcpdump data of Wednesday week 5 generated just two alerts. (SHELLCODE x86 signature originally generated four hundred and seventy

four alerts) indicating the netbus attack and the other was a false alert. Inside tcpdump data of Wednesday week 5 was used because it contained instances of the netbus attack and benign email attachments. Research adopted this tuning techniquecombination for the SHELLCODE x86 NOOP signature (original rule with SID 1394 was turned off).

4.2.5 CHAT IRC and the PORN BDSM signatures

The policy based CHAT IRC signatures and the PORN BDSM signature respectively have been disabled. This tuning technique decision was carried out by the research based on the loose policy of the MIT test evaluation (MIT website, 2008).

5 Guidelines/Recommendations

The following guidelines / recommendations on fine tuning have been put forward based on the findings of this research:

i. Custom tailored rules based on keyword detection should be designed with context keywords that remain constant and peculiar to the attack.

ii. When setting thresholds for probe attacks consisting of stealth and clear versions, the limit threshold type is better suited to detect instances of all attack versions.

iii. Combination of two tuning techniques improves attack detection with great reduction in number of false alerts.

iv. Threshold time period is of utmost importance. Improperly set time periods increase chances of missing attacks. Time periods should be set based on the approximate time interval between successive alert instances.

v. Rules are turned off only if they do not conform to the set policy of the network under guard or appropriate tuning measures have been put in place.

vi. Content modifiers should be used in design of custom tailored rules in order to improve alert quality.

6 Further work and Conclusion

6.1 Further work

Snort's pre-processors just like has been observed by Caswell et al (2007) have evolved so much since the inception of Snort; their functions are not restricted to anomaly detection and protocol normalization alone but also generate their own alerts. Through the course of this research, Snort pre-processors generated several alerts which could not be analysed due to time constraints. Investigation into alerts produced by this pre-processors are worth carrying further to determine if these alerts are false or actually true alerts and also a study could be carried out in order to

determine optimum tuning techniques for false alerts generated by Snort's pre-processors. Snort's respective pre-processors can be manually configured; if Snort can be designed to alert and drop protocols/data traffic which do not meet pre-processors set configurations before they traverse the detection engine, great reductions in false alerts and system processor overhead could be achieved.

6.2 Conclusion

Research effort was focused on proposing guidelines/recommendations on best practices in fine tuning IDS alarms. This study made use of fine tuning techniques namely: thresholding, custom rule design, rule disabling and combination of techniques aimed at false alert reduction with minimal risk of missing true attacks. This research work will be of benefit to the corporate (information technology personnel-network managers, administrators and support staff) as well as the academic world. It will alleviate the problems faced by Information technology personnel because it will save time spent on Intrusion detection system logs, improve device performance and justify cost on device investment. As regards the benefits to the academic world, future research in this area can hinge on the guidelines proposed by this study.

7 References

Abimbola, A., Munoz, J. and Buchanan, W., (2006). "Investigating False Positive Reduction in HTTP via Procedure Analysis. International conference on Networking and Services", [online], p 87-87. Available at: http://ieeexplore.ieee.org/iel5/11125/35640/01690558.pdf?temp=x&htry=1[accessed 2 August 2008].

Caswell, B., Beale, J. and Baker, A. (2007) "Snort IDS and IPS Toolkit". Burlington, MA: Syngress Publishing, Inc.

Cox, k.J. and Gerg, C.,(2004). Managing Security with Snort and IDS Tools. Sebastopol, CA: O'Reilly Media, Inc

Greenwood, B., (2007). Tuning an IDS/IPS From The Ground Up. Available at: http://www.sans.org/reading_room/whitepapers/detection/1896.php [accessed 15 June 2008]

Law, K., (2007). Reduction Of IDS False Alarms Using KNN Classifier. Available at: http://lbms03.cityu.edu.hk/theses/ftt/mphil-cs-b22180461f.pdf [accessed 3 August 2008].

Massachusetts Institute of Technology Website (2008). Available at: http://www.ll.mit.edu/IST/ideval/index.html [accessed 8 January 2008]

Symantec, (2007) "Internet Security Threat Report: Trends for January-June 2007". Available at: http://eval.symantec.com/mktginfo/enterprise/white_papers/ent-whitepaper_internet_security_threat_report_xii_emea_09_2007.en-us.pdf [accessed 25 August 2008]