Analysis and Evaluation of IDS Alerts on a Corporate Network

C.Rousseau, N.L.Clarke and B.V.Ghita

Network Research Group, University of Plymouth, Plymouth, United Kingdom
e-mail: info@cscan.org

Abstract

More and more, organizations rely on their network (Lundin Barse, 2004). This makes them vulnerable and the actual security means are no longer powerful enough. In order to bring more security than the traditional firewalls, IDS came out. Unfortunately, they do not bring the expected level of security. As they generate a lot of false positive, they tend to makes administrator of such systems turn them off. This paper then tries to analyze the cost effective of IDS for organizations. They today do not have the same means to face threats and vulnerabilities. If some companies are willing to invest a lot in security, some others are not. This research work has been based on the University of Plymouth network. It pointed out that IDS had to be properly configured in order to involve less investment for the administrators. But it also underlined that designers of such systems have to improve their effectiveness. Today, considering the investment that IDS represent, they do not seem cost effective enough to be used by all organizations.

Keywords

Intrusion Detection System, False Positives, Log analysis

1 Introduction

These last years, corporate networks have seen a huge increase in network threats. “During the first half of 2006, Symantec observed an average of 6,110 DoS attacks per day” (Symantec Website, 2006). Where many variants of attack have been created, number of malware has also increased. The last year, there was a growth of 48% in viruses, worms, Trojan, and spyware (Sophos, 2005). In 2006, 41,536 new threats have been detected by Sophos. The actual security tools corporate networks use are not powerful enough. Firewalls cannot handle threats alone anymore. FBI recently underlined that “98% of organizations use firewalls, but that 56% of them had still experienced unauthorized network access” (Porter, 2005). A few years ago, the goal of the attacks was only the proud. Most of them are now designed in order to cause economical impacts The Financial Insights estimated in 2006 that the lost would be “$400 million or more due to phishing shemes”. Universities are also the target of attackers. The University of Oxford has recently been hacked. Two students have been able to “find out anyone's email password, observe instant messenger conversations and control parts of the university's CCTV system” (Slashdot, 2004). A quite similar attack also happened in the University of California where students’ personal information have been stolen (Hines, 2005). If universities are today facing the same threats than companies, they don’t have the same means to face them. Indeed, universities do not have any security team to analyze generated events by security systems. Most of the time, their own law forbids them to monitor the traffic for confidential matters. Universities network are then more “open” and vulnerable to threats. Then for all these organisations, the need for security was obvious. Several security systems have come out but one has particularly attracted attention. Intrusion Detection Systems (IDSs) “inspect traffic inbound from the Internet for attacks against exposed Internet facing systems” (Stevenson, 2006). But as it is a quite new technology, IDS have some weaknesses in construction and configuration. They can sometimes generate much more than 1000 alerts per day. These alerts are, for most of them false positives (legitimate traffic tagged as attack by the system). This quite often compromises their effectiveness and makes the administrator of such system turn it off. But such system as presumed to be very powerful in attacks detection.

In order to test this effectiveness, this paper focuses of the efficiency of an IDS on the University of Plymouth campus network. It will first of all present a brief overview of the different IDS technology and will then present the methodology of the research. This will be followed by the findings of this research and a discussion of these results.

2 Overview of existing IDS

The IDS technology first started in 1987 with a generic IDS model presented by Dorothy Dening of the University of Georgetown. The model had to be independent from the environment in which it was evolving and its vulnerabilities. It also has to be independent of the types of intrusion.

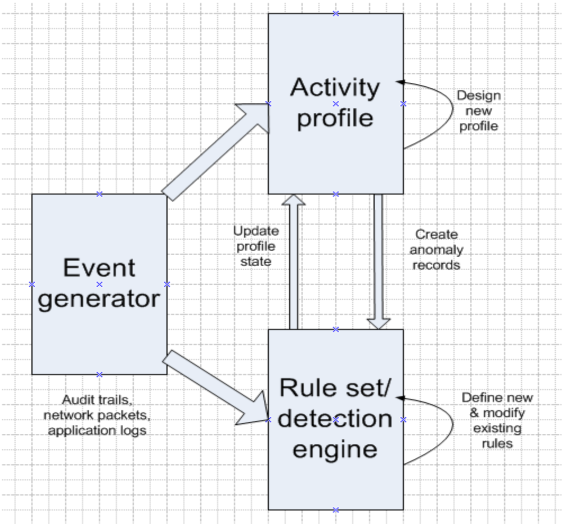

Activity profile
Design new profile
Update profile state
Create anomaly records
Event generator
Rule set/ detection engine
Audit trails, network packets, application logs
Define new & modify existing rules

Figure 1: A typical IDS model (Escamilla, 1998)

From this model many others have come out and have implemented more accurate functions. Today, IDSs use the CIDF model (Staniford-Chen, 1998). However they are all based on this basic one. Briefly, the "Event generator" monitors the traffic. The "Activity profile" contains variables of the network and "defines" the environment. The last module, the "Rule set/Detection Engine" represents the engine used by the IDS to detect intrusion. Many engines exist and represent different IDS. A brief overview of the different IDS algorithms is given in this part.

IDSs have to face two issues. They have to be able to detect well known attacks but also to anticipate future attacks. That is why many different algorithms have come out. Unfortunately, no one can deal correctly the both matters and all have some advantages and disadvantages. The most popular are anomaly detection and pattern-matching engines. The anomaly detection is based upon thresholds. Statistics on users' behaviors are made during a defined period in order to record their "normal" behaviors. From these results, thresholds are set up. They can represent many parameters concerning users, group, servers, and files. The anomaly detection engine adjusts its thresholds in order to automatically update behaviors. Once these thresholds set up, each time a behavior will go over one of them, an alert will be triggered. This system presents a main advantage: it is able to detect new attacks because every variant of attacks will obviously differ from the normal behaviors. But two major problems remain in this system. First of all, an attacker can slowly insert an attack behavior inside the system as it automatically updates its thresholds. Secondly, the system will maybe trigger a lot of false positives because a user often changes its behavior. Another statistical method has been created: the Heuristic-based analysis. This algorithm does not work upon statistics about user's behavior but upon the attack's behavior (CISCO System, 2002). It looks at the requests' behaviors inside the network and where they come from. This algorithm can sometimes be the only way to detect malicious activity in a network

The pattern-matching engine works differently. It contains signatures that basically define a known attack. By this way they theoretically only generate a low number of false positives as they recognize a known attack. But this system is extremely vulnerable to new attacks. It has to have a rule for each new attack that makes it slowing down. Then, the aim of these rule is to, by changing their structure, be able to detect new variants of attacks. The definition of the rule can then represents an event but also a sequence of events or regular expressions. To improve the efficiency of the pattern-matching algorithm, the stateful pattern matching has been brought out. This algorithm considers that an attack can be hidden in different packets. For example, the commands the attacker sends to execute malicious code can be divided into two different packets. A default pattern-matching algorithm would not recognize it because it deals packet by packet. By memorizing previous packets, this system deals with a stream of data and not with only one packet. If this system is not difficult to implement, it can generate a high number of false positives. Indeed, by considering data as a stream, it multiplies the probability of misdetecting an attack. To limit the high number of false positives that could be generated by the stateful pattern matching, the protocol decode-based analysis algorithms are based on the protocol fields. Where the previous algorithm looks for a pattern matching everywhere in the payload, this algorithm looks for a specific field of the protocol. By this way the detection of pattern matching is much more accurate. But the protocol has to be completely defined which is not always easy.

Many other systems do exist and try to implement advantages of two different systems. Emilie Lundin Barse (2004) cites some of them as example. RIPPER is based upon the anomaly and the pattern-matching detection. Briefly, it creates statistics of the previous data stored by data mining process. From these statistics and from the current intrusion, it defines rules. These rules fit much more the intrusions than hand created ones. A new type of IDS systems has also been brought out: the visualization systems. The Girardin's system (1999) is based upon a neural network and represents attacks as a map where the axes represent the different factors involved in the attack. The attack in then placed in this map according to the value of the different factors it represents. Erbacher and Frincke (2000) have created another visualization IDS. This one represents the entire network with nodes and links. The attacks are represented according to the different colors and different shapes that each node and link can take.

3 Methodology

In order to evaluate the need of an IDS for the University of Plymouth, an analysis of events has been made. When analyzing events, a methodology is essential. The methodology of this paper is based upon the incident handling procedure described in the Handbook for Computer Security Incident Response Teams (CSIRTs)" (West-Brown *et al.*, 2003). It describes the actions a CISRT has to take to deal with incident. Once an incident is opened for analyzing, an incident report has to be created. The incident report (or incident handling form) has been created upon the form described in the TERENA's Incident Object Description and Exchange Format Requirements (Arvidsson *et al.*, 2001). This report has to contain a tracking number to follow the event throughout the analysis process. It also contains the basic information about the event such as when it occurred, the attackers and the victims. Once these basic informations are collected, the incident report goes to a deeper analysis state. It is then important to define the depth of analysis. This last one depends upon different factors. It first of all depends on the team's mission and technical capabilities but also on the severity of the incident, the chance of repetition of the incident and the knowledge the analysis can bring to the team. The amount of data collected was too important to deeply analyze every event. In order to be able to provide a great analysis, the six most popular events (which represented more than 96% of the entire data) have been deeply analyzed. Then, all the high-priority incidents have also been deeply analyzed in order to judge the efficiency of the IDS on high-severity threats. The deep analysis consisted of analyzing particular day, hour, source IP address, destination IP address, and so on. When possible, a justification of each deep analysis has been made. Then the deeper analysis was able or not to classify the incident. Four classes of results were possible. The "false positive" class represented the incident considered as mostly false positives. The class "depends on IP" represented incidents for which some events were probably false positives but some were potential attacks. The third class called "potential attacks" represented incidents for which most of the generated events have been considered are potential attacks. Finally, the class "Unknown" represented the incidents for which it was not possible to give any hypothesis on their nature.

4 Results

To carry out this research, the data have been collected from the University of Plymouth network. It represented a common organization network and was then a relevant example for the cost effective of IDS for an "open" organization. The data have been collected two times. The first one was the 27th of March to the 11th of April and a second one from the 14th of June to the 23rd of June. TCPDump has been used to capture the traffic. Then Snort has been used as the intrusion detection system. Snort is a free signature-based NIDS (Network Intrusion Detection System). Scripts have been applied to the output alert file to anonymized the data

For the analysis of the different incidents, it has been presumed that a typical attack scenario was matching some essential criteria. First of all, the attempts occur grouped and are not spread out over time. Many other critera can be considered but mostly depend on the nature of the attack itself. Different source IP addresses can be used to launch an attack but no many different ones. Depending of the nature of the attacks, many or only one IP destination could be considered as a typical attack scenario.

As explained in the methodology, each event has been classified in one of the four categories. The chart below represents the classification of all the high-priority events analyzed.

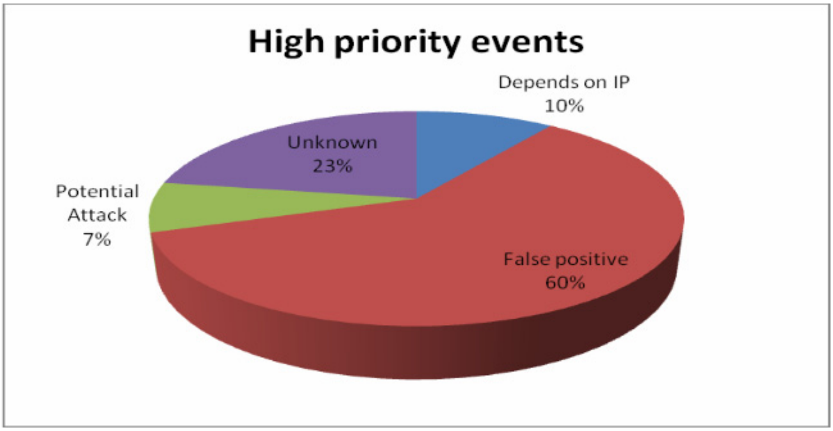
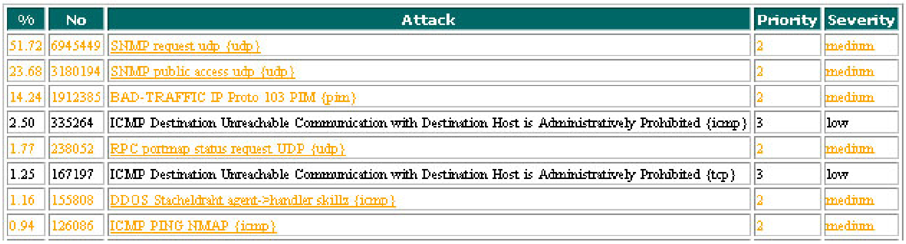


Figure 2: Classification of the high-priority events

More than the half analyzed events (60%) have been classified as false positives. Only 7% of these high-priority events have been classified as potential attacks. But quite a lot of events have not been classified and represent 23%.



%	No	Attack	Priority	Severity
51.72	6945449	SNMP request udp {udp}	2	medium
23.68	3180194	SNMP public access udp {udp}	2	medium
14.24	1912385	BAD-TRAFFIC IP Proto 103 PIM {pim}	2	medium
2.50	335264	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited {icmp}	3	low
1.77	238052	RPC portmap status request UDP {udp}	2	medium
1.25	167197	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited {tcp}	3	low
1.16	155808	DDOS Stacheldraht agent->handler skillz {icmp}	2	medium
0.94	126086	ICMP PING NMAP {icmp}	2	medium

Figure 3: The six most popular attacks

According to Figure 3, the six most popular incidents have been analyzed as well. Five of them are medium-priority and one is low-priority.

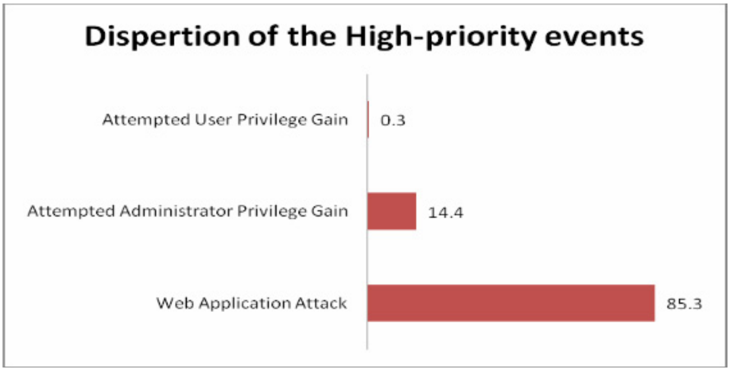


Figure 4: Dispersion of the high-priority events

Among the high-priority events, it was expected to find out a lot of injecting worms attempts. Surprisingly, only one has been potentially identified as so. For quite a few incidents, the evolution of the events over time looked the same as the global traffic. As it appeared that most of the high-priority events were web application attacks, the false positives hypothesis has been made in most cases. Indeed, attacks should not have any correlation with the global traffic otherwise it is obvious that legitimate traffic is “wrongly” tagged as attack.

The results of these analyses in addition to the results of the high-priority incidents gave the overall proportion of category class in the alert file. The graph below shows these proportions.

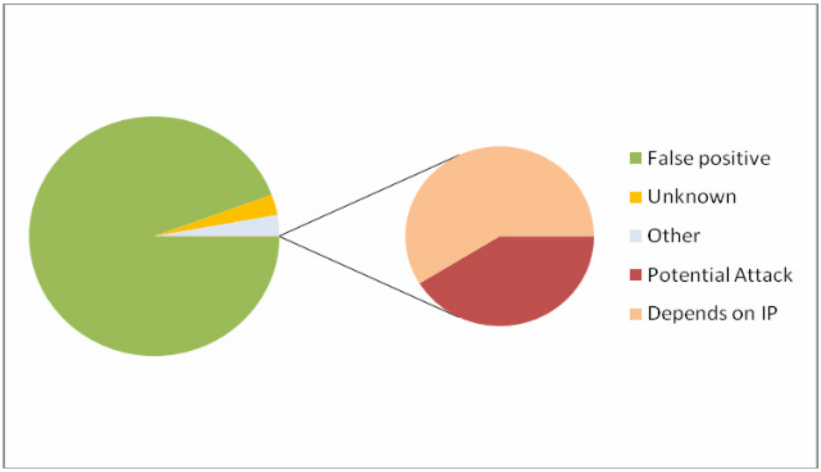


Figure 5: Dispersion of the events in the alert file

The number of potential attacks looks insignificant. But it still represents a number of potential attacks between 31 and 4300 per day by considering the “Unknown” events and the “Depends on IP” events. If it is not negligible in term of attacks attempts, it is in term of effectiveness. Indeed, five of the six of the most popular incidents have been classified as false positives. They are categorized as either medium or low priority and represent 94.8% of the entire alert file. In addition with the high-priority false positives, the entire proportion of false positives in the alert file represent more than 97%.

Several causes have been identified to these false positives. Many times it appeared the source or the destination IP addresses did not match the rule correctly. The rule that matches an attack can require an external or an internal IP address as source or as destination. Quite a few times the IP addresses involved in the generated events were incorrect according to the definition of the rule. It has been found out that the problem was in the operating system and the IDS configuration. The variables used by Snort could be substituted by the operating system variables, generating a lot of false positives.

It also appeared that the events generated by the http_inspect pre-processor of Snort have generated a lot of false positives. Briefly, the http_inspect pre-processor is a HTTP decoder implemented in Snort that can do the work that 1000 rules would do (Sturges, 2007). To work properly, the http_inspect pre-processor has to be accurately configured. The hypothesis of an incorrect configuration has been reinforced by the fact that similar incidents to those triggered by the pre-processor have also been triggered by a rule. Moreover this first version implemented in Snort does not handle the stateful processing. This can lead evasion attacks to bypass the system.

A few times, it appeared that the evolution of an incident over time looked surprisingly the same as the opposite of the global traffic. The number of events was slightly evolving to reach the highest in the middle of the night and to reach the lowest in the middle of the afternoon. Unfortunately the research did not come up with any hypothesis for that. A few times, a relationship between incidents have been seen and analyzed. Some incidents had a common source/destination IP address or generated events at the same time the same day. No certitude has been brought concerning a real link between these incidents. They could show a real attack as they could confirm a false positive hypothesis. However this way of research has shown a potential to bring more information concerning events.

5 Discussion

These results obviously show that too many false positives have been generated by the IDS. The proportion of false positives in the alert file represents an average of 142 763 false positives per day. These false positives are parasites for the quality of the collected data and make more difficult for administrators to find out attacks. Obviously the analysis of each incident could be deeper. To be more accurate, rather than analyze events as one incident, each event should be analyzed. However these results give a good overview of the composition of what the alert file likely is. The high-level priority events represent only 0.3% of all events and contain only 7% of potential attack. Even if 23% of events have not been classified as false positive or potential attack, this percentage is still low. In order to bring more reliability on the IDS, solutions have been proposed for most of the problems outlined in this work. This paper underlines the need for users of such system to configure it. They cannot do it properly without a good knowledge of the network and its need. Indeed, the resolving of IP address can bring answers on the legitimacy of traffic only if the potential communication of the University network with an external organization is well known. But this paper, across the http_inspect pre-processor, also showed that some weaknesses still remains in the design of IDS. So far, attackers have always had a step ahead the administrators and designers of such system. Security updates and patches come out after the attacker has already had the time to exploit the vulnerability. This main problem makes IDS focusing more on the detection of new variants of attack as quickly as possible. But by focusing on the efficiency, the effectiveness is maybe slowing down. This is maybe one main reason for the high rate of false positive generated.

6 Conclusion

From these results, the use of IDS seems to need a lot of investment. Their efficiency has to be much improved. The percentage of potential attacks detected represents indeed a real threat for organizations. But considering the huge number of parasite that can be generated in the log, it would cost a lot of time to really detect attacks. The percentage of false positives is definitively too high to show the effectiveness of such system in a network. However, a better configuration would definitively improve the quality of the alert file and could let think of a future with an IDS for every organizations. But so far the cost investment it represents is too much important. Only organizations that deal with high confidentiality data will be ready to invest a lot in that system. For organizations that do not have the same means, such as the University of Plymouth, it does not seem essential to set up an IDS.

7 References

Arvidsson, J., Cormack, A., Demchenko, Y., Meijer, J. (2001), “TERENA’s Incident Object Description and Exchange Format Recquirements”, http://www.ietf.org/rfc/rfc3067.txt (Accessed 11 July 2007)

CISCO System (2002), “The Science of IDS Attack Identification”, http://www.cisco.com/en/US/netsol/ns731/networking_solutions_white_paper09186a0080092334.shtml (Accessed 02 August 2007)

Dorothy Denning Web Site (1996), “An Intrusion-Detection Model”, http://www.cs.georgetown.edu/~denning/infosec/ids-model.rtf (Accessed 20 July 2007)

Erbacher, R.F. and Frincke, D. (2000) “Visualization in Detection of Intrusions and Misuse in Large Scale Networks”, http://www.cs.usu.edu/~erbacher/publications/SecurityVisPaper1-Mar2000.pdf (Accessed 03 May 2007)

Girardin, L. (1999), “An eye on network intruder - administrator shootouts”, http://www.usenix.org/publications/library/proceedings/detection99/full_papers/girardin/girardin.pdf (Accessed 09 August 2007)

Hines, M. (2005), “Security Threats”, http://news.zdnet.co.uk/security/0,1000000189,39192114,00.htm?r=2 (Accessed 08 August 2007)

Lundin Barse, E. (2004), “Logging for intrusion and fraud detection” http://www.cs.kau.se/~simone/Swits-IV/lundin.pdf (Accessed 08 August 2007)

Porter, T. (2005), “The Perils of Deep Packet Inspection”, http://www.securityfocus.com/infocus/1817 (Accessed 06 May 2007)

Slashdot Web Site (2004), “Oxford Students Hack University Network”,http://it.slashdot.org/article.pl?sid=04/07/16/021200(Accessed 08 August 2007)

Sophos, (2005), “Sophos Security Threat Management Report”, http://www.sophos.com/security/whitepapers/SophosSecurity2005-mmuk (Accessed 05 August 2007)

Staniford-Chen, S., Tung, B., and Schnackenberg, D. (1998),” The Common Intrusion Detection Framework (CIDF)”, http://gost.isi.edu/cidf/papers/cidf-isw.txt(Accessed 06 May 2007)

Stevenson, T. (2006), “Extrusion Detection: Security Monitoring for Internal Intrusions ”,http://www.windowsitlibrary.com/BookReviews/BookReview.cfm?BookReviewID=95(Accessed 20 July 2007)

Sturges, S. (2007), “HTTP Inspect”,http://www.snort.org/docs/snort_htmanuals/htmanual_2615/node61.html (Accessed 11 July 2007)

Symantec Web Site (2006), “News Release - Vulnerabilities in Desktop Applications and Use of Stealth Techniques”,http://www.symantec.com/en/hk/about/news/release/article.jsp?prid=20060926_01(Accessed 24 January 2007)