NETWORK-BASED INTRUSION DETECTION SYSTEMS EVALUATION THROUGH A SHORT TERM EXPERIMENTAL SCRIPT

Leonardo Lemes Fagundes, Luciano Paschoal Gaspary
Programa Interdisciplinar de Pós-Graduação em Computação Aplicada,Universidade do Vale do Rio dos Sinos
Av. Unisinos 950 – 93.022-000 – São Leopoldo, Brazil

Keywords: Security, intrusion detection systems, evaluation.

Abstract: Intrusion Detection Systems (IDSs) have become an essential component to improve security in networked environments. The increasing set of available IDSs has stimulated research projects that investigate means to assess them and to find out their strengths and limitations (in order to improve the IDSs themselves) and to assist the security manager in selecting the product that best suits specific requirements. Current approaches to do that (a) require the accomplishment of complex procedures that take too much time to be executed, (b) do not provide any systematic way of executing them, and (c) require, in general, specific knowledge of IDSs internal structure to be applied. In this paper we address these limitations by proposing a script to evaluate network-based IDSs regarding their detection capability, scalability and false positive rate. Two Intrusion Detection Systems, Snort and Firestorm, have been assessed to validate our approach.

1 INTRODUCTION

With the large scale use of Internet, the number of attacks against all kinds of organizations has increased considerably. Through the exploration of different types of vulnerabilities such as configuration flaws, implementation flaws and improper use of available resources, a universe of possible attacks emerge. Examples of such attacks go from port scanning, denial of service, connection hijacking to more sophisticated attacks, such as distributed denial of service, insertion and evasion. Aiming at minimizing an intruder's chances to obtain success in his/her activities, several protection mechanisms are used. Among these mechanisms are cryptography, digital certification, public key infrastructures, firewalls, authentication protocols and intrusion detection systems.

The IDSs represent an important monitoring technique, whose main function is to detect malicious actions, such as attack attempts and illegal access to information. There are several intrusion detection systems available in the market. Among these IDSs some that stand out are Snort (Roesch, 1999), Bro (Paxson, 1999), NFR (Network Flight Recorder) (NFR, 2001), Firestorm (Firestorm, 2001) and RealSecure (ISS, 1999). Considering the increasing number of IDSs, the identification of their strengths and limitations is essential, not only to stimulate specific research niches in the area (to improve the IDSs), but to assist the security managers in their grueling task of selecting the most appropriate system for the environment used as well.

Several approaches have been proposed to assess intrusion detection systems (Puketza et al., 1997; Lippmann et al., 1999; Alessandri, 2000; Barber, 2001). Those approaches, however, don't describe a systematic way of executing the procedures, presenting a series of exhausting activities that take several weeks and demand specific knowledge from the users, such as the internal structure of IDSs (which is not possible in case of proprietary IDSs).

This paper presents an alternative approach to evaluate network-based IDSs by presenting a script with a set of systematic procedures, which can be done in a short period of time and that do not require previous knowledge of the detection tools to be assessed. Despite the fact that the results of the evaluation don't present as many details as some approaches just mentioned, they can be easily carried out (without requiring highly specialized human resources and materials). The tests here evaluate the following IDS characteristics: detection capability, scalability and false positive rates. These characteristics were chosen for they are the most

Paschoal Gaspary L. and Lemes Fagundes L. (2004).
NETWORK-BASED INTRUSION DETECTION SYSTEMS EVALUATION THROUGH A SHORT TERM EXPERIMENTAL SCRIPT.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 54-60
DOI: 10.5220/0001399300540060

representative to an organization decision process when choosing an IDS. It is also worth mentioning that our approach has been proposed to comparatively access two or more intrusion detection systems (and not to measure the individual capabilities of a single system). The paper is organized as follows: section 2 presents some related work. Section 3 describes the script proposed to conduct the assessment. The results obtained with Snort and Firestorm IDSs are presented in section 4. Section 5 concludes the paper with some final remarks and presents perspectives for future work.

2 RELATED WORK

This section presents the main approaches developed up to this date to assess intrusion detection systems. The criteria applied in this comparison were: type of assessment, nature of background traffic generated to perform the experiments and requirement of test settings.

Regarding the type of assessment, it has been observed that most of the approaches assess only the IDSs' signatures bases (Puketza et al., 1997; Lippmann et al., 2000, Barber, 2001) which, in addition to being exhaustive work considering the size of these bases, generates a valid result only for a short time period, because signatures are developed by IDSs' creators or even by users quite frequently. Therefore, in order to consider the results of these approaches reliable, it is necessary to run all the experiments every time a new signature is released. On the other hand, approaches such as the ones proposed in this paper and in (Alessandri, 2000), instead of testing the signature base, actually test the IDS's detection capabilities. By using them, experiments must be re-run only when new detection functionalities are added to the system itself.

The representation of background traffic is a fundamental feature in the assessment of IDSs, because it interferes directly in the results of some tests, such as the ones on false positive rates and scalability. There are approaches like (Lippmann et al., 1998) and (Lippmann et al., 1999) that do not describe how background traffic is composed. This leads to the objection of results, especially in the assessment of false positives, because there is no way to assure if there are attacks inserted on this traffic, nor there are ways to identify which reasons might have led the IDSs to generate such results.

Except for the methodology proposed by (Alessandri, 2000), all the others require some sort of test setting to run the experiments. Approaches such as (Puketza et al., 1997) and (Lippmann et al., 1999) require complex test settings, with dozens of stations (attackers, victims, evaluated systems, traffic generators and traffic collectors), various interconnectivity equipments (hub, switch and routers), and even firewalls. These requirements in a test setting often result in an impracticable choice, due to the fact that they demand an extended time period and a dedicated environment up until completion of tests. Furthermore, the use of firewalls prevents several attacks from being captured by the IDSs, since they are blocked before reaching the company's internal network. The use of firewalls is extremely crucial for any business and must be part of every study which aims at assessing security infrastructure. However, considering this specific purpose, it is a factor that limits the assessment process.

As a general rule, what is observed in the approaches quoted is that they lack a proposal which could be applied by organizations security staff. In order to accomplish that, it is necessary to develop an approach that presents well defined procedures, that can be easily achievable, and that can fully reflect the reality of the criteria assessed. The approaches referenced here fail in these aspects. In addition to not providing adequate documentation on how some important tests were conducted, some of these proposals have not yet been properly validated, or do not offer the necessary means to be applied.

3 EVALUATION SCRIPT

The script is composed by five steps: selection of attacks, selection of tools, generation of traffic settings for evaluation, assembly of evaluation environment and IDSs analysis.

3.1 Selection of Attacks

The goal is to select a set of attacks that present distinct technical characteristics amongst them. Instead of simply gathering a set of attacks, we propose to select, by the end of this phase, attacks whose detection is possible through different existent mechanisms in an IDS. For instance, for an IDS to be capable to detect an insertion attack, the *URL Encoding*, it needs more than the capability of analyzing an HTTP packet, because a content decoding mechanism of the packet header is also necessary. Similarly, to detect denial of service attacks such as the *teardrop* a mechanism capable to rebuild fragmented IP packets is required. Thus, the attacks selected in this step present a unique set of characteristics that allows the evaluation of different

				HTTP			IP					TCP										ICMP		UDP
	Multiple packets	Does not establish a connection	Establishes a connection	Requisition line	Requisition encoding	Requisition size	Fragment off-set	IP packet with an enabled DF bit	Fragment identification	Raw IP packet	Fragmentation control flags	Options (NOP, MSS, Windows, ...)	Field: service type	Sequence number	TCT initial window	Packet with SYN flag	Packet with FIN flag	Packet with ACK flag	Packet with URG flag	Packet with PUSH flag	Packet with all flags disabled	ICMP packet	ICPM error message size	0 bytes UDP packet
TCP Connect	X		X																					
Syn scan	X	X														X								
Ack Scan	X	X																X						
Window Scan	X	X																X						
Fin scan	X	X															X							
UDP Scan	X	X																						X
Null Scan	X	X																			X			
Xmas	X	X															X	X	X	X				
TCP Ping	X	X																X						
TCP Fragmentation	X	X					X		X		X					X	X	X	X	X				
IP Scanning	X	X								X														
Fingerprinting	X	X						X				X	X	X	X		X	X					X	
Ident Reverse TCP	X		X															X						

Figure 1: Technical description of the initial attack setting proposed

IDSs detection capabilities and not only the signature database.

Figure 1 shows the characteristics explored by the set of possible attacks to be used in the IDSs evaluation (in columns). In the lines all attacks are listed. Due to space limitation, the figure illustrates only the port scanning attack, but evasion, insertion and denial of service attacks were also considered. The attacks used in the assessment are those that explore combinations of different characteristics (in bold). To ease the problem that the IDS does not have the signature of the selected attack and to avoid reaching a conclusion that the IDS is not capable to detect the attacks, we always selected two attacks that explore the same characteristics: one past and one recent. This approach allows reducing the initial attack setting. We assume, for instance, that if an IDS is capable to detect an *Ident Reverse TCP* attack, it should also be capable to detect a *TCPConnect* (that explores the same characteristics); it only needs to be configured with appropriate signatures to do so.

3.2 Selection of Tools

This step is dedicated to the obtention of tools that allow reproducing the attacks selected in the previous stage. This task can be accomplished in a short time period, since the tools applied are easy to find and use. For instance, to reproduce the port scannings listed in figure 1 the tool Nmap 2.54 (GNU/Linux) could be used.

3.3 Generation of the Evaluation Traffic Settings

The evaluation set up is formed by the selected attacks and by the background traffic (necessary for the scalability analysis). Following, we present the description proposed in the script (a) to store the attack traffic and (b) to generate the background traffic.

3.3.1 Gathering of the attack traffic

In order to avoid the manipulation of each attack tool every time that the several tests (presented in section 3.5) have to be performed, we suggest the previous gathering and storage of the attack traffic. In order to do that, an environment such as the one illustrated in figure 2a must be setup. In the *Attacker* station all attack tools selected in the tools selection step are installed, while the *Victim* station has all services to be attacked installed. Finally, the *Sniffer* station collects the traffic (using a tool such as tcpdump) generated by both the attack tools and the

attacked station (when, in some level, it presents a reaction to them). So, for each attack to be collected and stored, the following sequence of steps is suggested: (a) start up the tcpdump, (b) execute the attack, (c) stop the tcpdump and (d) store the traffic.

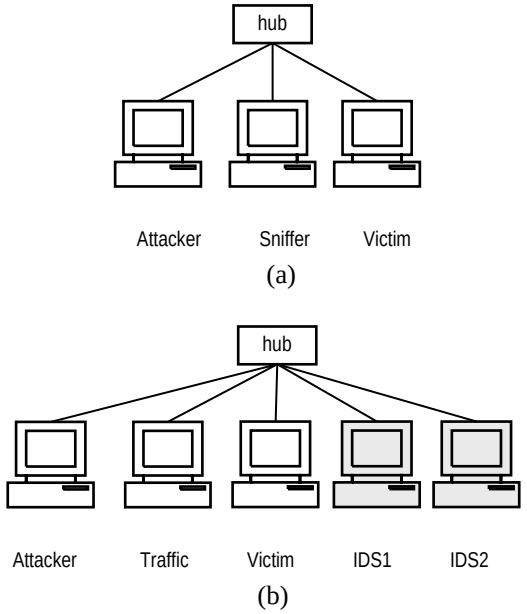

Figure 2: Network environment (a) to collect and store the traffic of attacks and (b) to evaluate the IDSs

3.3.2 Generation of background traffic

The background traffic is necessary in order to analyze the IDSs scalability. Our group decided to use uniform artificial background traffic in the analysis because when using real traffic, due to the throughput rate oscillation and the alternation of the applications used, it would be difficult to identify, for instance, at which rate the IDS starts to discard packets. Besides, the use of real traffic (unless the traffic was known in full detail), could lead to non-foreseen alarms (for instance, if there were any attacks inserted in that traffic), what would generate noise in the evaluation in course.

So we propose to use a background traffic composed by 256 bytes UDP packets. This traffic must be reproduced at different rates (e.g. 4, 6, 8, 10 and 12 Mbps). An appropriate tool to do so is `udp_generator`, developed at LAND/Federal University of Rio de Janeiro, since it is easy to manage and, for that, makes it unnecessary to store the traffic for subsequent reproduction.

3.4 Assembly of the Assessment Environment

The evaluation environment should be composed by the IDSs to be evaluated, the target stations (*Victims*) that will undergo the attacks, a station to reproduce the attack traffic (*Attacker*) and another to reproduce the background traffic (*Traffic*) during the scalability tests. Figure 2b shows the environment used in our evaluation, while the results are presented in section 4.

The number of victim stations and the operating system installed in those stations may vary according to the attacks selected to compose the evaluation setting. For instance, if the evaluation setting is comprised of attacks to Solaris and Windows 2000 Server stations, the network environment represented in figure 2b should use two additional victim stations, in which those systems should be properly installed and configured. Also, the amount of IDSs evaluated may vary. As consequence, the number of stations to host these systems can also be larger.

3.5 IDSs Analysis

As already mentioned, the script proposed evaluates the following IDSs characteristics: detection capability, scalability and false positive rates generated by these systems. Detection capability is the test that allows identifying the IDSs detection strengths, in other words, which types of attacks the system is able to detect. The scalability test allows identifying at which transmission rate the IDS starts to discard packets. The false positive rate shows the tendency of the IDS to generate false alarms. It occurs when normal traffic is erroneously considered an attack or when an attack takes place but the IDS generates alarms informing of other attacks instead of the one going on.

3.5.1 Detection capability

To evaluate the IDSs detection capability the following sequence of steps is suggested: (a) clean the log files and begin the IDSs log service, (b) reproduce, one by one, the traffic collected from the attacks (see section 3.3.1), (c) stop the log service, (d) save the generated files, and (e) count and identify the detected and non detected attacks (from the log analysis). The reproduction of the attacks will be triggered from the Attacker station, at a low rate (so the IDS does not discard packets), using a tool such as `tcpreplay`.

3.5.2 Scalability

For the scalability analysis results to be reliable, only the attacks detected by the IDSs in the previous analysis are reproduced. For example, if in an evaluation A five attacks are identified and an evaluation B detects only three, the scalability analysis is going to consider five and three attacks, respectively.

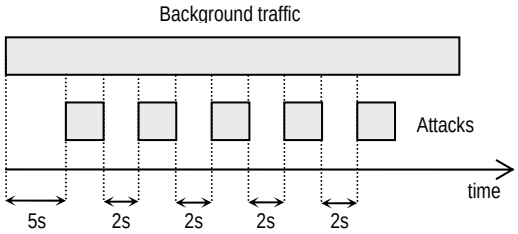

Figure 3: Traffic generation and reproduction sequence used for scalability analysis

Figure 3 represents the relationship between the attack traffic and the background traffic (uniform, generated at a constant rate), reproduced simultaneously in this analysis. Each type of attack considered in the evaluation (in our case, denial of service, evasion, insertion and port scanning) must be executed in a different series of tests. Figure 3, for example, refers to the scalability analysis of an IDS under the denial of service attacks. In this case, five attacks (Smurf, UDP Storm, Syn Flood, Teardrop and ICMP Fragmentation) are being reproduced in parallel with the background traffic (a). The attacks are executed one by one with a two second interval between them. The generation of background traffic starts five seconds before the first attack is reproduced and is only stopped after all attacks have been transmitted. So, as soon as that sequence has been terminated, (b) the *log* system must be stopped, (c) the number of alerts generated by the IDS should be stored (for later account), (d) the *log* system must be restarted and, soon afterwards, (e) the next type of attack must be reproduced. After every possible type of attack has been reproduced for each IDS, (f) the background traffic transmission rate must be increased and the procedure described must be repeated (a).

3.5.3 False positive rates

False positive is every alarm that indicates that an attack is happening, when actually another kind of activity is taking place. For example, when a support user executes a ping command for a server and the IDS stores this event as an attack. Another example occurs when the network is actually under one type of attack (e.g. *UDPstorm*), but the IDS generates alarms not only for this attack but for other type of attacks, that are not happening at the moment (e.g. *ICMP fragmentation*).

Our proposal to identify the false positive is based on the analysis of the *logs* generated in the detection capability tests. After a set of attacks, the IDS *log* stores the alarms. The security manager can easily sort the alarms that identify actual attacks and false positive alarms. The ratio between the number of additional alarms over the total of alarms generated for a set of attacks represents an important indicator of the IDS tendency to generate false positives. For that, the *log* files generated by the detection capability test should be checked again.

4 CASE STUDY

This section presents the results achieved by two IDSs submitted to our experimental script described in the previous section. The IDSs used in this case study were Snort 1.83 (Roesch, 1999) and Firestorm 0.4.6 (Firestorm, 2001), both available under GNU GPL version 2 license.

4.1 Detection Capability

The detection capability analysis was carried out based on the attacks mentioned in section 3.1. This analysis was made simultaneously with the two chosen IDSs. The results achieved by Snort and by Firestorm are shown in Figure 4. It is important to emphasize that, although the sequence of tests described in section 3.5.1 has been repeated ten times, the results were always the same (statistical variance equals zero).

The results demonstrate that Snort is a tool capable to detect insertion, evasion, port scanning and denial of service attacks very efficiently. Firestorm, on the other hand, did not present an efficient mechanism to decode HTTP requests.

4.2 Scalability

To evaluate the scalability, the IDSs were submitted to the procedure described in section 3.5.2, with background traffic of 4, 6 and 8 Mbps. The tests were carried out ten times and presented a 2.5% variance. It was observed that, at a transmission rate of 4 Mbps, the IDSs don't present packet loss. So, the number of alerts generated (including false positive) corresponds to the maximum possible for the set of attacks being analyzed.

	Snort	Firestorm
Evasion		
Method Matching	X	X
Session Splicing	X	X
Insertion		
Long URL	X	X
Self Reference	X	X
URL Encoding	X	
Port scanning		
UDP Scan	X	
Xmas	X	X
TCP Fragmentation	X	X
IP Protocol Sweeping	X	
Fingerprinting	X	X
Ident Reverse TCP	X	X
Denial of Service		
Smurf	X	X
UDP Storm	X	X
Syn Flood	X	X
Teardrop	X	X
ICPM Fragmentation	X	X

Obs.: The "x" means that the respective IDS detected the attack. If the space is blank it means that the IDS did not detect the attack.

Figure 4: Detection capability analysis results

In figure 5, the results obtained with the IDS evaluation are presented. As it can be observed, Snort is slightly superior compared to Firestorm regarding the scalability analysis as to all attack types considered. However, both IDSs can fail to detect attacks even at a low transmission rate (8 Mbps).

	Evasion	Insertion	Port Scanning	Denial of Service
4 Mbps				
Snort	100%	100%	100%	100%
Firestorm	100%	100%	100%	100%
6 Mbps				
Snort	98,81%	97,86%	99,32%	99,83%
Firestorm	97,56%	95,18%	99,57%	99,36%
8 Mbps				
Snort	89,99%	86,10%	94,85%	94,41%
Firestorm	86,04%	83,42%	90,49%	92,24%

Obs.: The value in each cell is obtained by dividing the number of *logs* stored (including false positive) by the number of maximum alarms expected.

Figure 5: Scalability analysis results

4.3 False Positive Rate

The false positive rate analysis, as described in section 3.5.3, is done by using the *log* file data generated by the IDS when executing a detection capability analysis. The sequence of tests described in section 3.5.3 was carried out ten times, always obtaining the same results (statistical variance equals zero).

Figure 6 shows the false positive rates generated by Snort and by Firestorm. The high values indicate that the IDSs has an imprecise set of signatures, what causes the mistaken generation of alarms. The table demonstrates that this matter is more critical in Firestorm, although the results presented for Snort are not encouraging as well.

	Evasion	Insertion	Port Scanning	Denial of Service
Snort	36,73%	29,97%	4,71%	4,63%
Firestorm	41,32%	42,97%	12,93%	7,47%

Obs.: The value in each cell is obtained by dividing the number of additional alarms (false positives) by the total of alarms generated.

Figure 6: False positive rate analysis

5 CONCLUSIONS AND FUTURE WORK

Nowadays the main approaches regarding IDS assessment use a large amount of reproduced attacks, because it is believed that this allows the evaluation to be more detailed. However, there aren't any preexistent criteria for the selection of the attacks. Therefore, many of them explore the same characteristics making a wide and detailed evaluation of the IDS strengths impossible. Besides, the selection of attacks leads to extremely exhausting experiments given the amount of attacks that they analyze. The script proposed in this paper describes a method for attack selection, in which the setting used in the IDSs analysis is composed only by attacks that present unique characteristics. Through this selection, described in section 3.1, the initial attack setting is reduced by approximately 50%.

Regarding the IDSs scalability evaluation, it can be said that the sustained capacity of the system being evaluated is very clear, even though the script is based on a uniform traffic rate. Another aspect is that even when submitted to low traffic, the IDSs begin to discard packets (compromising the

detection process). It still remains to be done an extended scalability evaluation of those IDSs for rates higher than 10 Mbps (ex: up to 100 Mbps).

The evaluation of false positive rates generated, as proposed in this paper, is influenced by the power of description languages to describe signatures and by the precision of the network manager when specifying them. An additional mechanism for this analysis, that takes into account the typical background traffic found in the organization where the IDS is going to be used, requires further investigation.

Future work include (a) the extension of the evaluation setting, through the selection of other types of attacks, (b) the investigation of procedures to evaluate other criteria (ex: capacity to handle concurrent attacks) and (c) the development of a tool to assist the execution of the script proposed.

REFERENCES

Alessandri, D. (2000). Using rule-based activity to evaluate intrusion - detection systems. In *Third International Workshop on Recent Advances in Intrusion Detection (RAID)*, pages 183-196.

Barber, R. (2001). The evolution of intrusion detection systems – the next step. *Computer & Security*, 20(2):132-145.

Firestorm (2001). *Firestorm network intrusion detection system Homepage*. http://www.scaramanga.com.uk/.

ISS (1999). *Real Secure Systems Inc. Homepage*. http://iss.net.

Lippmann, R., Fried, David J., Graf, I., Haines, Joshua W., Kendall, Kristopher R., McClung, D., Weber, D., Webster, Seth E., Wyschogrod, D., Cunningham, Robert K. and Zissman, M. (1998). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX) 2000, IEEE Computer Society Press, Los Alaminos, CA.

Lippmann, R., Haines, D., Fried, D. J., Das, K. J., and Korba, J. (1999). Evaluating intrusion detection systems the 1999 darpa off-line intrusion detection evaluation. *Computer Networks*, 34 (4):579-595.

NFR (2001). *Network Flight Recorder, Inc. Homepage*. http://www.nfr.com/.

Paxson, V. (1999). Bro a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435-2463.

Puketza, N., Chung, M., Olsson, R. A., and Mukherjee, B. (1997). A software platform for testing intrusion detection systems. *IEEE Software*, 14(5):43-51.

Roesch, M. (1999). Snort – lightweight intrusion detection for networks. In *USENIX LISA Conference*.