Evading IDS Detection

P. Jarmak and M. Papadaki

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

Abstract

Intrusion Detection Systems (IDS) is software capable of monitoring incoming and outgoing traffic. The project is to provide a benchmarking study in order to find a trade-off between performance and level of detection as well as to show how it is easy to evade an IDS. This publication describes the evasion techniques, the structure of the experiments as well as the trade-off between performance and level of detection. The results show the necessity of several pre-processors, the resources required by the IDS to guarantee a high level of detection as well as advice to configure Snort.

Keywords

Intrusion Detection System (IDS), Evasion Techniques, Snort, Pre-processors

1 Introduction

During the last years, multiple threats such as new virus or worms have been discovered. An IDS is become essential and should be used in every network where private and critical data flow.

Furthermore, new mechanisms called evasion techniques are able to bypass the security settled by the IDS. Indeed the new evasion techniques consist of combining multiple basic techniques used by the past in order to create new ones. The problem is that the available IDS software is usually not able to detect them. According to Stonesoft (2011) we have only seen "*the tip of the iceberg and over 90% of that iceberg is still unexplored. The theory and practice of evasion techniques needs to proceed hand-in-hand with leaps, not steps*".

In this paper, we focus on the mechanisms deployed by the IDSs to detect the threats but also on a benchmarking study in order to provide results of the resources required as well as the level of detection possible if configured with cautious and reliability.

The paper is divided in three sections. Section two is going to explain the background of the project such as the evasion techniques, Intrusion Detection System (IDS). Section three is going to discuss about the experiments. Therefore the methodology as well as the results will be presented. Finally, Section four is a conclusion of the research and discuss about the aims achieved.

2 Background

2.1 Intrusion Detection System (IDS)

IDS or Intrusion Detection System "*is the process of monitoring computers or networks for unauthorized entrance, activity, or file modification*" (Innella *et al.*, 2001).

Snort is one example of open-source IDS. It provides the basic techniques of threats detection. Snort is able to detect and alert an administrator when an intrusion is detected (Roesch *et al.*, 2011).

2.2 Evasion techniques

It is the ability "*to fool the IDS into seeing data different from what the target host will see*" (Oh *et al.*, 2007). Therefore the IDS system and the host will see different things. There are three kinds of evasion techniques:

- Insertion: The IDS "*makes the mistake of believing that the end-system has accepted and processed the packet when it actually hasn't*" (Ptacek & Newsham, 1998). It is what we call Insertion.

- Evasion: The IDS rejects a packet that the host accepts. It is what we call Evasion. As the previous techniques the IDS and the host will not see the same things.

- Denial of Service: This attack consists to overwhelm the IDS. Thus, it cannot detect the attack contained in a packet. In order to launch the kind of attack the attacker usually uses different hosts, generally a botnet. It is a computer infected by a program. According to Microsoft (2011) a botnet is "*malicious software (also known as malware) that can turn your computer into a bot (also known as a zombie). When this occurs, your computer can perform automated tasks over the Internet, without you knowing it*".

Example of basic evading techniques:

- String matching, IDS can search a specific pattern in a packet. However this technique is very weak because it is easily to change the pattern without changed the meaning. For sample the IDS searches the pattern /etc/passwd but the packet contains /etc/rc.d/.../passwd. The two patterns have the same meaning but not the same words.

- Fragmentation attack which consists of fragmenting the piece of code containing the attack in several packets.

- Fragmentation overlap, this technique consists to send a packet followed by a second packet which overwrites a part of the first one.

- Bad Header fields (No IP addresses set or header length too small, bad "DF" (Don't fragment) flag…)

- End system fragmentation bugs depending on the Operating system, the packet will not be reconstructed into the same way

Source: Ptacek & Newsham, 1998; Timm, 2006

As treated in the introduction, the evasion techniques evolved much faster than the security community thought. Stonesoft (2011) found new kind of evasion techniques. They are completely different than the precedent evasion techniques. They call it, Advanced Evasion Techniques (AET). The mechanisms to detect them are completely different and the techniques used to detect the other evasion techniques do not work. Therefore it is really important than the techniques evolve fast to be able to handle these new threats.

3 Experiments

The experiments conducted, have been divided in two objectives, one to show of how it is easy to evade an IDS and the roles of the pre-processors and another one to find the trade-off between performance and level of detection.

3.1 Influence of the pre-processors

In this experience, one configuration file of Snort has been used. That configuration file has been modified to identify the different threats (Scan, vulnerability scan, exploits, etc.). Then for different dataset such as scan, exploits, vulnerability scanner, Snort has been launched with different modification in its configuration file. Indeed some of the pre-processors have been deactivated to show their influences. These conditions have been reproduced for each dataset tested. Here, one example of the result obtained:

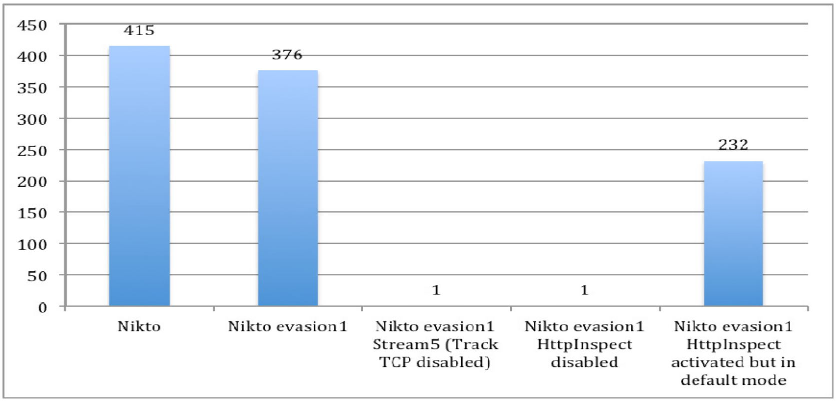

Figure 1: Influence pre-processors on vulnerability Scan detection (URL encoding evasion)

As we can see, the configuration of the pre-processors is essential in the detection. One modification change completely the number of alert found. Of course the modifications have been made to show the influence. Therefore the modifications have been done on the pre-processors essential for the detection of the evasion techniques.

One conclusion can be drawn for these experiments, is that some of the pre-processors are essential for the detection of the threats and without them it is straightforward to fool Snort. These pre-processors are Frag3, Stream5, HttpInspect, Sfportscan. Basically they contribute to provide the mechanisms capable of handling the defragmentation or tracking the connection as well as detecting the evasion techniques. These mechanisms are able to detect most of the evasion techniques which usually use fragmentation, delay, etc.

3.2 Trade-off between performance and level of detection

A second experiment has been done to observe the performance and level of detection according to three different configurations of Snort. These configurations are:

- A modified configuration according to the Snort manual to optimize Snort
- A configuration provided by the Snort website
- A configuration where all the options are by default

These configurations have been tested on a "test-file" regrouping several threats such as scans, vulnerability scanner, exploits, etc. Different parameters have been measured such as the CPU usage, the memory used, the number of alert detected and the time spent of analysing the file.

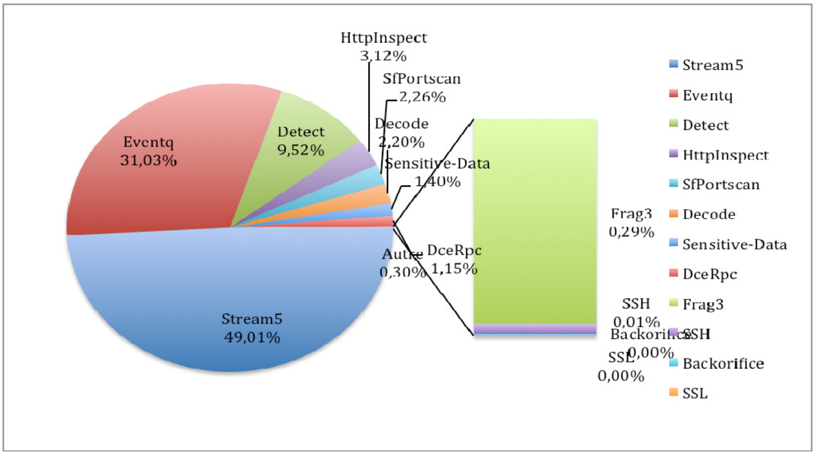

Figure 2: Percentage of time spent for each pre-processor

As we can see, three pre-processors take almost all the resources during the detection step. There are Stream5, Eventq, Detect and I can add also HttpInspect. These pre-processors take the majority of the resources and times because there are the pre-processor able to deal with the evasion techniques.

Moreover the pre-processors Eventq is responsible to generate the alerts. Therefore the way of how the alert are generated should be optimized in order not to take too much time in generating the logs and alerts even if it is a critical part for a later analyse by the administrator. Another solution could be to generate the alerts by another module speratated of Snort. That kind of software already exists such as Barnyard (Securixlive, 2012) which allows unifying the alert as well as improving the time spent to generate the alert. This allows Snort to focus on the detection of intrusion.

The graphs below show the trade-off between performance and level of detection. It shows the results of the three previous configuration files according to different parameters such as the number of alerts, the time spent and the memory usage. A table has also been provided to be more readable:

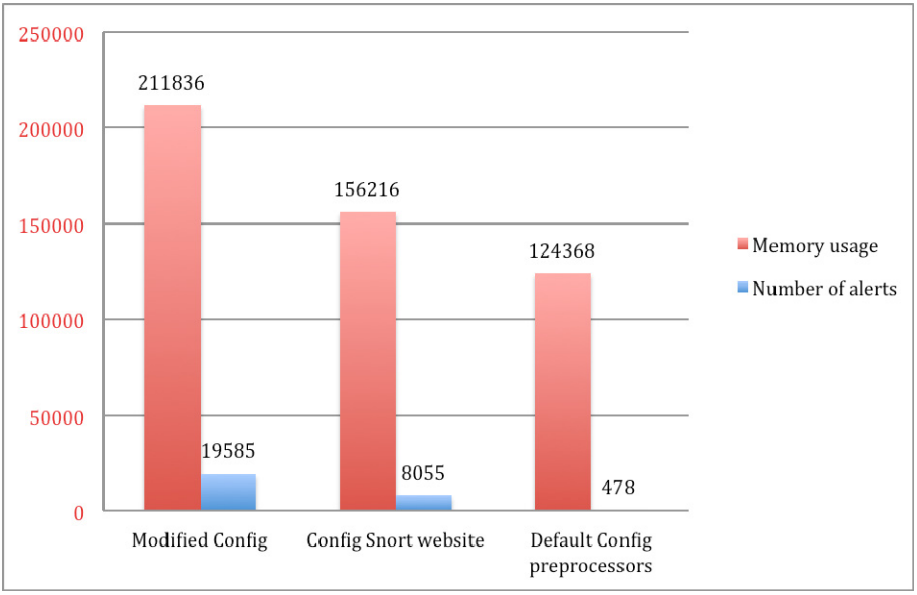

Figure 3: Memory usage versus number of alerts

As shown in the table and the graphs, the modified configuration provides better results in terms of detecting the intrusions but required as well more resources. However the resources stay reasonable. The level of detection is more accurate for the detection of the suspicious traffic.

Indeed the modified configuration file takes more time because it contains more options activated which required more time when Snort checks a packet.

Furthermore because Snort generates more alerts it spends more time to analyse the packets and write the logs.

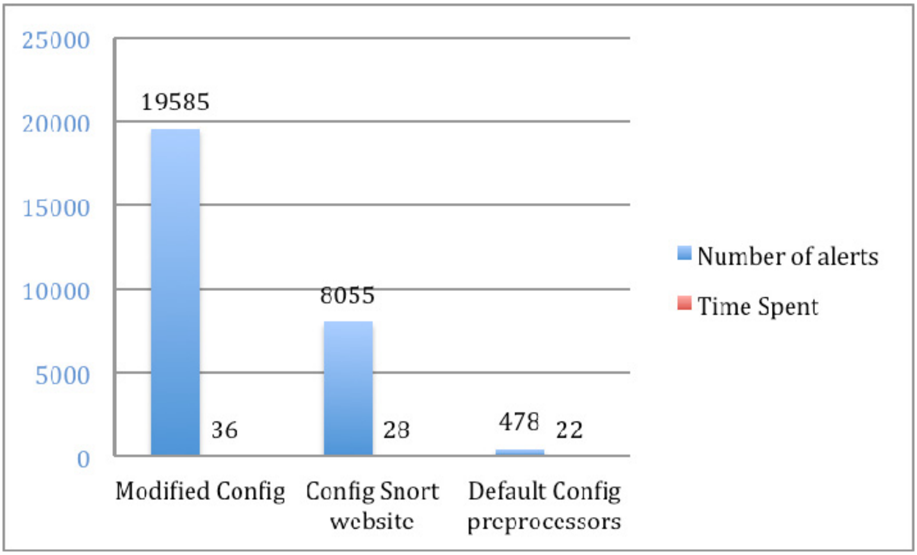

Figure 4: Time spent versus number of alerts

The table shows also the importance of properly configure the IDS. The default configuration found in the Snort website even if it is able to detect some of the threats, it is not enough to secure correctly the network. The same conclusions can be drawn with the third configuration. A default configuration should not be used, because it is not optimized and specific for a particular network. Moreover because a default configuration aims to be used by a large number of users, it contains a lot of flaws and therefore the network is not secured and easily fooled.

The experiments conducted show the importance of an IDS. Evasion techniques are something natural nowadays and therefore should be taken seriously. The paper has shown that Snort is able to detect the majority of them. However this level of detection requires resources and time. These requirements could be too high for certain network or company. Indeed a huge company with a high bandwidth, thousand of hosts, dozens of software could be asked enormous resources. Therefore a network monitor has to be optimized in order to be able to do its work and avoid being overwhelmed which could mean missed intrusion or threats. This is the last thing a company want. It could be interesting to conduct these experiments in a busy environment to number the resources and performance of an IDS.

The paper focused on the configuration of the pre-processors. Another lead could be to optimize the number of rules as well as the way to write them. Indeed too many rules will slow down the IDS. Furthermore a rule poorly written could cause a misunderstanding from the administrator who will read the log. A rule poorly written could cause as well false positives. These false positives may participate as well to slow down Snort if they are too many.

4 Conclusion

The research tried to investigate the resilience of Snort in term of evasion techniques as well as show its performance and level of detection. The experiments conducted, have shown that Snort if properly configured, is able to detect the majority of the evasion techniques. The pre-processors Frag3, Stream5 and HttpInspect play an essential role in the detection of the evasion techniques as well as different threats such as scans or exploits. The experiments have shown that if they were bad configured the level of detection can decrease and therefore miss threats.

Furthermore the experiments showed as well that the resources and the time required increase due to the number of options activated in the configuration of Snort. It is perfectly normal since the number of checking and computation increase as well. However the resources required stay reasonable and the level of detection increases. Therefore according to the resources available the user can obtain good performance with Snort.

However in the experiments conducted, the traffic was not overloaded and it explains why the resources required stay reasonable. In a busy network the results could change and it could be interesting to investigate that question but it should not change a lot. Furthermore it could be interesting as well to compare the results with other IDS for instance with commercial IDS.

The results are strongly linked to the network used in this paper. Therefore one configuration can be optimal for a network and inappropriate for another. The trade-off between performance and detection is sensible to the network and the technologies used. It means that simply copy the security measures for a network would not be efficient for a another network even in the same company using different networks. Each network should be considered with the same importance and level of protection and examine with strong attention.

To conclude the results identified in this paper show the importance of an IDS and its danger if not configured properly as well as the variety of threats nowadays. These threats are in a constant evolution and therefore the security controls must evolve as well in order no to get left behind.

5 References

Innella, P., McMillan, O., Tetrad Digital Integrity, LLC, (2001), '*An Introduction to IDS*', Available at: http://www.symantec.com/connect/articles/introduction-ids (accessed on 22/05/2012)

Micorsoft, (2011), '*What is a botnet*', Available at: http://www.microsoft.com/security/resources/botnet-whatis.aspx (accessed on 22/06/2012)

Oh, J., Park, S., Jeon, Y., (2007), '*Detection of DDoS and IDS Evasion Attacks in a High-Speed Networks Environment*', Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.9533&rep=rep1&type=pdf (22/05/2012)

Ptacek, T., Newsham, T., (1998), '*Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*', Available at: http://insecure.org/stf/secnet_ids/secnet_ids.html (accessed on 22/05/2102)

Roesch, M., Green, C., Sourcefire Inc, (2011), '*SNORT Users Manual 2.9.2*', http://www.snort.org/assets/166/snort_manual.pdf (accessed on 22/05/2012)

Securixlive, (2012), '*A unified output system for Snort*', http://www.securixlive.com/barnyard2/download.php (accessed on 30/08/2012)

Stonesoft, (2011), '*Anti-evasion*', Available at: http://www.antievasion.com/principles (accessed on 22/06/2012)

Timm, K., (2006), '*IDS Evasion Techniques and Tactics*', Available at: http://www.symantec.com/connect/articles/ids-evasion-techniques-and-tactics (accessed on 22/06/2012)