Evading Intrusion Detection Systems

I.AlRobia and M.Papadaki

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

Abstract

Snort is a well-known open source Intrusion Detection System that can be used as a second line of defense in a network to detect any incoming attacks from any source (such as Nikto) and alert the network administrator about this attack. This research will test Snort's durability against Nikto's evasion attacks.

Keywords

Evading Intrusion Detection Systems (IDS), Evading Snort, Nikto anti-IDS Evasion Techniques.

1 Introduction

Computer Systems and Networks suffer from complex security threats that arise and grow rapidly along with new computer and network technologies. IT mangers and network administrators are trying to find solutions for new security threats that might arise in the future by any necessary means, such as deploying firewalls, antivirus programs, or any other defending devices. Kerry Cox and Christopher Greg states that in the old days a firewall was most of what an administrator needed to protect a network from attack. (Cox and Greg, 2004) However, it is not enough to focus our trust into firewalls and updated antivirus programs. A second line of defense must be implemented in order to ensure the 'Optimum' level of security. According to James Anderson, Intrusion Detection Systems (IDS) can be that second line of defense. (Anderson, 1980) Where firewalls act like locked doors and windows leading to your computer, intrusion detection systems act more like a burglar alarms to your computer. (Wang, 2003)It will alert you about different intrusions and attacks that have a probability of affecting your system. Additionally, the network administrator must know how to evade an IDS in order to defend it.

The role of an intrusion detection system is to identify and sometimes isolate intrusions against computer systems (Ptacek and Newsham, 1998). It is used as a second line of defence along with the firewall and any other component that can be used to secure the computer system and detect suspicious activities. It can provide detection and notifications for new attacks that have not been discovered by any other security component. Moreover, intrusion detection systems can provide forensic information that might allow administrators to discover the origins of an attack and capture those attackers (Ptacek and Newsham, 1998). However, it is a challenging task to detect these attacks, this is because that attackers tries to develop

different evasion techniques in order to bypass the intrusion detection system. Therefore, an administrator has to update his security systems regularly and to be prepared for any suspicious events.

IDS have been one of the key countermeasures against network compromise however this is only if the IDS have been well-configured, they have to know how to select and configure intrusion detection systems for their specific computer system and network environments, and most importantly they have to know how to deal with the intrusion detection system's output and integrate it with the rest of the organization's infrastructure (Bace and Mell, 2001).

In the early generations, it is possible to say that early intrusion detection systems could be just like a tool that have been used to specify extended regular or hexadecimal expressions to match against data payloads of packets called 'ngrep' (Niphadkar, 2008). In other words, detection was heavily relying on the detection of character at the packet payload rather than using more sophisticated detection methods.

The purpose of this project is to investigate how resilient modern Intrusion Detection Systems are to traditional IDS evasion techniques. Apart from detection capability, another issue will be examined which is the performance cost of anti-evasion techniques.

This research will answer the following questions:

- Whether Snort has the capability to detect Nikto evasion techniques with default configurations?
- How well will Snort detect the incoming attacks in when the processing power is being shared by other applications?
- Will Nikto anti-IDS be able to evade Snort by using a Single technique? What about multiple techniques when combined together?
- If the detection was successful, what are the preprocessors used by Snort to detect Nikto evasion techniques?

2 Aims and Objectives of the research

The aims and objectives of this research are to:

- Demonstrate awareness of intrusion detection technologies as well as IDS evasion techniques.
- Design and implement experiments that will investigate the evasion resilience of Snort, an open source IDS tool, under different configurations
- Based on the results of the experiments, propose optimal configurations that help to tackle evasion tools.

3 Research Design

Since this research relays on several components and in order to localize any mistakes and avoid them, this research will divide the research design and setup into five parts and test them individually. These five parts are:

1. Setup and test VMware Workstation.
2. Setup Nikto anti-IDS scanner tool.
3. Setup and test Snort.
4. Test Snort in basic configuration with Nikto anti-IDS scanner tool.
5. Test Snort with configuring the preprocessors (such as frag3 preprocessor) with Nikto anti-IDS scanner tool.

After setting up and testing the mentioned parts the network topology should be like the following figure.

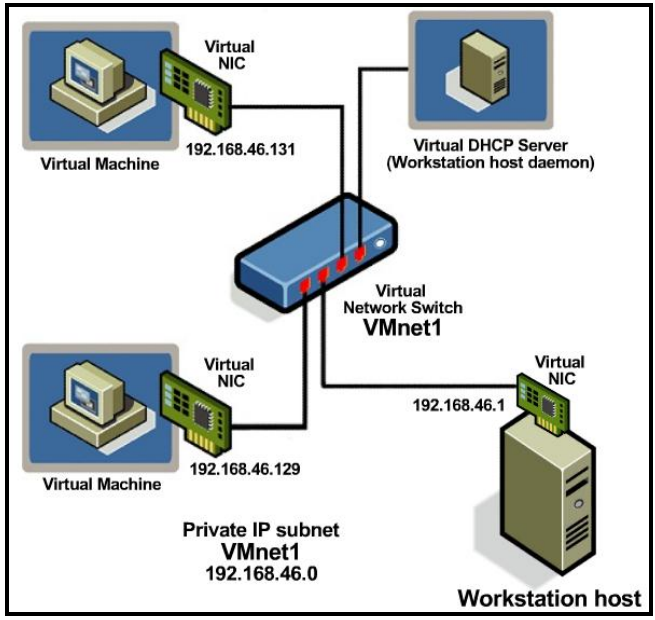

Figure 1: Network topology with packet Monitoring
Source: (Originally from) VMware eLearn course.

4 Results and Analysis

The version of Snort that has been used in this research did detect and analyzed all the attacks that have been sent to it and have an equal amount of alerts (104 alerts) as seen on Figure 2. Most of these detections were by the help of the preprocessors and Snort updated set of rules. When comparing this version of Snort with previous versions, we can find a lot of improvements in Snort detection ability and it processing power.

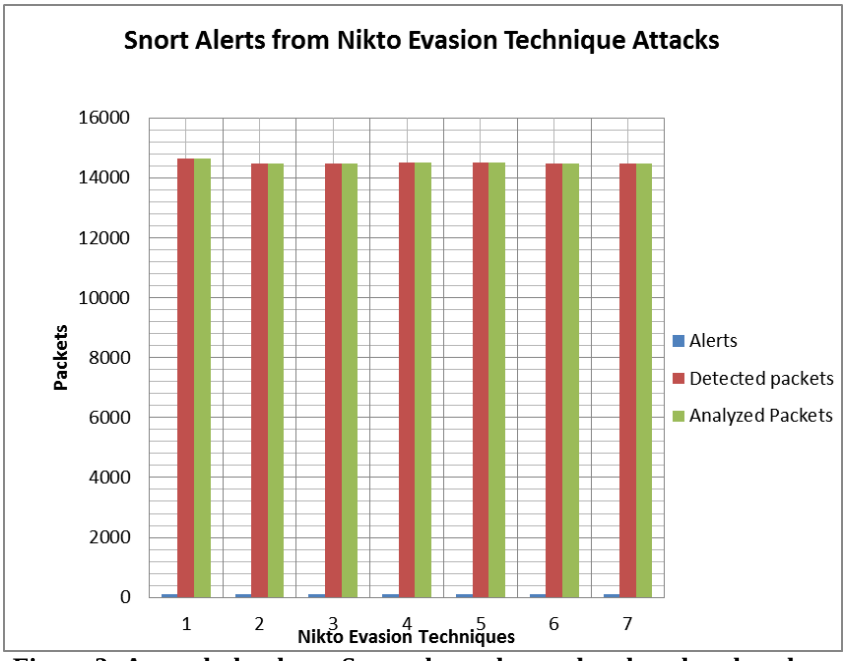

Figure 2: A graph the shows Snort alerts, detected and analyzed packets

This research will compare pervious test results that were conducted in previous researches such as Jarle Ytreberg (2007) research to show how much did Snort improve throughout the years. In Ytreberg's research, all evasion techniques have almost equal amounts of detections with the exception of evasion techniques number four and nine. This exception had occurred because of the method used when the attack had been conducted and transferred through the network. The methods that have been used for techniques number four and nine are *Prepend long random string* and *Session splicing* for evasion techniques four and nine respectively. Evasion technique four send packets that have unordinary long string in the GET request when sent to the virtual machine which will cause Snort to alert more than expected (Ytreberg, 2007). In the other hand, evasion technique nine (which is not supported by Nikto anymore) splits the attack to many small fragments to cause Snort IDS to spend a lot of processing power to reassemble the attack before processing it (Ytreberg, 2007). These evasion techniques did disturb Snort IDS in older versions by either making Snort generate a huge number of unnecessary alerts or by make it to consume a lot of time to reassemble it before processing it which might lead to dropping some legitimate packets. However, these problems did not cause any problem in the current version of Snort that has been used in this research and all evasion techniques did have equal numbers of alerts. The significant improvements in newer versions of Snort were caused by the help of Snort Preprocessors and Snort latest rule set.

When testing Snort in a busy environment, Snort did a great job detecting all evasion techniques generated by Nikto (see Figure 3). All packets that have been sent by Nikto have been analysed without dropping any incoming packets. This is because Snort did get a huge help from Barnyard since it makes Snort to run in full speed by

decoupling output overhead from Snort IDS and convert almost any spooled file by adding input and output plugins (sourceForge.net, 2010). Additionally, Snort preprocessors are enabled to help Snort by normalizing any incoming packets in order to make things easier for Snort. In comparison with Yterberg's results (2007) from Figure 3 and 4, we can see that Snort have improved in its detection capability and performance even if tested in a busy environment. Unlike Yterberg's results when testing Snort in a busy environment, current version of Snort did detect and analysed all incoming packets.

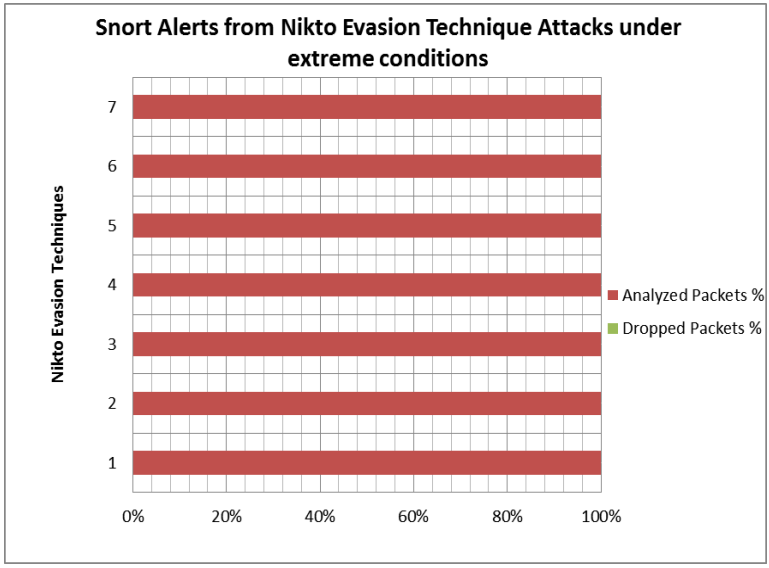

Figure 3: A graph that shows number of analyzed and dropped packets

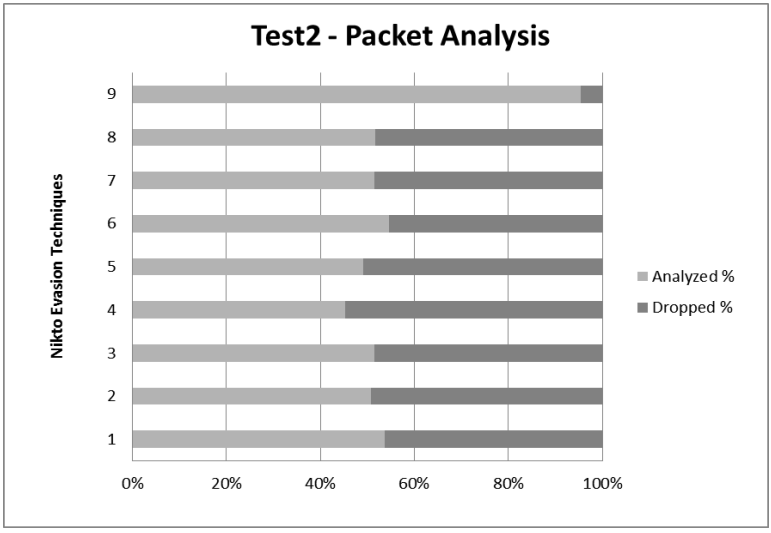

Figure 4: A graph that shows number of analyzed and dropped packets (Yterberg, 2007)

5 Conclusion

Network Intrusion Detection Systems (NIDS) can provide the level of security needed to be able to protect a company's or organization's assets. However, it will not be able to stop the attacks directed to the company's network. It will monitor all packets that are received from a designated network physical or virtual interface and will trigger an alert if it detects any packet that might harm the company's assets. Afterwards, it is up to the network administrator to decide whether to just ignore this packet or to figure out a way to prevent it from attacking the company's system.

Intrusion Detection systems (IDS) does have some flaws like any other device such as firewalls and antivirus programs. However, these programs can be very powerful if kept updated regularly. In addition, we can consider the defense in depth countermeasure according to Kerry Cox and Christopher Greg, which deploy multiple overlapping defense measures (such as firewalls, IDS, etc.) in order to get a well secured system. (Cox and Greg, 2004) Nevertheless, updating your systems and countermeasure devices does not give you the reason to rest; every system administrator should know the ins and outs of his system. He should read and try to hack his own system in order to defend himself from possible attacks since the best way to defend is by attacking. Kevin Timm published that BlackHat community develop several methods to evade IDS sensors while IDS vendors, IDS developers, and security researchers tries to develop counter act to bypass these attacks. (Timm, 2002)

Therefore, it is essential for a network administrator to create safe virtual network that is isolated from the physical network and start testing the latest rules that have been provided by Snort. This task is important to see if Snort capable of detecting latest attacking techniques or not and most importantly is to make these test in an isolated network that will not interfere with any external networks to avoid attacking other systems unintendedly. Therefore, it is advisable for network administrators to use one of VMware Workstations in order to create this environment. In addition, VMware Workstation gives the network administrator several options to when it comes to test Snort in different environments. It gives the administrator the option to create several virtual machines in order to mutate a real network. Moreover, the administrator can use the created virtual machine in order to attack Snort IDS with different tools in order to test Snort's durability. Additionally, VMware gives the network administrator the option to test Snort IDS in different environments by changing the specification of each virtual machine. This option provides a great opportunity for network administrators to test Snort IDS in extreme conditions.

From Nikto anti-IDS scanner tool, the network administrator can test Snort's detection capability by sending packets from the attacker virtual machine to Snort IDS with different evasion techniques. These evasion techniques can be sent individually or by combining several evasion techniques together. All of these tactics are being implemented to try to confuse or exhaust Snort IDS in order to let it ignore some of the received packet which will lead to a successful penetration.

6 References

Anderson, J. (1980) “Computer Threat Monitoring and Surveillance”, a Technical Report, Fort Washington, Pennsylvania.

Bace, R. and Mell, P (2001), “Intrusion Detection Systems”, NIST Special Publication on Intrusion Detection Systems, [online] Available at:http://www.bandwidthco.com/whitepapers/nist/NIST%20800-31%20Intrusion%20 Detections%20Systems.pdf

Cox, K. and Greg, C. (2004) “Managing Security with Snort and IDS tools”, O’Reilly Media ISBN: 0-596-00661-6, Pages 1-3.

Niphadkar, S. (2008). “Analysis of Packet Sniffers: TCPdump VS Ngrep VS Snoop”, Available at: http://mason.gmu.edu/~sniphadk/sniffer.pdf

Ptacek, T. and Newsham, T. (1998). “Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection ”, Secure Networks Inc., Available at: http://www.insecure.org/stf/secnet_ids/secnet_ids.pdf

Timm, K (2002) “IDS Evasion Techniques and Tactics”, [Online] Available at: http://www.securityfocus.com/infocus/1577

Wang, W. (2003) “Steal This Book 3: What they won’t tell you about the Internet?”, William Pollock Publications.

Ytreberg, J. (2007) “Network Intrusion Detection Systems Evasion Techniques: an Investigation using Snort”, Masters Dissertation, Plymouth University, UK.