Smurf Attack Detection using Snort IDS

Dr. T. Subbulakshmi, Shantanu Sakpal, Surendra Kumar Prajapat, Aditya Gaikwad and Kiruthika Devi BS

Abstract--- *The movement towards more secured computing system continues to rise as management becomes mindful of the numerous threats that exist to their organizations. Today Intrusion Detection Systems (IDS) have become a standard component of network security. Intrusion detection technology can help the system to deal with network attacks, extend the security management ability of the system manager and increase the integrality of information security foundation structure. So this paper helps in analyzing and evaluating the performance of SNORT IDS through alert Generated by intrusion detection system in high-speed networks.*

I. INTRODUCTION

THE term intrusion means an attempt of unauthorized usage of a computer system or computer resources, causing willful or incidental damage. Intrusion detection is the ability of detecting inapt, incorrect or anomalous activity. An IDS is a computer programs that tries to perform intrusion detection by comparing observable behavior against suspicious patterns, preferably in real time. Intrusion is primarily a network-based activity. With increasing global network connectivity, the topic of intrusion has gained prominence, spurring active research on efficient ideas.

1.1 Classification of NIDS:

IDS can be classified on the basis of a multitude of factors. With respect to the place where the intrusion detection system takes place we have two kinds of IDSs.

Network Based IDSs: IDS[10] which operate on network data flows are called network based IDS. Intrusion Detection Techniques Host Based IDSs and Network Based IDS may use any of the following methods for detecting the unauthorized intrusionEm ail-based IDSs: email-based IDS examines ongoing traffic, activity, transactions, or behavior for matches with known patterns of events specific to known attacks.

Anomaly-based IDSs: Anomaly-based IDS examines ongoing traffic, activity, transactions, or behavior for anomalies on networks or systems that may indicate attack.

II. MODES OF SNORT

Snort[8] is a single-threaded application, which can be configured to operate in four modes.

2.1 *Packet Sniffer Mode* : Packet Sniffer mode simply reads the packets off of the network and displays them in a continuous stream on the console (screen).

2.2 *Packet logger Mode*: Packet Logger mode logs the packets to disk. To record the packets to the disk, specify a logging directory and Snort [8] will automatically know to go into packet logger mode. A directory named log in the current directory would be created. When Snort runs in this mode, it collects every packet it sees and places it in a directory hierarchy based upon the IP address of one of the hosts in the datagram.

2.3 *Detection Mode* : Network Intrusion Detection System (NIDS) mode allows Snort to analyze network traffic for matches against a user-defined rule set and performs several actions based upon what it sees.

2.4 *Prevention Mode*/ Inline Mode It prevents the network threats. Snort Inline obtains packets from IP tables instead of libpcap and then uses new rule types to help IP tables pass or drop packets based on Snort rules. There are three rule types you can use when running Snort with Snort Inline:

1) Drop - The drop rule type will tell IP tables to drop the packet and log it via usual Snort means.
2) reject - The reject rule type will tell IP tables to drop the packet, log it via usual Snort means, and send a TCP reset if the protocol is TCP or an ICMP port unreachable if the protocol is UDP.
3) Sdrop - The sdrop rule type will tell IP tables to drop the packet. Nothing is logged.

Dr. T. Subbulakshmi , School of Computing Science and Engineering, VIT University, Chennai-600127, Tamil Nadu.
Email: research.subbulakshmi@gmail.com
Shantanu Sakpal (II MCA) , School of Computing Science and Engineering, VIT University, Chennai-600127, Tamil Nadu.
E-mail: sakpal.shantanudilip2014@vit.ac.in
Surendra Kumar Prajapat (II MCA), School of Computing Science and Engineering, VIT University, Chennai-600127, Tamil Nadu.
E-mail: surendra.kprajapat2014@vit.ac.in
, Aditya Gaikwad (IIMCA), School of Computing Science and Engineering, VIT University, Chennai-600127, Tamil Nadu.
E-mail: aditya.gaikwad2014@vit.ac.in
Kiruthika Devi BS (PhD Research Scholar), School of Computing Science and Engineering, VIT University, Chennai-600127, Tamil Nadu.
E-mail: kiruthikadevi.bs2015@vit.ac.in

ISBN 978-93-85477-73-7

III. PROPOSED WORK:

Snort is a NIDS[8], implements real time scanning of attack detection and port scanning detecting. The System-Architecture as shown in the Figure1 consists of the following modules.

- Packet capture: Used to capture network traffic using libpcap library[9]. This module will capture packets to analyze raw traffic for detecting malicious behavior like sniffing, finger-printing, and many other uses. Libpcap Library is the library which we are going to use to grab packets right as they come off the network. Packet Decoder: This module will ensure to form the data structures of the packets captured and identify the network protocol. This module will extract the information of the origin and the destination of that particular packet so that is will help the system in identifying attacker.

- Pre-processor: Pre-processor are plugins developed generally in C and process the packets provided by the decoder and ensembles the packets received. This preprocessor are configured in snort.conf file configuration.

- Detection engine: Analyze the packets based in our rules configured in the rule-set.

- Rule Set: It is a list of parameters and specific set of rule which will guide the snort tool to differentiate between a normal user IP address and an attacker IP address.

- Alert Generation: The IP addresses which are identified as an attacker will be listed and an alert will be generated stating the list of attackers and intruders.

- Alert Database: The details of the attackers and intruders are stored in the database for future reference. This database can help in making new rule-sets also.

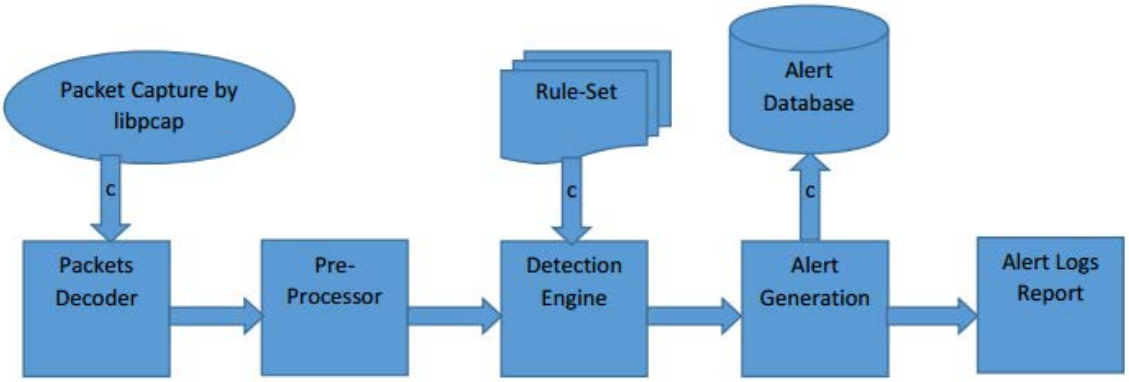

Figure 1. Proposed System Architecture

IV. SMURF ATTACK

The Smurf Attack is a distributed denial-of-service attack in which large numbers of Internet Control Message Protocol (ICMP) packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP Broadcast address[2]. Most devices on a network will, by default, respond to this by sending a reply to the source IP address. If the number of machines on the network that receive and respond to these packets is very large, the victim's computer will be flooded with traffic. This can slow down the victim's computer to the point where it becomes impossible to work on.

A ping flood is a simple denial-of-service attack where the attacker overwhelms the victim with ICMP Echo Request (ping) packets. This is most effective by using the flood option of ping which sends ICMP packets as fast as possible without waiting for replies. Most implementations of ping require the user to be privileged in order to specify the flood option. It is most successful if the attacker has more bandwidth than the victim[5]. The attacker hopes that the victim will respond with ICMP Echo Reply packets, thus consuming both outgoing bandwidth as well as incoming bandwidth. If the target system is slow enough, it is possible to consume enough of its CPU cycles for a user to notice a significant slowdown.

The following steps lead to a smurf attack as shown in Figure 2.

1. Huge numbers of ICMP requests are sent the victim's IP address.
2. The source destination IP address is spoofed.
3. The hosts on the victim's network respond to the ICMP requests.
4. This creates a significant amount of traffic on the victim’s network, resulting in consumption of bandwidth and ultimately causing the victim’s server to crash.

ISBN 978-93-85477-73-7

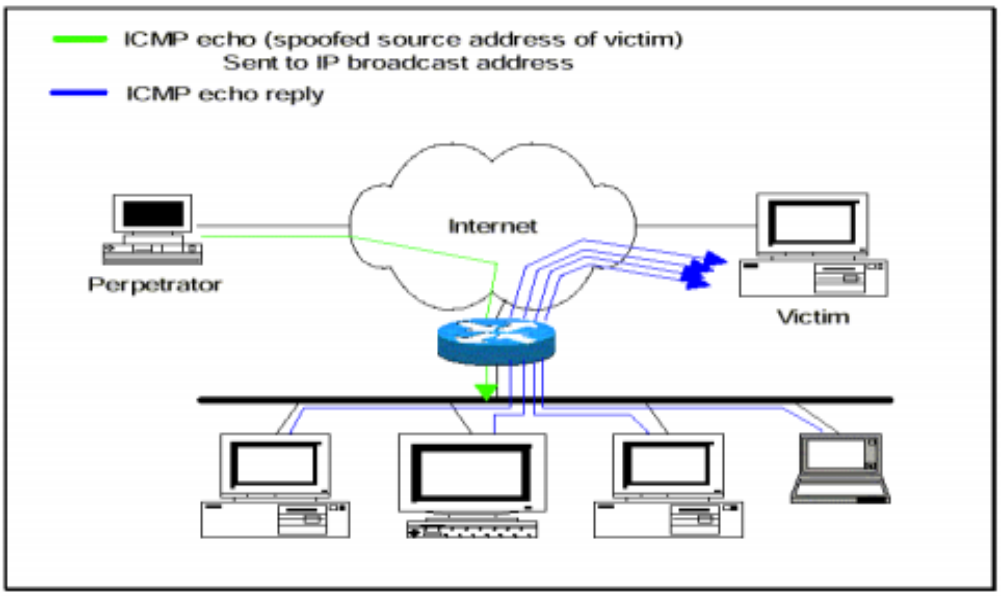

Figure 2. Smurf Attack

REFERENCES

[1] CERT Coordination Center, Denial of Service attacks.
[2] Computer Security Institute and Federal Bureau of Investigation, CSI/FBI Computer crime and security survey 2001, CSI, March 2001.
[3] D. Moore, G. Voelker, S. Savage, Inferring Internet Denial of Service activity, in: Proceedings of the USENIX Security Symposium, Washington, DC, USA, 2001, pp. 9–22.
[4] L.D. Stein, J.N. Stewart, The World Wide WebSecurity FAQ, version 3.1.2, February 4, 2002.
[5] D. Karig, R. Lee, Remote Denial of Service Attacks and countermeasures, Department of Electrical Engineering, Princeton University, Technical Report CE-L2001-002, October 2001.
[6] CIAC, Information Bulletin, I-020: Cisco 7xx password buffer overflow.
[7] Kenney, Malachi, Ping of Death, January 1997, Available from. 662 C. Douligeris, A. Mitrokotsa / Computer Networks 44 (2004) 643–666
[8] SNORT IDS, Available from: www.snort.org
[9] D. Davidowicz, Domain Name System (DNS) Security, 1999.
[10] CERT Coordination Center, Trends in Denial of Service attack technology, October 2001.

ISBN 978-93-85477-73-7