4 Test Design

This chapter begins with an overview of the testing. It then proceeds to elaborate on the attacks included in the test data. Following this, a breakdown of each test data set, in terms of the attacks that it contains, is described. Finally, the chapter discusses the four steps of the testing process.

4.1 Overview of Tests

In order to achieve the objectives, a test environment was set up, testing conducted on the test machine and their results gathered and evaluated. Four sets of test data were run through four configurations of Snort. In total, there were 16 test runs; an example of a test run is described in section 5.4.

Figure 4.1 illustrates the overview of the testing:

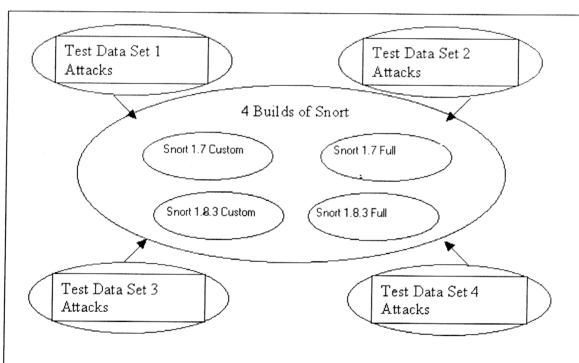


Figure 4.1: Test Overview

4.2 Test Data

The test data sets were obtained from http://www.ll.mit.edu/IST/ideval. These off-line data sets are publicly available to provide researchers with extensive examples of attacks and background traffic. This data consists of four weeks of traffic captured using tcpdump, a well-known open-source packetsniffer, which can be replayed in a network environment. The test data was downloaded and unpacked using gunzip into the root directory in the test machine.

The sequence attacks injected into test data set 1, 2, 3 and 4 are listed in appendices A, B, C and D, respectively. The attacks came from 5 different categories, they were:

- **Denial of Service (DoS)**: Attacks of this type are designed to disrupt a host or network service. As a result, legitimate user access or requests are denied.
- **User to Root (U2R)**: This category consists of attacks where a local user on a machine is able to obtain privileges normally reserved for the Unix super user.
- **Remote to Local (R2L)**: In these attacks, an attacker, who does not have an account on a victim machine, gains local access to the machine, exfiltrates files from the machine, or modifies data in transit to the machine.
- **Probe**: These attacks automatically scan a network of computers or a DNS server to find valid IP addresses, active ports, host operating system types, and known vulnerabilities.
- **Data**: The goal of a data attack is to exfiltrate special files which the security policy specifies should remain on the victim hosts.

4.2.1 Attack Descriptions for the Test Data Sets

A) Denial of Service Attacks

Table 4.1 describes the Denial of Service attacks in the test data sets.

Table 4.1: Description of the Denial of Service Attacks

Apache2	The Apache2 is an attack against an apache web server where a client sends a request with many http headers. If the server receives many of these requests it will slow down, and may eventually crash.
arppoison	In this attack the goal is to trick hosts on the same Ethernet into "learning" the wrong "Mac" address for known IP addresses. The attacker must have access to the Local Area Network.
back	In this Denial of Service attack against the Apache web server, an attacker submits requests with URL's containing many front slashes. As the server tries to process these requests it will slow down and becomes unable to process other requests.
crashiis	Crashiis is a Denial of Service attack against the NT IIS web server. The attacker sends a malformed GET request via telnet to port 80 on the NT victim. The command "GET ../.." crashes the web server and sometimes crashes the ftp and gopher daemons as well, because they are part of IIS.
dosnuke	Dosnuke is a Denial of Service attack that sends Out Of Band data (MSG_OOB) to port 139 (NetBIOS), crashing the NT victim (blue screens the machine).
land	The Land attack is a Denial of Service attack that is effective against some older TCP/IP implementations. The Land attack occurs when an attacker sends a spoofed SYN packet in which the source address is the same as the destination address.
mailbomb	A Mailbomb is an attack in which the attacker sends many messages to a server, overflowing that server's mail queue and possible causing system failure.

SYN Flood (Neptune)	A SYN Flood is a Denial of Service attack to which every TCP/IP implementation is vulnerable (to some degree). Each half-open TCP connection made to a machine causes the 'tcpd' server to add a record to the data structure that stores information describing all pending connections. This data structure is of finite size, and it can be made to overflow by intentionally creating too many partially-open connections.
Ping of Death	The Ping of Death is a Denial of Service attack that affects many older operating systems. Although the adverse effects of a Ping of Death could not be duplicated on any victim systems, some systems will react in an unpredictable fashion when receiving oversized IP packets. Possible reactions include crashing, freezing, and rebooting.
Processtable	The Processtable attack can be waged against numerous network services on a variety of different Unix systems. The attack is launched against network services which fork() or otherwise allocate a new process for each incoming TCP/IP connection.
Selfping	The Selfping attack is a Denial of Service attack in which a normal user can remotely reboot a machine with a single ping command. This attack can be performed on Solaris 2.5 and 2.5.1.
Smurf	In the Smurf attack, attackers use ICMP echo request packets directed to IP broadcast addresses from remote locations to create a Denial of Service attack address of many subnets, resulting in a large, continuous stream of 'ECHO' replies that flood the victim.
Sshprocesstable	Sshprocesstable is similar to the Processtable attack in that the goal of the attacker is to cause sshd daemon on the victim to fork so many children that the victim can spawn no more processes. This is due to a kernel limit on the number of processes that the OS will allow.
Syslogd	The Syslogd exploit is a Denial of Service attack that allows an attacker to remotely kill the syslogd service on a Solaris server. When Solaris syslogd receives an external message it attempts to do a DNS lookup on the source IP address. If this IP address doesn't match a valid DNS record, then syslogd will crash with a Segmentation Fault.
Tcpreset	Tcpreset is a Denial of Service attack that disrupts TCP connections made to the victim machine. That is, the attacker listens (on a local or wide-area network) for tcp connections to the victim, and sends a spoofed tcp RESET packet to the victim, thus causing the victim to inadvertently terminate the TCP connection.

Teardrop	The Teardrop exploit is a Denial of Service attack that exploits a flaw in the implementation of older TCP/IP stacks. Some implementations of the IP fragmentation re-assembly code on these platforms do not properly handle overlapping IP fragments.
Udpstorm	A Udpstorm attack is a Denial of Service attack that causes network congestion and slowdown. When a connection is established between two UDP services, each of which produces output, these two services can produce a very high number of packets that can lead to a Denial of Service on the machine(s) where the services are offered. Anyone with network connectivity can launch an attack; no account access is needed.

(B) User to Root Attacks

Table 4.2 describes the User to Root attacks in the test data sets.

Table 4.2: Description of the User to Root Attacks

Anypw	Anypw is a console User to Root attack that allows the attacker to logon to the system without a password. A boot disk is used to modify the NT authentication package so that a valid username can login with any password string. Logins via telnet also work with any password.
Casesen	Casesen is a User to Root attack that exploits the case sensitivity of the NT object directory. The attacker ftps three attack files to the victim: soundedt.exe, editwavs.exe, psxss.exe (the names of the files were chosen to make the attack more stealthy). The attacker then telnets to the victim and runs soundedt.exe. A new object is created in the NT object directory called \??\c: which links to the directory containing the attack files. A posix application is started thus activating the trojan attack file, psxss.exe, which results in the logged in user being added to the Administrator§ user group.
Eject	The Eject attack exploits a buffer overflow is the 'eject' binary distributed with Solaris 2.5.
ffbconfig	The Ffbconfig attack exploits a buffer overflow is the 'ffbconfig' program distributed with Solaris 2.5.

Fdformat	The Fdformat attack exploits a buffer overflow is the 'fdformat' program distributed with Solaris 2.5. The fdformat program formats diskettes and PCMCIA memory cards. The program also uses the same volume management library, libvolmgt.so.1, and is exposed to the same vulnerability as the eject program.
Loadmodule	The Loadmodule attack is a User to Root attack against SunOS 4.1 systems that use the xnews window system. The loadmodule program within SunOS 4.1.x is used by the xnews window system server to load two dynamically loadable kernel drivers into the currently running system and to create special devices in the /dev directory to use those modules. Because of a bug in the way the loadmodule program sanitizes its environment, unauthorized users can gain root access on the local machine.
Ntfsdos	This console-based attack reboots the system from a floppy disk containing NTFSDOS.EXE. This executable is used to mount the hard drives, giving the attacker the ability to read and copy files that would otherwise be protected by Windows NTFS security. The attack may be considered a User to Root attack because the attacker can access files that only the Administrator has permission to use.
Perl	The Perl attack is a User to Root attack that exploits a bug in some Perl implementations.
Ps	The Ps attack takes advantage of a race condition in the version of 'ps' distributed with Solaris 2.5 and allows an attacker to execute arbitrary code with root privilege.
Sechole	The attacker (a regular user) ftps to the victim and uploads test.exe and testfile.dll (filenames were chosen to be stealthy). The attacker then telnets to the victim and runs test.exe. The result is the attacker is added to the Administrators group.
Xterm	The Xterm attack exploits a buffer overflow in the Xaw library distributed with Redhat Linux 5.0 (as well as other operating systems not used in the simulation) and allows an attacker to execute arbitrary instructions with root privilege.
Yaga	Yaga is a User to Root attack. It adds the attacker to the Domain Admins group by hacking the registry. The attacker edits the victim's registry so that the next time a system service crashes on the victim, the attacker is added to the Domain Admins group.

(C) Remote to Local Attacks

Table 4.3 describes the Remote to Local attacks in the test data sets.

Table 4.3: Description of the Remote to Local Attacks

Dictionary	The Dictionary attack is a Remote to Local attack in which an attacker tries to gain access to some machine by making repeated guesses at possible usernames and passwords. Users typically do not choose good passwords, so an attacker who knows the username of a particular user (or the names of all users) will attempt to gain access to this user's account by making guesses at possible passwords.
FrameSpoofer	This attack tricks the victim into believing he is viewing a trusted web site, but in actuality the page's main body is spoofed with a frame created by the attacker.
Ftp-write	The Ftp-write attack is a Remote to Local attack that takes advantage of a common anonymous ftp misconfiguration. The anonymous ftp root directory and its subdirectories should not be owned by the ftp account or be in the same group as the ftp account. If any of these directories are owned by ftp or are in the same group as the ftp account and are not write protected, an intruder will be able to add files (such as an rhosts file) and eventually gain local access to the system.
Guest	The Guest attack is a variant of the Dictionary attack described in Section 8.1. On badly configured systems, guest accounts are often left with no password or with an easy to guess password. Because most operating systems ship with the guest account activated by default, this is one of the first and simplest vulnerabilities an attacker will attempt to exploit.
Httptunnel	In an Httptunnel attack, the attacker gains local access to the machine to be attacked and then sets up and configures an http client to periodically query a web server that the attacker has setup at some remote host. When the client connects, the server is able to send cookies that could request information be sent by the client, such as the password file on the victim machine. In effect, the attacker is able to "tunnel" requests for information through the http protocol.
Imap	The Imap attack exploits a buffer overflow in the Imap server of Redhat Linux 4.2 that allows remote attackers to execute arbitrary instructions with root privileges. The Imap server must be run with root privileges so it can access mail folders and undertake some file manipulation on behalf of the user logging in.

Named	The Named attack exploits a buffer overflow in BIND version 4.9 releases prior to BIND 4.9.7 and BIND 8 releases prior to 8.1.2. An improperly or maliciously formatted inverse query on a TCP stream destined for the named service can crash the named server or allow an attacker to gain root privileges.
Ncftp	Ncftp is an ascii ftp program for Linux. This attack exploits one of the popular features of the program: the ability to get subdirectories recursively. New (sub) directories are created on the local machine using the system() command (e.g. if any directories on the remote host contain an expression in backticks, that expression will be evaluated on the local machine when the directory is created.
Netbus	In this attack the attacker uses a Trojan program to install and run the Netbus server on the victim machine. Once Netbus is running, it acts as a backdoor. The attacker can then remotely access the machine using the Netbus client.
Netcat	In this attack the attacker uses a trojan to install and run the netcat program on the victim machine on a specific port. Once netcat is running, it acts as a backdoor. The attacker can remotely access the machine through the netcat port without a username or password.
Phf	The Phf attack abuses a badly written CGI script to execute commands with the privilege level of the http server. Any CGI program which relies on the CGI function escape_shell_cmd() to prevent exploitation of shell-based library calls may be vulnerable to attack. In particular, this vulnerability is manifested by the "phf" program that is distributed with the example code for the Apache web server.
Ppmacro	This Remote to Local attack uses a trojan PowerPoint macro to read secret files. This attack is based on a particular scenario. The victim user usually receives PowerPoint templates from an outside source via email attachment. He runs a built-in macro which inserts a graph displaying web statistics, saves the presentation as a ppt file, and posts it on the web.
Sendmail	The Sendmail attack exploits a buffer overflow in version 8.8.3 of sendmail and allows a remote attacker to execute commands with superuser privileges. By sending a carefully crafted email message to a system running a vulnerable version of sendmail, intruders can force sendmail to execute arbitrary commands with root privilege.

Sshtrojan	In a Sshtrojan attack, the attacker tricks the system administrator into installing (as a "Y2K Upgrade") a trojan version of the SSH program. This trojan version allows the attacker (or anyone) to login to the victim, via ssh, with the login "monkey" and no password. Upon login, a root privilege shell is spawned for the attacker.
Xlock	In the Xlock attack, a remote attacker gains local access by fooling a legitimate user who has left their X console unprotected, into revealing their password. An attacker can display a modified version of the xlock program on the display of a user who has left their X display open (as would happen after typing 'xhost +'), hoping to convince the user sitting at that console to type in their password.
Xsnoop	In the Xsnoop attack, an attacker watches the keystrokes processed by an unprotected X server to try to gain information that can be used gain local access the victim system. An attacker can monitor keystrokes on the X server of a user who has left their X display open. A log of keystrokes is useful to an attacker because it might contain confidential information, or information that can be used to gain access to the system such as the username and password of the user being monitored.

(D) Probe Attacks

Table 4.4 describes the Probe attacks in the test data sets.

Table 4.4: Description of the Probe Attacks

Insidesniffer	Here the attacker merely attaches a new machine to an inside Ethernet hub, configured with an IP, and begins sniffing traffic.
Ipsweep	An Ipsweep attack is a surveillance sweep to determine which hosts are listening on a network. This information is useful to an attacker in staging attacks and searching for vulnerable machines.
Ls_domain	Here the attacker uses the "nslookup" command in interactive mode to "list" all machines in a given DNS domain from a mis-configured primary or secondary DNS server. Thus the attacker can learn what machines (IP addresses) belong to (and perhaps exist in) the domain.
Mscan	Mscan is a probing tool that uses both DNS zone transfers and/or brute force scanning of IP addresses to locate machines, and test them for vulnerabilities.

Nmap	Nmap is a general-purpose tool for performing network scans. Nmap supports many different types of portscan options inclusive of SYN, FIN and ACK scanning with both TCP and UDP, as well as ICMP (Ping) scanning. The Nmap program also allows a user to specify which ports to scan, how much time to wait between each port, and whether the ports should be scanned sequentially or in a random order.
NTInfoScan	NTInfoScan is a NetBIOS based security scanner. It scans the NT victim to obtain share information, the names of all the users, services running, and other information. The results are saved in an html file named .html where victim is the victim's hostname.
QueSO	QueSO is a utility used to determine what type of machine/operating system exists at a certain IP address. QueSO sends a series of 7 tcp packets to any one port of a machine and uses the return packets it receives to lookup the machine in a database of responses.
ResetScan	ResetScan sends reset packets to a list of IP addresses in a subnet to determine which machines are active. If there is no response to the reset packet, the machine is alive. If a router or gateway responds with "host unreachable," the machine does not exist.
Saint	Saint is the Security Administrator's Integrated Network Tool. In its simplest mode, it gathers as much information about remote hosts and networks as possible by examining such network services as finger, NFS, NIS, ftp and tftp, rexd, statd, and other services.
Satan	Satan is an early predecessor of the Saint scanning program described above. While Saint and Satan are quite similar in purpose and design, the particular vulnerabilities that each tools checks for are slightly different.

(E) Data Attacks

Table 4.5 describes the Data attack in the test data sets.

Table 4.5: Description of the Data Attack(s)

Secret	A Secret attack is an attack where the attacker maliciously or mistakenly transfers data which they have access to a place where it doesn't belong. For example, transferring data from a classified computer/network to a non-classified computer/network would constitute a Secret attack.

4.2.2 The Test Data Sets

There were four attack-filled test data sets used in the testing of Snort. The particular attacks in each test data set are listed in the following subsections.

(A) Test Data Set 1

Figure 4.2 lists the attacks in test data set 1 by category.

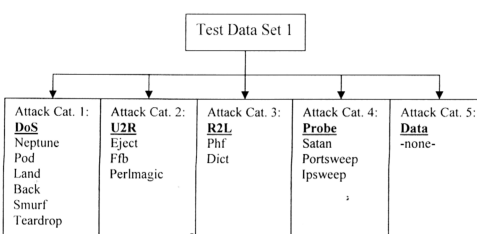


Figure 4.2: Attacks in Test Data Set 1

Majority of the attacks in test data set 1 were from the Denial of Service category. There were some attacks from the User to Root, Remote to Local and Probe categories,

but there were no attacks from the Data category. The sequence of the attacks executed in this test data set can be seen in Appendix A.

(B) Test Data Set 2

Figure 4.3 lists the attacks in test data set 2 by category.

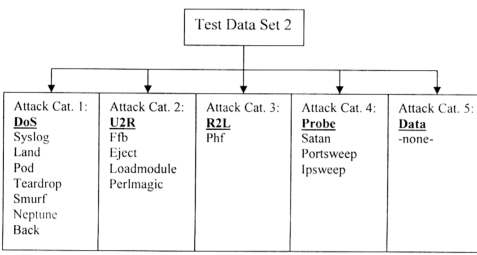


Figure 4.3: Attacks in Test Data Set 2

Majority of the attacks in test data set 2 were from the Denial of Service category. There were some attacks from the User to Root, Remote to Local and Probe categories, but there were no attacks from the Data category. The sequence of the attacks executed in this test data set can be seen in Appendix B.

(C) Test Data Set 3

Figure 4.4 lists the attacks in test data set 3 by category.

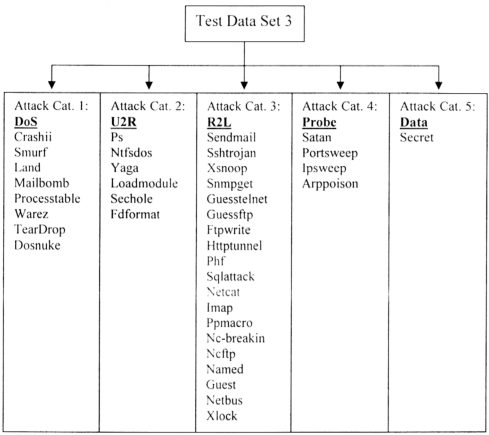


Figure 4.4: Attacks in Test Data Set 3

Test data set 3 has attacks from all categories. This test data set is bigger than test data sets 1 and 2. Most of the attacks came from the Remote to Local category. There was one type of attack in the Data category. In the Probe category, Arppoison was introduced. This attack was not included in the previous test data sets. The sequence of the attacks executed in this test data set can be seen in Appendix C.

(D) Test Data Set 4

Figure 4.5 lists the attacks in test data set 4 by category.

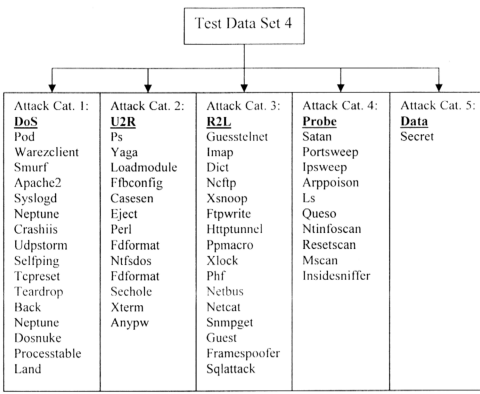
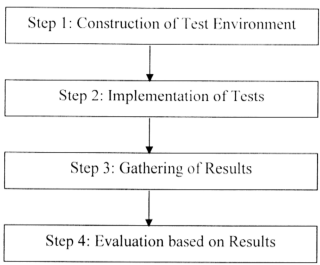


Figure 4.5: Attacks in Test Data Set 4

Test data set 4 has attacks from all five categories. This is biggest test data set. Most of the attacks came from the Remote to Local and the Denial of Service categories. In the Probe category, numerous attacks, such as Resetscan and Mscan, were introduced. These attacks were not included in any of the previous test data sets. This test data set had the most number of Probe attack types compared to the previous sets. The sequence of the attacks executed in this test data set can be seen in Appendix D.

4.3 Steps to Testing

The testing process can be divided into four distinct steps, they are:



4.3.1 Construction of Suitable Test Environment

The test environment consists of the test machine, test data, the operating system on the test machine, Snort (the IDS to be tested), SnortSnarf (the utility that parses Snort's alerts) and the test results forms. Firstly, the test machine specifications were decided on, as described in section 3.5.1, to suit the speed and processing space required to run the test data, Snort and SnortSnarf. Then installation of the operating system, test data, Snort and SnortSnarf were undertaken.

A) Test Machine

Only one test machine was needed. This sole computer had a reasonably fast Pentium Celeron processor (1.7 GHz) and 512 MB of RAM. A 40 GB hard disk was fitted on to the machine to hold the huge alert files generated by Snort and the data parsed by SnortSnarf.

(B) Test Data

Four test data sets with a multitude of attacks included in each set were required to do a thorough testing. The attacks injected into the test data sets are from the following categories of attacks: Denial of Service (DoS), Probe, User to Root (U2R), Remote to Local (R2L) and Data. The specific attacks in the test data were described previously in section 4.2.2. The order of the attacks in the test data sets are provide in Appendices A, B, C and D. These test data sets are publicly available from http://www.ll.mit.edu/IST/ideval/ and were downloaded into the /root directory.

(C) Operating System

The operating system installed on the test machine was Linux 8.0. Linux was selected as the operating system because it is the best variation of Unix that suits stand-alone desktop installation. The version, 8.0, was picked because it was the most up-to-date version of Linux at the time of this research. The following installation process took place:

1. Insert Redhat CD-Rom1 and boot
2. The installation screen appears
3. Select the keyboard configuration
4. Select the mouse configuration
5. The Welcome screen appears
6. Select installation options
7. Select Disk Partitioning Options
8. Partition the Hard drive
9. Configure LILO

10. Configure the users

11. Set the time zone

(D) Snort (The Selected IDS)

Four different configurations of Snort were used for testing. The different configurations were:

- Snort 1.7 Full
- Snort 1.7 Custom
- Snort 1.8.3 Full
- Snort 1.8.3 Custom

The different configurations were determined by the different ruleset used for the particular configuration. The four different ruleset configurations used are provided in Appendix E.

(E) SnortSnarf (IDS Alerts Reporter)

SnortSnarf is a Perl program that processes alerts files from Snort and produces HTML output intended for diagnostic inspection and tracking down problems. Snort can be executed on every alerts file produced by the test runs to generate a convenient HTML breakdown of all the alerts.

(F) Result Forms

Four result forms were constructed according to the attacks in the test data sets that are listed in appendices A, B, C and D. The result forms were constructed as follows:

- Result Form for test data set 1 (Appendix F) – List of Attacks from Appendix A
- Result Form for test data set 2 (Appendix G) – List of Attacks from Appendix B

- Result Form for test data set 3 (Appendix H) – List of Attacks from Appendix C
- Result Form for test data set 4 (Appendix I) – List of Attacks from Appendix D

4.3.2 Implementation of Tests

The implementation of the tests consisted of configuring Snort and running the four attack-filled test data sets through the four configurations of Snort. This is presented at length in chapter 5.

4.3.3 Gathering of Results

Once the test data was run and response ensued by the IDS, the results of attacks detected would be written into the Snort alerts file in /var/log/snort/alert. The resulting alerts file was huge thus requiring the use of SnortSnarf which read the alerts from the alerts file and generate alerts reports. During the test, false alarms (can hereafter be addressed as 'false positives') were also picked up by Snort and reported by SnortSnarf, so manual filtering of the SnortSnarf reports were done in order to transfer the relevant alerts to the results forms as described in section 3.4.

4.3.4 Evaluation of Results

An evaluation of intrusion detection systems is essential to understand their capabilities and limitations so that there is no relaxation of vigilance based on unproven assumptions concerning system performance. Here, the four configurations are evaluated to see how they fare against the attacks in the test data sets. At this point, they are conferred their respective performance ranking based on the test results. This is discussed at length in chapter 7.

4.4 Summary

This chapter provides an overview of the testing process. It describes the test data, the attacks in the test data and lists the attack types in each of the four test data sets by category. The attacks in the test data sets can be divided into five categories, they were: Denial of Service (DoS), Remote to Local (R2L), User to Remote (U2R), Probe and Data. Finally, the chapter addresses the four steps of the testing process that consisted of:

1. Construction of test environment.
2. Implementation of tests.
3. Gathering of results.
4. Evaluation based on test results.