

Testing Network Security Using OPNET

Agustin Zaballos, Guiomar Corral, Isard Serra, Jaume Abella
Enginyeria i Arquitectura La Salle, Universitat Ramon Llull, Spain
Paseo Bonanova, 8, 08022 Barcelona Tlf: +34 93 2902400, Fax: +34 93 2902416
E-mail: {guiomar, zaballos, isards, jaumea}@salleURL.edu

Abstract

Network security nowadays has become a major priority for both network design and implementation. Due to this fact, the interest in making network communications secure has increased at the same rate as the accessibility to Internet services.

Although security is a critical issue in e-business, it is often impossible to measure its effectiveness in real life because of the network administrators' fears or prejudice.

In order to find a solution to this particular issue, once more simulation opens the path to solving problems that are hard to fix in real life.

Introduction

As the development of network communications increases, the importance of making them secure has become a common priority. In order to improve and support network security development it is important to consider the help that OPNET can provide in testing security performance before its deployment. The main goal of our study is to give an overall view of all the devices and techniques available within the OPNET Modeler related to security.

To make the network security test more realistic, we have taken as a reference the proposal for secure network implementation called Cisco SAFE [1], which has been developed by Cisco Systems. This proposal offers a global security comprehension as well as an approach to real network implementation. Once we had the environment in which the corresponding test was to be done, we performed these tests for all the devices and techniques that are available with OPNET.

In order to have a reference in security testing, we referred the tests to the Open Source Security Testing Methodology Manual (OSSTMM) [6]. This manual outlines a global comprehension of the many security issues to be checked in order to perform a valid security test.

Finally, we took the previous studies that had been done in security testing using OPNET, and we made a new proposal for continuing the development of network security devices and techniques. Actually, our proposals are focused on building an Intrusion Detection System device in order to detect different traffic patterns identified as attacks. On the same subject, we have started a new project for developing an attack traffic database, capable of performing denial of service attacks in a simulated environment.

Network security environment

In order to focus our study in a real life scenario, we researched on a proposal that gave us a comprehensive knowledge about network devices placement and configuration. The proposal had to fulfill several requirements, which should lead to an overall security environment. The Cisco Systems SAFE proposal was well explained and took into account details such as device configuration, which led us to take it as the reference point [1,2].

The following network topology is an approach to the mentioned manufacturers' proposal. As its architecture and device performance are revealed, the content of the proposal will be briefly explained in this paper.

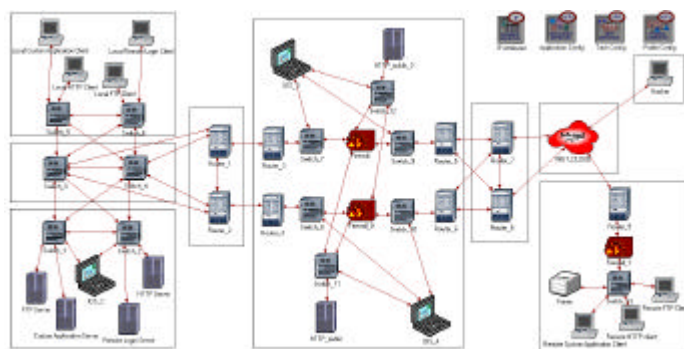


Figure 1: Network security testing scenario

As can be seen in Figure 1, the network has been divided into modules; each of them has a specific role and provides part of the global security implementation. From left to right we will describe each module and its network security contribution.

First of all, there is the Building module, which represents the internal part of the corporate business and holds the department computers, the server farm and the interconnection devices such as switches and routers [1]. This module deploys virtual local area networks for internal use. This VLAN technique is performed by switches and has the ability to separate broadcast domains, as well as dividing the network logically, which is useful to separate department traffic flows. This traffic division is considered part of the security implementation due to its internal separation of traffic, and the goal is the following: the fewer people have access to resources, the better it is. It is also important to keep in mind that internal security depends on physical access to resources and for this reason it is always good to have an internal security policy. [1]

The next part is called the Distribution module. It basically performs high-speed network interconnection. This module

separates the internal facilities from the public access services. The devices that can be found here are mainly switches, but it can also hold routers. Switches perform the high-speed interconnection and routers keep on the connections coming from the outside at the same time as they permit the outgoing connections. It is important to keep in mind that security in these modules is based on access control lists, because it is important to keep fast connections.

The next module is responsible for the on-line business and provides public access to corporate services. From the security point of view, this module is essential because it is a main target of hacker attacks. Denial of service attacks and database server intrusion are the main attractions for hackers. It is important that the network administrator performs a consistent configuration of all the devices involved in this module because a router misconfiguration could lead to damage in business and to a probably loss of client confidence. The devices that are part of this module are many and varied: routers, switches, firewalls and intrusion detection systems. Security in this module depends not only on access control lists, but also on proxy firewalls and intrusion detection systems logs and alerts. Due to the above-mentioned reasons, it is very important to protect on-line business resources in order to maintain client confidence and also save money.

The next module refers to the Internet service provider. Security issues cannot be managed from the corporate module, but it is important to keep in mind the policies that are applied to directly connected routers, in order to apply the appropriated traffic filtering. The contract with the Internet service provider should specify what policy is going to be applied and who is responsible in case an attack occurs. Once the contract is established, corporate policies should be made in correspondence to the responsibilities that have been assumed.

Finally, there are three more modules: the Internet, the Hacker and the remote branch office. The Internet module represented by a cloud includes four wide area network routers. The main goal is to simulate Internet behavior by controlling traffic flow through a routing protocol that handles each connection. The connections can be initiated either by the Hacker or the remote branch office. The Hacker node has been attached directly to the Internet cloud, but technically this node also could be connected through a dial-up technology. This node is capable of performing a remote login connection to the remote login server in the corporate server farm. In addition the remote office has been configured to have access to the internal corporate resources, which means that it has full access to all available services.

Security testing reference

The following section will brief the main issues exposed in the OSSTMM document. This manual explains how TCP, UDP, ICMP, IP, and various application level protocols like FTP, DNS, TFTP, HTTP, HTTPS, etc work and how to test them [4]. In our case of study, we only looked for network and transport protocols, but it is always important to keep in mind how these protocols are related to application level protocols.

Basically, the information used for our study is related to ports and services, protocols and packets. Ports are memory address spaces used to hold TCP and UDP services. Each service expects a request to follow a specific procedure of interaction called a protocol. In this procedure the requests are delivered to the service in the form of packets.

Finally the parts that we focused on were TCP, UDP and ICMP protocol, packet structure and connections as well as flags used in the connections, expected responses and tricks that can be used to by-pass normal performance. Although not all the mentioned tests could be performed, it is also important to be aware of what can be done to perform a security test.

Testing network security

The scenario described above was implemented using OPNET to perform our network security study. Devices used for the tests included routers, firewalls and switches. Routers for the implementation of access control lists and virtual private network support, firewalls for access control lists and proxy services, and finally switches to test virtual local area network performance. With these devices and techniques we were able to achieve a considerable degree of likeness to our proposal.

The first scenario was designed to test the packet filtering technique using access control lists (ACL). ACLs are rules defined at the routers configuration in order to discard non-authorized packets. These rules are widely applied in networking to manage traffic flow accurately. It is important to define a consistent and global network policy in order to use the access control lists that filter unauthorized traffic and permit authorized connections. In this scenario, the Hacker node attempts to initiate a remote login connection to the remote login server inside the corporate server farm. This connection is dropped by the upper branch firewall due to an ACL definition that denies all traffic coming from the Hacker IP address. In an additional scenario, we have configured the firewall node to refuse all network connection to remote login service by configuring an access control list that denies all packets with destination port number 23, which corresponds to the remote access.

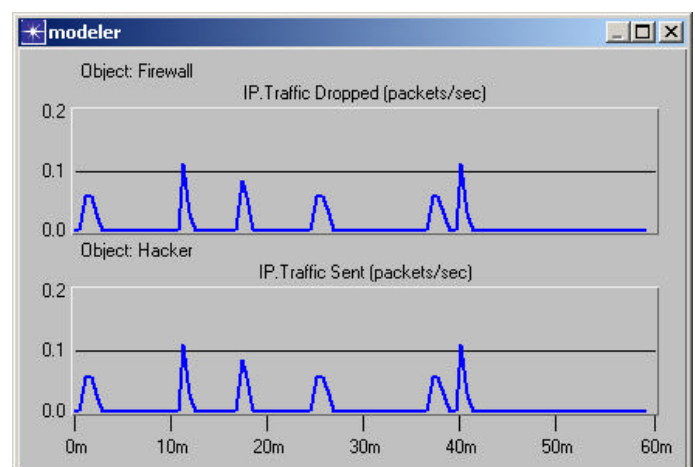


Figure 2: Packet filtering using ACLs

Figure 2 shows the packet filtering technique used at the upper branch firewall of the network topology, in order to filter the Hacker remote login connections. It can be seen that the firewall drops every single packet the Hacker send.

The next scenario was designed to test a second technique, which corresponds to proxy filtering. This technique can be used in firewalls, and it allows them to filter traffic flow, depending on its characteristics. For instance, FTP traffic can be filtered via proxy in a firewall causing all the connections to the FTP service to be refused. In our scenario, once again the firewall node was configured to hold this technique. The result of this test was that none of the connections that the remote site tried to initiate were successful.

The following scenario used the technique called virtual local area network (VLAN). This technique consists of configuring a switch to perform logical traffic division. In fact, the switch performs a port-based virtual network separation that permits corporate internal traffic to be divided logically. In our study we configured two VLAN. The first one was composed of the HTTP and FTP local clients and servers, and the second one was composed of the remote login and custom application client and server respectively. The main goal of this technique is to separate internal traffic to follow the rule mentioned before: the fewer the people that have access to a resource, the better it is.

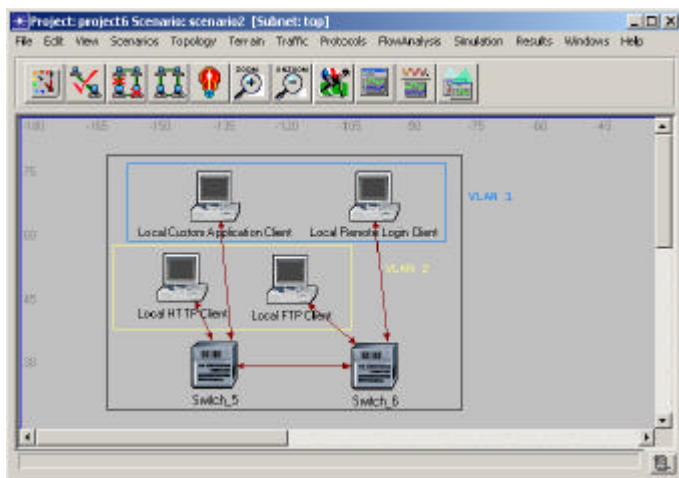


Figure 3: VLAN scenario configuration

Finally, the last technique was based on the performance of virtual private networks (VPN). VPN are configured at routers and permit safe communication through an unsafe environment. VPN creates what is called a “tunnel” (a logical group of routers that establish a safe path to destination). The “tunnel” is defined within the Internet and allows private traffic to be sent through the public path [3]. In our scenario, the tunnel was established between the corporate and the remote office that is, in fact, a common configuration.

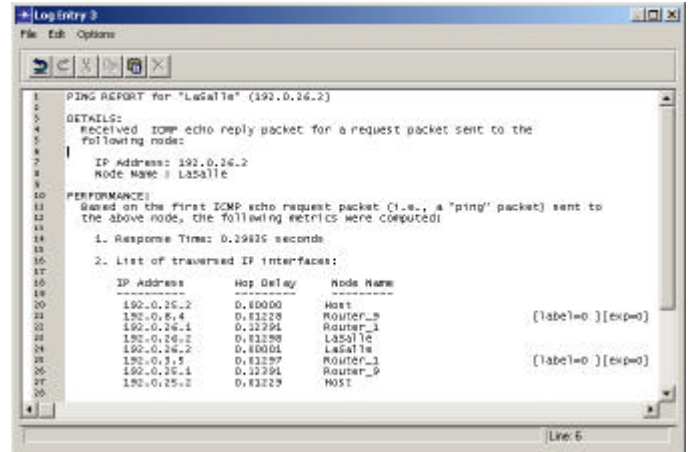


Figure 4: PING trace through a VPN tunnel

Traffic analysis

After testing all security devices and techniques provided by OPNET, we decided to go a little further. We studied another device that has been used in our scenario, the packet analyzer, which provides traffic sniffing properties. The traffic captured by this node can be viewed using the Application Characterization module that OPNET provides. The ACE module allows packet content to be examined carefully [5]. The following figure shows data packet information presented with ACE. In fact, the data shown in the picture corresponds to remote access traffic from the Hacker node to the corporate server.

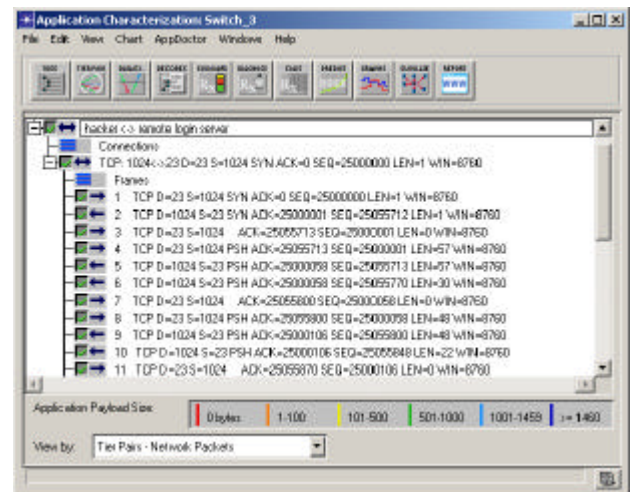


Figure 5: Remote access traffic presented with ACE

Packet analyzer is a powerful tool included in OPNET Modeler that enables the capture of data packets from traffic flow. This can be done by either capturing every single packet or filtering only the relevant traffic. Captured traffic can be exported to an external file, which can be viewed by other programs. The above-mentioned capabilities, gave us the idea of developing a viability study in order to build an Intrusion detection System with the packet analyzer as a basis.

Intrusion Detection System viability development

Once the security tests using OPNET devices and techniques were completed, the next step was to create a proposal for developing an Intrusion Detection System. The goal was to seize the Packet Analyzer node and rebuild it with the ability to alert when illegal traffic patterns are detected. However, as easy as it seems, the Packet Analyzer has to be rebuilt with an internal structure capable of matching traffic patterns with captured packets.

In order to improve our understanding of an IDS architecture we studied an existing IDS software called SNORT, which is open source [4]. This software defines three phases regarding an IDS performance.

The first phase, known as the “sniffing phase”, is responsible for capturing and analyzing data packets. In the first place, data packets have to be copied and saved into memory in order to allow further examination. In second place, packets are decoded according to their type. Once this is done, packet information is extracted and saved in the appropriated structure.

The second phase, known as “IDS”, is responsible for processing matching rules for every single packet captured in the previous phase. SNORT implementation is handled by a two-part function: rule translator and detection engine. The first module is responsible for keeping rules or signatures in structures to perform a quick identification of a matching attack pattern. The second module is responsible for traffic filtering and will save attack patterns to match every packet as they arrive.

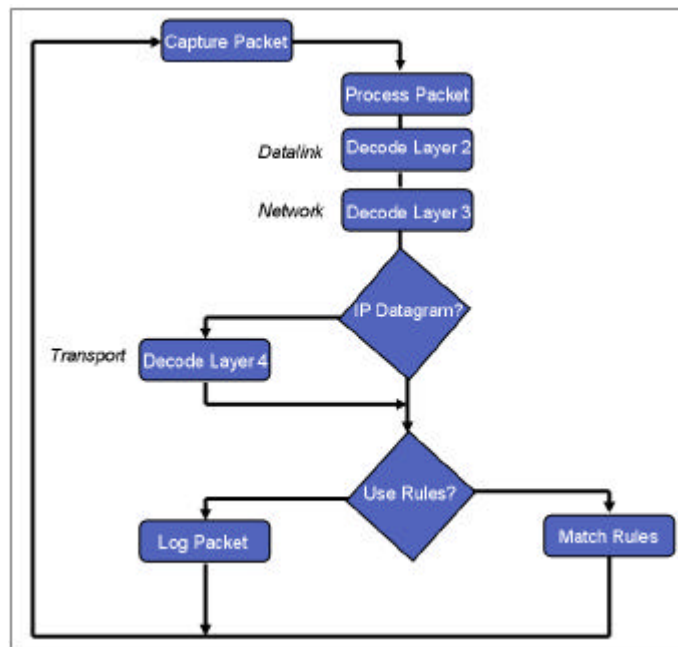


Figure 6: Data packet decoding process

Figure 6 shows the data packet-decoding process inside SNORT architecture. Once the data packet is captured, information contained inside is decoded at different levels in a hierarchical manner. Afterwards, pattern rules are applied to incoming

packets. When an attack is detected, a log file and an alert are generated.

Finally, the last and also the most important phase is the log file and alert generation. This phase will be responsible for warning the user of any traffic pattern match that occurs during the process. When speaking of an IDS rule matching, the concepts of false positive and false negative always arise. The first concept, refers to a rule matching that has been bogusly detected, meanwhile the false negative is an alert that should have been generated but was not.

Modeling network attacks

The second proposal we make, currently under study, is the capture of real traffic attacks. Our aim is to import this traffic to OPNET via the Application Characterization module. The next step will be to continue performing security tests using OPNET, but now with real traffic. In combination with the traffic captures, our goal is to test the IDS device explained above by using traffic attacks against it.

To perform this task it will also be important to keep on track with the OSTTMM document, as it can provide a real security test. This manual will also help to better comprehend how attack traffic can affect the victim system. In addition, it will help to find out what the chances are that the deployed IDS node detects the attack.

Conclusions

Since network security has become a high priority and real security testing is not always easy to perform, our study gives an overall guide to security testing using OPNET. Our goal was to perform a security test as complete as possible in order to be used as a reference in network security testing. The results of our study were very successful and we aim to continue with further tests and the development of new devices and techniques.

To bring the test closer to reality we took as a reference the proposal for secure network implementation called Cisco SAFE, developed by Cisco Systems. The proposal gave us a global security comprehension as well as a closer look to reality. This document was very useful because it made the case study more realistic. It also provided a very well suited understanding of every device and technique performance as well as their location in a network topology.

In order to maintain the study closer to reality, we referred to the OSSTMM for a global comprehension of the many security issues needed to check the performance of the tested devices and techniques in OPNET. This manual also gave us a new area of study: an IDS development and the real traffic attacks captures and importation to OPNET.

Finally, we took the previous studies in network security testing environment and made two new proposals. The first one is based on the Packet Analyzer. Its aim is to turn the Packet Analyzer into an IDS node capable of detecting traffic pattern attacks. The second one is concerning the real traffic attack generation. Its target is to perform real attacks to network systems and capture them in order to, afterwards, import the traffic to OPNET via the ACE module.

References

- [1] SAFE: A Security Blueprint for Enterprise Networks
<http://www.cisco.com/warp/public/779/largeent/issues/security/safe.html>
- [2] SAFE: Extending the Security Blueprint to Small, Midsize, and Remote-User Networks
http://www.cisco.com/en/US/netsol/ns110/ns170/ns171/ns128/networking_solutions_package.html
- [3] SAFE VPN: IPSec Virtual Private Networks in Depth
http://www.cisco.com/en/US/netsol/ns110/ns170/ns171/ns128/networking_solutions_white_papers_list.html
- [4] Snort 2.0 - Protocol Flow Analyzer
<http://www.snort.org/docs>
- [5] OPNET Documentation
<http://www.opnet.com>
- [6] OSSTMM Documentation
<http://www.isecom.org>