

Design and implementation of a user- extensible network packet generator

Guozhen Shi ^{1*}, Yaoming Pan ¹, Feng Yang ¹, Shengyu Chen ², Kai Zhang ³

¹School of Information Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China

²School of Telecommunications Engineering, Xidian University, Xi'an 710071, Shanxi, China

³Xi'an center new software Co., Ltd, Xi'an 710071, Shanxi, China

Abstract. The security problems of network equipment are becoming more and more important with the fast development of the Internet. Aiming at the security, stability and protocol consistency of network security equipment, a user extensible network security device testing framework is proposed, and a user extensible network packet generator is designed based on Libnet and Libpcap library. The system uses the extensibility of XML document and metadata-based reflection technology to design a set of extensible protocol template, which is composed of user-defined data and data length to construct a specific network data packet to meet the specific network security equipment testing. The security, stability and protocol consistency of the network security equipment are tested by the test method through the test in the LAN simulation environment to ensure the security of the network equipment and the network environment.

1. Introduction

Internet services are increasingly rich, online banking, online shopping, mobile office and other increasingly popular, at the same time a growing number of hidden security risks, viruses, worms, and other malicious attacks get endless. IN the actual network environment, network security devices, such as firewall, virtual private network (VPN), intrusion detection system (IDS) and various encryption machines are all essential parts of the network, however, if the network security equipment itself is attacked, it may cause the entire network paralyzed. Therefore, the security of network security equipment is an important guarantee for network system security [1].

During the safety equipment leave factory or security overhaul, the equipment reliability and security of targeted testing is needed. The network packet generator

is an important means of network devices permeability tests to find potential safety problems. At the same time, in order to ensure that network security equipment from external virus attacks, the network protocol used in network security equipment is usually opaque, or of protocol customization. Although the traditional network packet generator can send TCP /IP data packets, it cannot meet so many requirements of network security equipment security, stability testing and protocol conformance testing. So this article designed a user-expandable network packet generator for the above problems. The test results show that the network packet generator can not only send TCP /IP packet simulation attacks, but also can send scalable custom protocol packets, in order to ensure the security, stability and protocol consistency of network security devices.

2 Related works

The network packet generator, the traffic generator, is a tool for detecting devices in the access network, by generating protocol packets through the network packet generator, the real network environment is simulated and the network equipment is tested for security protocol consistency, equipment reliability and performance stability. At present, there are many commonly used tools to generate IP packets, the mainstream tools are sendip, Nessus and sniffer [3].

Sendip is relatively short and more complete, more suitable for use in daily testing. Sniffer tool is easy to use; you can send any possible data packets. The function of Nessus tool is more comprehensive, but in terms of contract awarding sendip. At present, many users develop the network packet generator with automatic simulation attack function, which can tailor the attack time, quantity and so on for a few attacks. However, these test tools, with single function and limited application scope, can only send fixed-protocol packets. They are powerless for application layer protocol data packets and user-defined protocol packets and cannot make consistency tests for a variety of security device protocol, and the network equipment is not suitable for complex test requirements. Furthermore, most of these tools are based on Socket with the inefficient code, also affected by other tasks, the traffic rate and attack effects produced greatly reduced as well. The use of Libnet library can not only simplify programming, and can directly construct the packet and send it to the network, do not need server/client mode [4]. Therefore; this paper designs an extensible the network packet generator based on Libnet open source library on the embedded development board platform.

3. Introduction to technical theory

3.1 Introduction to scalability technology

To implement software scalability, techniques used include: Extensible mark-up Language (XML), plug-in, code automatically generated, dynamic compilation, metadata-based reflection technology and so on [5,6]. The extensible network packet generator designed in this paper achieves scalable design by using extensible mark-up

language, JAXB technology, and metadata-based reflection techniques.

3.1.1 XML extensible mark-up language

A mark-up language used to mark an electronic file to make it structural; XML (Extensible Mark-up Language) is an extensible mark-up language, a subset of the standard general mark-up language. Having self-descriptive, scalability, content and display separation and cross-platform features and advantages, XML is designed to transfer data [7]. This paper designs the extensible protocol template by using the extensibility of XML language. Users can generate custom data packets according to the test template and change the test data in real time.

3.1.2 JAXB technology

JAXB (JAVA Architecture for XML Binding, JAVA XML Binding Architecture) is an industry standard that can generate JAVA classes based on XML Schema. The core of JAXB is the bi-directional mapping technology for XML Schema and Java classes and interfaces, [8]. JAXB technology not only avoids the traditional way to manually parse XML documents, but also can generate Java class objects automatically. The extensible network packet generator designed in this paper uses JAXB technology to reverse line the XML document into Java objects, which solves the problem of writing specific Java class object code and makes the program have extended features.

3.1.3 Flective technology based on metadata

The basic meaning of Metadata is "data about data". In the field of software configuration, metadata is defined as an object in the program is not being processed. Reflective technology is in the program running state, for any class or object; you can know all the attributes and methods of this class, this dynamic access to information and dynamic call object method technology called reflection technology [9].

In this paper, through the reflection technology based on the metadata, you can easily access all attributes and methods of the running class or object, and then

use the acquired attributes and methods for dynamic scalable development.

3.2 Libnet open source library introduction

Libnet is a small interface library implemented by the C language that provides the construction, processing and sending of low-level network packets. The purpose of Libnet's development is to create a simple and unified network programming interface to block the differences in the underlying network programming of different operating systems, allowing programmers to focus on solving key issues [10, 11]. Based on the reusable and highly portable Libnet library, this paper generates network packets at a constant rate on the embedded development platform to detect the device security, stability and protocol consistency.

4 Design and implementation of system function

The network packet generator which can be expanded by users using the upper and lower machine running mode, the host computer is responsible for interactive control, the slave computer to achieve the contract function to ordinary PC and embedded board hardware platform, on which the software part running is developed by JAVA language and C Language correspondingly, the core part are the construction of the scalability of the design of the upper computer and the network protocol development package based on Libnet of the lower computer. The system integrates the general contractor's contracting capabilities with scalable service in a test system, to realize automatic test, multi-function testing and scenario testing, can detect including but not limited to, the IDS, encryption machine and other network security equipment.

4.1 Functional module design

The network packet generator, which can be expanded by user, consists of the upper computer interaction module and the lower computer simulation contract module,

(1) The upper computer interaction module includes the subcontract module input sub-module, the interac-

tive command set construction and sending sub-module, in which the sub-module can be subdivided into interface definition input and XML definition input, but the interface definition input only can complete the simple TCP / IP contract.

The sub-module is the core of the upper computer, mainly to complete the network security equipment online detection structure of the contract command interface display, XML document upload and the corresponding parameter settings. XML document includes network security device custom protocol template document and custom protocol test document, the former document is used to make the system to resolve the agreement corresponding to the POJO class, and the latter document parses the attribute values of the test protocol based on the template class to construct the interactive command set.

The interactive command set construction and delivery sub-module is used to convert multiple groups of packet commands into byte stream group, and send it to the lower machine through the upper and lower machine custom communication protocol.

(2) Lower machine simulation contract module includes control command set processing sub-module, network packet generation sub-module and constant rate output sub-module.

Control command set processing sub-module is mainly for processing the command set sent by the upper computer, including command set split, command recognition and other operations.

Network packet generation sub-module is the core of the lower machine, mainly based on TCP/IP network protocol and custom protocol, network data packets are generated, and the data packet function of the specified command is implemented on the base of Libnet library.

Constant rate output is an indicator of the performance of the network packet generator, through the constant rate output algorithm, making the lower machine in accordance with the upper computer to send network packets according to a constant rate.

4.2 Scalability design

In order to achieve the universality of custom protocol testing for network security devices, you need to

use the advantage of XML configuration language since the descriptive and extensibility, define a set of test templates that conform too many custom protocols. Users only need to change the test data in real time according to the test template. The scalable packetizer can automatically obtain the data of the tag in the template through the metadata-based reflection technology to generate the specified test data packet to realize the system scalability.

The custom protocol XML template uses the XML element tree structure model to determine the attributes of the tag element, the number of tags, and the length of the data, and so on in the attribute list of the root element. The extensible design is implemented in the tag element. The custom protocol XML template tree structure model is shown in Figure 1:

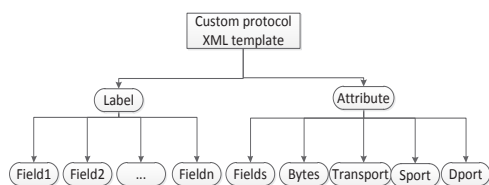


Figure 1. Custom protocol XML template tree structure model

The network protocol application layer header contains several fields, each field has a different data length, and the application layer protocol is based on the TCP/UDP transport layer protocol, so the custom protocol requires the field name, field length, and the transport layer of the protocol to be marked in the template file. The custom protocol can specify 1024 to 65535 any port number, but the standard network protocol already specified has a specified port number, for example, the DNS port number is 53 and the HTTP port number is 80, so the source port number and destination port number should be specified in the template file. In addition, the default value of each field also needs to be listed in the template file, and the system will determine the field attribute data type and create a custom protocol POJO class based on the default value. In short, the XML template file is the field specification for each upper layer protocol and the default padding. The attribute field and field description are shown in Table 1:

Table 1 XML attribute field and description.

XML Label	XML attribute field	Description
Protocol name (root node)	Fields	This is the enumeration of each protocol field, the order of each protocol field corresponds to the order of occurrence, and the middle of each field is separated by commas.
	Bytes	This is an enumeration of the number of bytes in the protocol header of each protocol field. The sequence of each protocol corresponds to the order in which the order appears. The integers are separated by commas.
	Transport	This is the description of the transport layer corresponding to the application layer, including "TCP" and "UDP".
	Sport	This is the source port description of the sender, because a specific protocol has a specific port number. If the protocol constructed by the user is not a standard protocol, any number in the 1024~65535 can be filled here.
	Dport	This is the destination port description of the receiver, and this port description is the same as the sport instruction.
Each protocol field	Null	This is the field description of extensible protocols, and also the key to stitching scalable protocols.

4.3 Network packet generation module design

The network packet generation sub-module is the core of the lower computer, and it is also one of the core of the whole scalable network packet generator. The generation of network packets mainly uses the functions of the construction data provided by the Libnet open source library. This system not only realizes the function of sending the TCP/IP layer network packet by the ordinary sender, but also realizes the application layer network packet which can be extended to send to user.

The specific process of the sub-module is:

(1) First, use the function libnet_init () for memory initialization and environment establishment. What is supposed to be noted here is if the command structure provides the content in the underlying protocol header field, parameter injection type use LIBNET_LINK. If the protocol is above the IP layer and the command structure only provides the application layer protocol header. The parameter injection type uses LIBNET_RAW4;

(2) Packet structure. The principle is from the upper to the lower, the contract unit network packets can be divided into four categories: the link layer packets (MAC, LLC, the SNAP packets), network layer, data packets (IP, ARP packets/RARP), packet transport layer (ICMP, TCP, UDP packets) and application layer packet packets (custom). The concrete structure is to construct the load first, then construct the upper layer protocol header and the lower layer packet header, then assemble each part into a byte array to complete the structure of the packet.

(3) After constructing, use the function Libnet_write () to send the packet through the network card;

(4) Then use the function libnet_destroy () to destroy the applied memory space and initialize the environment.

The specific process design shown in Figure 2:

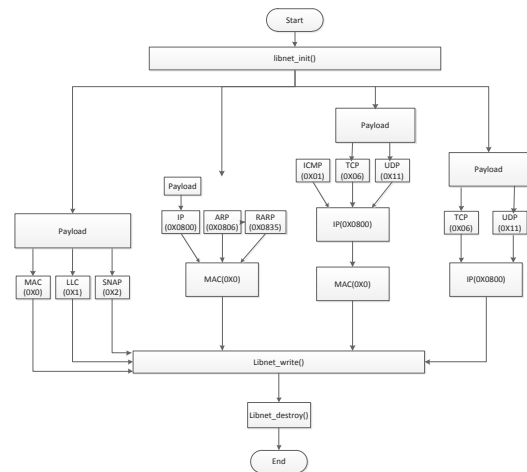


Figure 2.The network data package sub module constructs the packet flow

The table of concrete protocol constructs and protocols as follows:

Table 2.Protocol headers and constructors

protocols	Libnet concrete protocol constructs
MAC	libnet_build_ethernet ()
802.2LLC	libnet_build_802_2 ()
802.2SNAP	libnet_build_802_2snap ()
IPV4	libnet_build_ipv4 ()
ARP	libnet_build_arp ()
RARP	libnet_build_arp ()
ICMP	libnet_build_icmpv4_echo ()
TCP	libnet_build_tcp ()
UDP	libnet_build_udp ()

For the application layer protocol, Libnet only provides some of the protocol head of the construction methods such as DNS, and the use of existing protocol header construction method also limits the scalability of the scaler. In order to meet the scalability of the scaler, this paper chooses the LIBNET_RAW4 primitive socket injection type so that it will be from the network layer to start building data packets without having to fill the Ethernet head, The transport layer structure function of the application layer can be set up according to the transport layer protocol and port number obtained from the control command processing sub-module, and then all the contents of the command structure as a data load, placed in the transport layer protocol load location, and the transmission function complete the custom data

packet transmission.

5 System test results and analysis

5.1 Test the environment

In the test of network security equipment, very few devices placed in the actual network, because the actual network environment is uncontrollable, and the actual network environment is too strong, it is difficult to accurately test the device system, therefore, a special network test environment is constructed.

The test of network equipment is guided by black box testing, use active test, through test and online and offline testing combined method [12] to test the security, stability and reliability of network devices and the consistency of device protocols. Test environment consists of the upper computer, network security equipment, the next crew, three switches and monitoring host. The upper computer sends control commands to the lower machine, and receives the feedback from the lower machine. The lower computer sends the network test packet through the LAN, and the three-layer interaction machine mirrors the test packet port to the monitoring host, and forwards the test packet to the network security device. After crossing the network security device, the packet is processed by the lower computer and the data packet is returned to the upper computer, through comparing the packets captured by the upper computer Wireshark and monitor the host Wireshark and analysing the network security equipment custom network protocol security effect, achieve a goal of the detection of the network equipment security, stability and reliability.

The test environment of the network security device detection tool is shown in Figure 3:

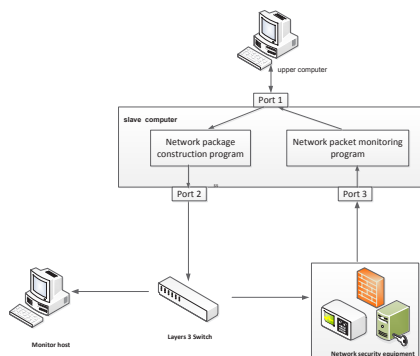


Fig 3. Testing environment for network security de-

vice testing tools

Based on the campus network as a local network, the scalability of the network transmitter is tested, and the computer configuration and tools are shown in Table 3:

Table 3. Computer configuration and tools

Experimental equipment	Configuration
Monitor host	AMD A10, 3.5GHz CPU, 4G Memory
Monitor host	AMD A10, 3.5GHz CPU, 4G Memory
Slave computer	MC-AM335X Development board, 800MHz clock speeds, 256MB DDR3 Memory dual Gigabit Ethernet port
Layers 3 Switch	BDCOM S2210
Network security computer	IDS、Firewall

5.2 Test results and analysis

5.2.1 TCP / IP packet authentication

This network packet generator implements basic contracting capabilities and can simulate network attacks such as Syn Flood, UDP Flood, and Land Attack to detect against network devices.

Extensible network packet generator implementation the function of sending TCP/IP packet, the packet interface shown in Figure 4:



Figure 4. TCP/IP packet sending interface

The network packet generator can send any protocol packets in the TCP / IP protocol stack and send the generated packet command for the above figure. The experimental results are shown in Figure5 below:

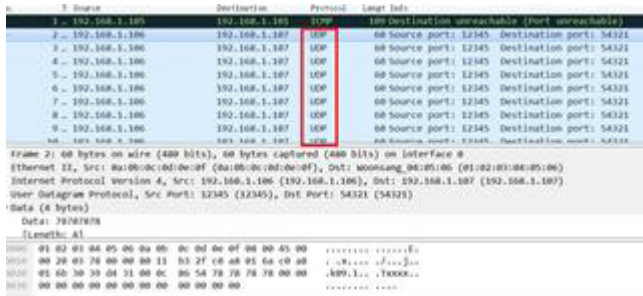


Figure 5. Test results generated by TCP/IP packets

(2) Extensible packet authentication

The network packet generator implements an application layer protocol such as SSL, SSH, DNS, and OICQ, as well as any custom network protocol. In order to verify the scalability of the scaler, the experiment selects the alarm protocol of the SSL protocol and the unencrypted SSH protocol to do the extensibility test. The XML test files are shown in Table 4 and Table 5 respectively.

Table 4. SSL alarm protocol XML file

```
<? Xml version="1.0" encoding="UTF-8"?>
<SSL
fields="contentType,version,length>alertMlever>alertM
descrip" bytes="1,2,2,1,1" transport="tcp"
sport="443" dport="54321">
<!-- TLS 记录层 -->
<contentType>22</contentType>
<version>771</version>
<length>2</length>
<!-- TLS 报警协议 -->
<alertMlever>1</alertMlever>
<alertMdescrip>112</alertMdescrip>
</SSL>
```

Table 5. SSH protocol XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<SSH fields="length,encryptedPacket"
bytes="4,0" transport="tcp" sport="22" dport="54321">
```

```
<length>18</length>
<encryptedPack-
et>encryptedPacketencrypedPacketencrypedPacket</encryptedP
acket>
</SSH>
```

The network packet generator can send any protocol for a given protocol template, construct a custom data packet for Table 4, and Table 5 XML file templates. The Wireshark capture packet results are shown in Figure 6 and Figure 7:

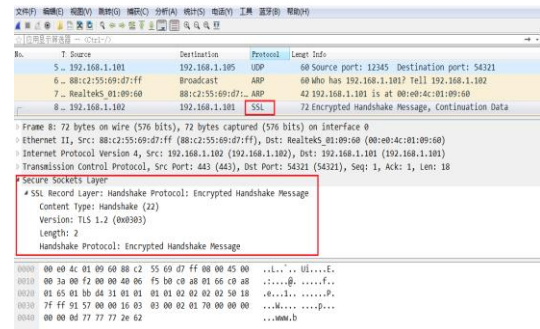


Figure 6. SSL protocol for custom protocol testing

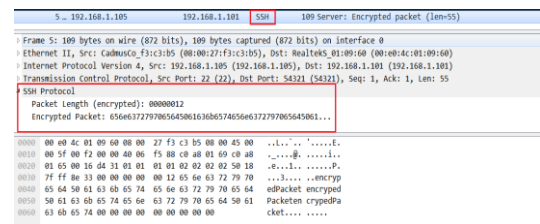


Figure 7. SSH protocol for custom protocol testing

6 Conclusions

This paper designs and develops a user-scalable network packet generator which can detect the security of general network security equipment. Through the concrete programming and project implementation, it is proved that the system can effectively and automatically send various network test packets, so as to assess the security of network security equipment. The system will increase the late password component evaluation module. Thus, the encryption and authentication of security equipment are tested professionally. The system has the advantages of strong expansibility and wide application range and it is easy to use. It is an effective tool for detecting network security equipment and has good practical value.

Acknowledgement

This work was supported by the Beijing Natural Science Foundation (4152048) and National Key Research Program of China (2016YFB0800304).

References

1. Wei Sun, Chuang Bao. Current situation and trends of international network security products market [J]. National Defense Science and technology, 2016, 37(2):59-64.
2. Hong Zhang. Design and implementation of online security detection system for network equipment [D]. Beijing University of Posts and Telecommunications, 2012.
3. Yiqiang Wang. Research and implementation of IP traffic generator [D]. Chengdu University of Technology, 2010.
4. Jordan P, Patten C V, Peterson G, et al. Distributed PowerShell load generator (D-PLG): A new tool for dynamically generating network traffic[C]// International Conference on Simulation and Modeling Methodologies, Technologies and Applications. IEEE, 2017.
5. Oquendo F, Leite J, Batista T. Designing Scalability in Software Architectures [M]// Software Architecture in Action. Springer International Publishing, 2016.
6. Jindal A, Kumar J, Kasralikar R S, et al. Systems and methods for increasing the scalability of software-defined networks [J]. 2016.
7. Xiangyi Geng, Yueping Zhang. XML Basic course, The Second Edition [J]. 2012.
8. Cover R. Java Architecture for XML Binding (JAXB) [J].
9. Deng X, He G, Chen Y, et al. CIM lead-in based on java reflection mechanism in AEMS of Shanghai power grid [J]. Automation of Electric Power Systems, 2007, 31(18):21-25.
10. Quan D, Pei C, Zhu C, et al. Design and Implementation of Network Traffic Generator Based on Libnet [J]. Modern Electronic Technique, 2005. Wang L X, Wen-Wei L I. Active Measurement Tool Development Kit [J]. Computer Systems & Applications, 2014.
11. Meyer R A, Klassen M. Methods, systems, and computer readable media for controlling processor card power consumption in a network test equipment chassis that includes a plurality of processor cards [J]. 2017.